European IT Certification Curriculum
Self-Learning Preparatory Materials

EITC/CL/GCP

Google Cloud Platform

This document constitutes European IT Certification curriculum self-learning preparatory material for the EITC/CL/GCP Google Cloud Platform programme.

This self-learning preparatory material covers requirements of the corresponding EITC certification programme examination. It is intended to facilitate certification programme's participant learning and preparation towards the EITC/CL/GCP Google Cloud Platform programme examination. The knowledge contained within the material is sufficient to pass the corresponding EITC certification examination in regard to relevant curriculum parts. The document specifies the knowledge and skills that participants of the EITC/CL/GCP Google Cloud Platform certification programme should have in order to attain the corresponding EITC certificate.

Disclaimer

This document has been automatically generated and published based on the most recent updates of the EITC/CL/GCP Google Cloud Platform certification programme curriculum as published on its relevant webpage, accessible at:

https://eitca.org/certification/eitc-cl-gcp-google-cloud-platform/

As such, despite every effort to make it complete and corresponding with the current EITC curriculum it may contain inaccuracies and incomplete sections, subject to ongoing updates and corrections directly on the EITC webpage. No warranty is given by EITCI as a publisher in regard to completeness of the information contained within the document and neither shall EITCI be responsible or liable for any errors, omissions, inaccuracies, losses or damages whatsoever arising by virtue of such information or any instructions or advice contained within this publication. Changes in the document may be made by EITCI at its own discretion and at any time without notice, to maintain relevance of the self-learning material with the most current EITC curriculum. The self-learning preparatory material is provided by EITCI free of charge and does not constitute the paid certification service, the costs of which cover examination, certification and verification procedures, as well as related infrastructures.

TABLE OF CONTENTS

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: INTRODUCTIONS
TOPIC: THE ESSENTIALS OF GCP

Welcome to the didactic material on the essentials of Google Cloud Platform (GCP). In this material, we will provide you with an overview of GCP, its key features, and how to get started with it.

To begin, cloud.google.com is your starting point for accessing all the necessary information about GCP. It provides product information, documentation, detailed solutions with architectures and code, and pricing details. You can also find support, customer stories, and information on how GCP differs from other public clouds.

The Google Cloud Console is where you will spend a significant amount of time exploring and using the platform. It allows you to configure billing accounts, create and manage projects, and manage all your GCP resources. Each product and service has its own section in the console, with dashboards, configurations, and settings. The console also offers interactive quick start experiences for various products, allowing you to quickly get started with them.

Additionally, the Cloud Console provides a marketplace with ready-to-go software stacks, enabling you to deploy production-grade solutions with ease. It also integrates Identity and Access Management (Cloud IAM), which allows you to set up the right permissions for your employees and provides a unified view of security policies across your organization. Quota management and mobile apps for monitoring and managing your GCP applications are also available in the Cloud Console.

While the Cloud Console is powerful and flexible, you can also perform all actions from the command line using the gcloud command-line interface (CLI). The CLI is scriptable and comes with the Google Cloud SDK. If you prefer a web-based environment, you can use Cloud Shell, which is a shell environment hosted on GCP. It provides a web code editor and access to a virtual machine for managing your projects and resources.

Before using GCP, you will need a Google account. You can create a new account or use an existing one, such as your Gmail account. Enabling billing for your project is recommended, as it allows you to access GCP products beyond their free tier. If you're not eligible for the free trial, you can still benefit from the generous always free tier.

To get started with GCP, we recommend checking out the main menu in the GCP Console. Start with the platform tutorial to get a sense of how to use Cloud Console effectively. Interactive tutorials and hands-on labs are also available within the console, allowing you to learn while using the platform. Explore the main categories of GCP products, including compute, storage, networking, DevOps, tools, big data, and artificial intelligence.

Throughout your journey with GCP, make use of the key resources available to you, such as documentation, in-depth tutorials, support, training, and free code labs. GCP can be overwhelming at first, but with the guidance provided in this material, we hope you feel confident in bringing your best ideas to life in the cloud.

We are excited to see what you build and look forward to your feedback. If you found this material helpful, please like, subscribe, and comment.

Google Cloud Platform (GCP) is a cloud computing service provided by Google. It offers a wide range of cloud-based services and products that can be used to build, deploy, and scale applications and websites. In this didactic material, we will provide an introduction to the essentials of GCP.

One of the key benefits of using GCP is its scalability. GCP allows you to easily scale your applications and services up or down based on your needs. This means that you can handle sudden increases in traffic without any issues, ensuring a smooth user experience.

Another important aspect of GCP is its reliability. Google has a vast infrastructure that is spread across multiple data centers around the world. This ensures that your applications and data are always available, even in the event of hardware failures or natural disasters.

GCP also offers a wide range of services to support different types of workloads. For example, if you need to store and retrieve large amounts of data, you can use Google Cloud Storage. If you need to process and analyze data, you can use Google BigQuery. There are also services for machine learning, artificial intelligence, and serverless computing, among others.

To get started with GCP, you will need to create a GCP project. A project is a container that holds all the resources and services you will use within GCP. Once you have created a project, you can start provisioning resources such as virtual machines, storage buckets, and databases.

GCP also provides a web-based console called the Cloud Console, which allows you to manage and monitor your resources. The Cloud Console provides a user-friendly interface for performing tasks such as creating virtual machines, setting up networking, and configuring security.

In addition to the Cloud Console, GCP also offers a command-line interface (CLI) and APIs that allow you to automate tasks and integrate GCP with other tools and services.

GCP is a powerful cloud computing platform that offers scalability, reliability, and a wide range of services. It provides the tools and infrastructure needed to build, deploy, and scale applications and websites. Whether you are a small startup or a large enterprise, GCP has the capabilities to meet your needs.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: INTRODUCTIONS
TOPIC: GCP FREE TIER AND FREE TRIAL

Understanding the cost of using a cloud provider can be daunting, especially if you're just starting out and don't want to spend money on something you're not sure about. In this material, we will discuss Google Cloud Platform's free trial and the Always Free tier, and why it's important not to confuse the two.

Let's begin with the free trial. When you sign up for GCP, we recommend that you take advantage of the $300 free credits. To access this trial, you will need to create an account (if you don't already have one) and provide your credit card or bank account details. It's important to note that the trial is completely free, and your credit card will not be charged. At the beginning of each month, you will receive billing statements that detail how much of the $300 credit you have used in the previous month. This allows you to understand the potential costs of running your services on GCP and identify where you are consuming the most resources. To manage your budget effectively, you can set up budget alerts for your billing account or specific projects you are working on. These alerts will notify you if costs are escalating, enabling you to take action to control them.

During the free trial, you have access to the entire Google Cloud Platform, without any limitations. This means you can use the real services and not just a sandbox environment. However, there are a few restrictions. You cannot request credit increases, and you are not allowed to mine cryptocurrencies. Nonetheless, there are numerous things you can try out, such as creating virtual machines, transferring data to cloud storage buckets, setting up Kubernetes Engine clusters, running Hadoop or Spark workloads on Cloud Dataproc, or training machine learning models using Cloud AutoML. These are just a few examples, and the possibilities are vast.

It's worth mentioning that you can create multiple projects, all linked to the billing account that has the $300 credits. This allows you to experiment with different setups in separate environments. As you navigate GCP and encounter questions, there are several resources available to assist you. You can refer to the documentation and community forums for guidance, or use the free trial troubleshooter for specific concerns. If you require more formal support, you have the option to purchase silver level support using a portion of your $300 credit directly from the console's support section.

The free trial is valid for 12 months or until you exhaust the $300 credit. The remaining credit and days are displayed prominently at the top of the Google Cloud Platform console and in the billing section. If you use up all the credits, you will not be charged, but your resources will be paused unless you upgrade to a paid account within 30 days. Upgrading to a paid account at any time allows you to retain any remaining credit and makes you responsible for the cost of your GCP resources.

Now let's discuss the Always Free tier. Even after upgrading to a paid account, GCP continues to offer generous free tiers for 16 of its main products. These free tiers, also known as free quotas, allow users to utilize specific GCP products for free every month. For example, the first 2 million invocations of Cloud Functions per month are free, as are the first 60 minutes of Cloud Speech API or the first daily 120 build minutes of CloudBuild. Some services, such as Compute Engine or Cloud Storage, may require you to use specific GCP regions.

The Always Free tier does not have an expiration date and provides ongoing access to these products. If you would like more information about the free trial or the Always Free tier, you can refer to the detailed FAQ page.

We hope that the combination of the free trial, the Always Free tier, and the straightforward GCP pricing structure will empower you to build remarkable projects on GCP. If you found this material helpful, please show your support by liking, subscribing, commenting, and sharing. Stay tuned for more GCP Essentials materials.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: INTRODUCTIONS
TOPIC: GCP CONSOLE TOUR

Cloud Console is a powerful graphical tool provided by Google Cloud Platform (GCP) to manage all your GCP resources regardless of their data center location. In this material, we will cover the core features of Cloud Console that will help you build and manage your applications on GCP effectively.

GCP resources are the fundamental components that make up the Google Cloud Services. These resources include Compute Engine virtual machines, Cloud Pub/Sub topics, cloud storage buckets, App Engine instances, and more. To organize these resources, GCP uses a hierarchical structure with projects and folders. Projects are the first level of this hierarchy and allow you to group resources together. Every resource must belong to exactly one project. Optionally, projects can also belong to organizations, providing centralized visibility and control across all projects in a given organization. For added flexibility, you can further organize your company departments or teams into folders. This hierarchy offers an ownership chain, meaning that when you delete a parent, all its resources are also deleted. It also forms the basis for access control policy inheritance.

Navigating across your GCP projects is made easy with the scope picker in Cloud Console. By switching projects, you can tailor the view to that specific project and all its child resources. GCP services are accessible through the left-hand navigation menu, organized by product area such as big data, compute, networking, and more. The home dashboard provides a high-level overview of the selected GCP project, highlighting key metrics, billing information, and other useful details. You have the freedom to customize your dashboard by hiding, showing, and reordering cards on the page.

The Activity Stream feature in Cloud Console allows you to understand all the activities that occur across your GCP resources in one place. You can see updates to projects, track your teammates' actions, and audit access to your GCP resources. The search bar in Cloud Console enables you to quickly access Google Cloud Platform products and any of your resources across GCP. Simply search for the product or the name of one of your projects.

The APIs and Services section of Cloud Console is where you can manage hundreds of APIs offered by Google, including GCP, machine learning, Google Maps, G Suite, and Analytics APIs. Here, you can enable or disable a service, generate or revoke credentials, and monitor requests.

If you ever need assistance in navigating GCP, the support team is available to help. You can access your support cases relating to development questions and production issues directly from the Console navigation menu.

Google Cloud Identity and Access Management (Cloud IAM) is a feature that allows you to manage and create permissions for your GCP resources. As your team grows, you can grant access to teammates using the IAM and Admin section. You can add users, groups, or service accounts and assign them roles to grant them the necessary permissions. There are many predefined roles to choose from, providing a fine-grained access control system.

Google Cloud Shell is a convenient feature that provides command line access to your cloud resources directly from your browser. It eliminates the need for installing any software on your system and allows you to manage your projects and resources seamlessly. Cloud Shell comes with a web editor and is powered by a virtual machine with persistent disk space and up-to-date software for all your development needs.

The billing section of Cloud Console is where you manage billing accounts and link your projects to them. A billing account serves as a payment method, and without it, a project cannot use GCP products beyond their free tiers. You can change the billing account for a project at any time, set budget alerts to manage costs, and set up triggered actions for projects or accounts. Additionally, you can generate billing exports and reports to gain a better understanding of your spend.

Finally, Cloud Console is also available as a free mobile app for both Android and iOS. The app allows you to monitor the health of your services with customizable graphs showing CPU usage, network usage, QPS, and more. It also provides billing alerts and allows you to manage incidents and alerts, navigate through error and

crash reports, and perform actions such as starting and stopping Compute Engine instances and accessing their logs.

Cloud Console is an essential tool for managing and interacting with the Google Cloud Platform (GCP). In this didactic material, we will explore the features and functionalities of Cloud Console without referencing any specific material or speaker.

Cloud Console provides a user-friendly interface that allows users to easily navigate and control their GCP resources. It offers a centralized location for managing various aspects of your cloud infrastructure, such as virtual machines, storage, databases, and more.

One of the main advantages of Cloud Console is its simplicity and ease of use. It provides a visually appealing dashboard that gives users a comprehensive overview of their GCP projects. From the dashboard, users can access different services and resources with just a few clicks.

The navigation menu on the left-hand side of the console allows users to access different sections and services within GCP. Each section is organized in a logical manner, making it easy to locate and manage specific resources. For example, users can navigate to the Compute Engine section to manage virtual machines or the Cloud Storage section to manage storage buckets.

Within each section, users can perform various actions and configurations. For instance, in the Compute Engine section, users can create and manage virtual machines, configure networking settings, and monitor resource utilization. Similarly, in the Cloud Storage section, users can create and manage storage buckets, set access controls, and view storage usage.

Cloud Console also provides a search functionality that allows users to quickly find specific resources or services. By simply typing in keywords or names, users can locate the desired resource without the need to navigate through multiple sections.

In addition to managing resources, Cloud Console offers features for monitoring and troubleshooting. Users can view logs, metrics, and diagnostics information to gain insights into the performance and health of their GCP resources. This helps in identifying and resolving any issues or bottlenecks that may arise.

To summarize, Cloud Console is a powerful and user-friendly tool for managing and interacting with the Google Cloud Platform. It provides a centralized interface for managing resources, navigating through different sections, and performing various configurations. With its simplicity and comprehensive features, Cloud Console empowers users to efficiently build and manage their cloud infrastructure.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: INTRODUCTIONS
TOPIC: GCP DEVELOPER AND MANAGEMENT TOOLS

Google Cloud Platform (GCP) provides a range of tools to help users efficiently manage their resources. In this material, we will introduce three key tools: the Cloud SDK, the gcloud command line, and Cloud Shell.

The Cloud Console is a web-based UI that serves as a centralized platform for managing GCP resources, projects, and billing information. It offers a powerful and integrated environment to kickstart your GCP experience.

The Cloud SDK is a set of command line tools, including gloud, gsutil, and bq, which allow users to access Compute Engine, Cloud Storage, BigQuery, and other GCP products from the command line. These tools can be used interactively or in automated scripts. The Cloud SDK is supported on Linux, Mac OS, and Windows, and requires Python to run. It can be installed using apt-get, yum, or an interactive/non-interactive installer.

The gcloud command is the most commonly used part of the Cloud SDK. After installation, users can run "gcloud init" to perform setup tasks. This command authorizes the Cloud SDK tools to use user account credentials to access GCP resources and sets up a configuration for the active account, current project, and optionally, the default Compute Engine region and zone. Users can view the current configuration using the "gcloud config list" command.

Managing and updating the Cloud SDK is also done using the gcloud command. The SDK installs generally available (GA) gcloud commands by default, but additional functionality can be installed as SDK components named alpha and beta. Updates to existing components can be achieved with the "gcloud components update" command.

With the gcloud command, users can perform a wide range of tasks, such as managing VMs and storage buckets, setting up networking and firewalls, building and testing locally, deploying to production, and monitoring logs. These tasks can also be automated using scripts and CI/CD tools.

For Windows PowerShell users, Google provides Cloud Tools for PowerShell, which allows managing GCP resources in a familiar way.

Cloud Shell offers an always-available, browser-based environment with gcloud and other favorite tools, such as Git, Bash, Docker, kubectl, and language-specific tools. It is a temporary virtual machine running a Debian image on GCP, with 5 gigabytes of persistent disk storage. Cloud Shell can be used to run gcloud commands without the need for installation, update, or configuration. It also provides built-in authorization to GCP projects and resources.

Cloud Shell can also serve as a development environment with its web preview mode, enabling a browser to access a web server running on Cloud Shell. It offers advanced features like boost mode for better CPU and memory, tmux for session management, and the ability to customize the environment with additional tools using a user-provided Docker image.

It is important to note that a Cloud Shell session terminates after one hour of inactivity, and any modifications made outside of the home directory are lost when the instance is terminated. However, users can leverage customization features to preserve their modifications.

Google Cloud Platform offers various tools to facilitate the development and management of resources. The Cloud SDK, gcloud command line, and Cloud Shell provide powerful and flexible options for accessing and managing GCP services.

Cloud Computing - Google Cloud Platform - Introductions - GCP Developer and Management Tools

In this educational material, we will discuss the developer and management tools available in Google Cloud Platform (GCP). These tools are essential for building and managing applications on the cloud.

One of the key tools provided by GCP is the iCloud SDK. This software development kit allows developers to interact with GCP services and build applications using popular programming languages such as Python, Java, and Node.js. The iCloud SDK provides a set of command-line tools that enable developers to manage their cloud resources, deploy applications, and perform various administrative tasks.

Another important tool is the Cloud Shell. This web-based command-line interface provides developers with a fully functional Linux shell environment that is pre-configured with the necessary tools and libraries. With Cloud Shell, developers can easily access and manage their GCP resources from any device with a web browser. It eliminates the need for setting up local development environments and provides a seamless experience for developing and testing applications on the cloud.

By using the developer and management tools provided by GCP, developers have everything they need to build innovative and powerful applications. Whether it's deploying a simple web application or creating complex data analytics pipelines, GCP offers a comprehensive set of tools to support the entire development lifecycle.

GCP provides a range of developer and management tools, including the iCloud SDK and Cloud Shell, that enable developers to build and manage applications on the cloud. These tools offer a seamless and efficient development experience, allowing developers to focus on creating amazing applications without worrying about infrastructure management.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP BASIC CONCEPTS
TOPIC: COMPUTE ENGINE

Welcome to this educational material on Compute Engine, a fundamental concept in Google Cloud Platform (GCP) for cloud computing. Compute Engine refers to customizable virtual machines that run in the Google Cloud. These virtual machines can be tailored to meet your specific needs by selecting from predefined or custom machine types.

Predefined machine types come with combinations of CPU and memory that are suitable for general purposes. However, if these predefined options do not meet your requirements, you can opt for custom machine types. Custom machine types allow you to choose the exact number of CPUs and the amount of memory needed for your workloads.

There are three different machine-type families available in Compute Engine. The general-purpose family is well-suited for general workloads like web servers and databases. If you are unsure about which family to choose, general-purpose instances are recommended. Compute-optimized machines are ideal for compute-intensive applications such as high-performance computing, gaming, electronic design automation, and single-threaded apps. Memory-optimized instances are designed for memory-intensive workloads like in-memory databases, SAP HANA, or real-time analytics. Additionally, graphical processing units (GPUs) can be added to accelerate computationally-intensive workloads such as machine learning or medical analysis.

Using Compute Engine is straightforward. Simply select the desired machine type and location, and the instance will be created for your use in that specific location. Compute Engine offers several features that make it an excellent choice. Live migration allows your applications to continue running during maintenance mode without interruptions. It also provides sizing recommendations, helping you optimize costs by using the appropriate instance size for your workload. Furthermore, Compute Engine supports the deployment of containers, making it suitable for container workloads.

In terms of cost, Compute Engine follows a pay-as-you-go model. You only pay for what you use. However, there are opportunities to save costs through various discounts. Sustained-use savings are automatic discounts applied to instances that are run for a significant portion of the month. If you know your usage upfront, you can take advantage of committed-use discounts, which can lead to savings of up to 57% without any upfront cost. For certain workloads, you can save up to 80% by using short-lived preemptible instances, which are ideal for batch jobs and fault-tolerant workloads.

Compute Engine offers a wide range of use cases, including running websites and databases, migrating existing systems to Google Cloud, and running Windows applications by bringing your own licenses or using the included licensed images. To explore more about Compute Engine, visit cloud.google.com/compute.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP BASIC CONCEPTS
TOPIC: CLOUD STORAGE

Cloud Storage is a global, secure, and scalable object store provided by Google Cloud Platform (GCP). It is designed for storing immutable data such as images, text, videos, and other file formats. In Cloud Storage, data is organized into buckets, which are associated with a project and grouped under an organization.

You can upload objects to a bucket and download objects from it using the console or gsutil commands. By default, data at rest in Cloud Storage is encrypted. Additionally, you have the option to secure it with your own encryption keys using Google Cloud's Key Management service or your own key management service on-premise.

Cloud Storage provides fine-grained access control, allowing you to grant permissions to specific members and teams or make objects fully public for use cases such as websites. When creating buckets, you have different options depending on your budget, availability requirements, and access frequency.

- Standard regional or multiregional buckets are suitable for high performance, frequent access, and highest availability.
- Nearline storage is designed for data that is accessed once a month.
- Coldline storage is intended for data accessed less than once a quarter.
- Archive storage is the most cost-effective option for data that you want to put away for years.

While standard storage costs more, it offers automatic redundancy and frequent access options. Nearline, Coldline, and Archive storage classes provide 99% availability and cost significantly less.

Cloud Storage also offers automatic object versioning, eliminating the need to worry about version control. With Object Lifecycle Management, you can automatically transition data to lower-cost storage classes based on its age or when a newer version of a file is stored.

Accessing data stored in Cloud Storage is straightforward, as it can be done with a single API call for all storage classes. Standard regional and multiregional buckets are ideal for hosting static websites, streaming, and storing documents. Nearline and Coldline storage classes are commonly used for backups and disaster recovery. Archive storage is best suited for long-term archiving purposes.

Cloud Storage provided by Google Cloud Platform is a reliable and flexible solution for storing and managing data in the cloud. It offers various storage classes to cater to different access requirements and provides features like encryption, access control, versioning, and lifecycle management.

For more information on Cloud Storage, you can visit the official documentation at cloud.google.com/storage.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP BASIC CONCEPTS
TOPIC: CLOUD SQL

Cloud SQL is a fully managed relational database service provided by Google Cloud Platform (GCP). It supports MySQL, PostgreSQL, and SQL Server databases and offers various features to simplify database management.

One of the key benefits of Cloud SQL is that it reduces maintenance costs and automates database provisioning, storage capacity management, replication, and backups. It provides a quick setup process with standard connection drivers and built-in migration tools.

To set up a Cloud SQL instance, you need to select the region and zone where you want the instance to be created. You also have configuration options to choose the machine type, storage type (solid state or hard disk drives), and storage capacity. Higher storage capacity can lead to better performance.

Cloud SQL also offers automated backups and recovery options. You can set time slots and locations for backups. For production applications, it is recommended to enable high availability (HA). By enabling HA, the database instance will automatically failover to another zone in case of an outage. You can also create cross-regional replicas to protect from regional failures.

Migrating an existing MySQL database to Cloud SQL is made easy with the Cloud Console. It provides a "migrate data" button that guides you through the process. You need to provide your data source details, create a Cloud SQL read replica using a SQL dump file, sync the replica with the source, and finally promote the read replica to the primary instance with minimal downtime.

Data in Cloud SQL is encrypted at rest and in transit, ensuring its security. External connections can be encrypted using SSL or the Cloud SQL Proxy tool. This tool helps you connect to your Cloud SQL instance from your local machines.

Cloud SQL can be used as a relational database for applications hosted within Google Cloud, such as App Engine, Cloud Run, Compute Engine, Kubernetes Engine, or Cloud Functions. It can also be connected to applications hosted outside of Google Cloud.

The pricing of Cloud SQL varies depending on the type of database (MySQL, PostgreSQL, or SQL Server), instance type, storage, and network usage. SQL Server also has additional licensing costs.

Some common use cases for Cloud SQL include online transaction processing applications, like order or payment processing apps, where fast response times are crucial.

To learn more about Cloud SQL, you can visit cloud.google.com/sql.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP BASIC CONCEPTS
TOPIC: BIGQUERY

BigQuery is Google Cloud's enterprise data warehouse that enables users to ingest, store, analyze, and visualize large amounts of data easily. It is designed to help organizations aggregate data from different sources, process it, and make it readily available for data analysis to support strategic decision-making.

There are two ways to ingest data into BigQuery: batch uploading and streaming. Batch uploading allows you to upload data in batches, while streaming enables you to deliver real-time insights by directly streaming data into BigQuery.

As a fully-managed data warehouse, Google takes care of the infrastructure, allowing users to focus on analyzing their data at a petabyte scale. BigQuery supports Structured Query Language (SQL) for data analysis, making it familiar to those who have worked with ANSI-compliant relational databases in the past.

BigQuery also offers BigQuery ML, which allows users to create machine learning models using their enterprise data. With just a few lines of SQL, users can train and execute models on their BigQuery data without the need to move it around.

When it comes to data visualization, BigQuery integrates with Looker and other business intelligence tools in Google Cloud's partner ecosystem.

Getting started with BigQuery is straightforward. After creating a Google Cloud Platform (GCP) project, users can immediately start querying public datasets hosted by Google Cloud or load their own data into BigQuery for analysis.

Interacting with BigQuery can be done through three different methods: using the UI and Cloud Console, using the BigQuery command line tool, or making API calls using client libraries available in various languages.

BigQuery is integrated with Google Cloud's Identity and Access Management Service, ensuring secure data sharing and analytical insights across the organization.

The cost of using BigQuery involves paying for storing and querying data, as well as streaming inserts. Loading and exporting data are free of charge. Storage costs are based on the amount of data stored and have different rates depending on data change frequency. Query costs can be on-demand, where users are charged per query based on the data processed, or flat rate for customers who want to purchase dedicated resources.

To learn more about BigQuery, visit cloud.google.com/bigquery.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP BASIC CONCEPTS
TOPIC: DATAFLOW

Dataflow is a serverless, fast, and cost-effective service provided by Google Cloud Platform (GCP) that supports both streaming and batch processing of data. It is designed to capture, process, and analyze data generated in real-time from various sources such as web sites, mobile apps, IoT devices, and other workloads. Dataflow enables businesses to transform data into a format that is conducive for analysis and effective use by downstream systems.

Dataflow works by following a three-step data processing pipeline. Firstly, the data is read from a source and stored in a Parallel Collection (PCollection), which is designed to be distributed across multiple machines. Secondly, one or more operations, known as transforms, are performed on the PCollection, creating new PCollections after each transform. Finally, the final PCollection is written to an external sink.

To use Dataflow, you can create Dataflow jobs using various methods such as the Cloud Console UI, the gcloud command-line interface, or the API. Dataflow provides options for creating jobs, including the use of prebuilt templates, writing SQL statements, or utilizing AI Platform Notebooks. Dataflow templates offer a collection of prebuilt templates, and you can also create custom templates to share with others in your organization. Dataflow SQL allows you to use SQL skills to develop streaming pipelines directly from the BigQuery web UI. Additionally, AI Platform Notebooks can be used to build and deploy data pipelines using the latest data science and machine learning frameworks.

Dataflow provides inline monitoring, which allows direct access to job metrics for troubleshooting pipelines at both the step and worker level. It also offers security features such as encryption at rest and in transit. Access to internal systems can be restricted by turning off public IPs and leveraging VPC service controls. Furthermore, pipelines can be protected with customer-managed encryption keys.

The cost of using Dataflow is billed in per-second increments on a per-job basis, depending on whether it is streaming or batch data. Flexible resource scheduling can be utilized for batch data processing to reduce costs using advanced scheduling techniques. Each Dataflow job requires at least one Dataflow worker, and the price depends on the worker configurations.

Dataflow is a versatile tool suitable for processing and enriching both batch and streaming data for downstream systems such as analysis, machine learning, and data warehousing. It can be used for various scenarios, including stream analytics for real-time business insights, real-time AI for predictive analytics and fraud detection, processing log data streams for system health insights, and data aggregation and analysis.

To learn more about Dataflow, you can visit cloud.google.com/dataflow.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP BASIC CONCEPTS
TOPIC: GOOGLE KUBERNETES ENGINE GKE

Google Kubernetes Engine (GKE) is a managed service provided by Google Cloud Platform (GCP) for running Kubernetes. Kubernetes is an open-source platform used for managing containerized workloads and services. GKE makes it easy to create clusters and offers advanced cluster management features such as load balancing, auto scaling, auto upgrades, auto repairs, logging, and monitoring.

A GKE cluster consists of a control plane and one or more nodes. The control plane includes the Kubernetes API server, scheduler, storage, and core resource controllers. It is responsible for managing the cluster's nodes, scheduling workloads, managing networks, storage, lifecycle, scaling, and upgrades. Nodes run the services necessary to support the containers that make up the cluster's workloads. Each node includes a container runtime and the Kubernetes node agent, Kublet, which communicates with the control plane and is responsible for starting and running containers as scheduled on that node.

In order to deploy a workload on a GKE cluster, the workload must be packaged into a container. This can be done using Cloud Code, which allows you to write your apps and send the code to a source code repository. From there, a build process in Cloud Build creates container images that can be stored in Container Registry and deployed into GKE.

GKE provides high availability options with two types of clusters: zonal and regional. Regional clusters have multiple control planes across multiple zones in a region, making them better suited for high availability. Zonal clusters have a single control plane in a single zone. Regional clusters have longer propagation times for cluster configuration changes because they must propagate across all control planes. It is recommended to choose regional clusters when availability is more important than flexibility, and to use zonal clusters when availability is less of a concern and rapid cluster creation or upgrades are needed.

GKE also offers four types of autoscaling for workloads and infrastructure. Horizontal pod autoscaler adds or removes pods based on utilization metrics like CPU and memory. Vertical pod autoscaler sizes pods based on resource requirements. Cluster autoscaler adds or removes nodes based on the scheduled workload. Node auto-provisioning dynamically creates new nodes with resources that match the needs of the pods.

GKE is designed with security in mind. It is secure by default, with automatic data encryption at rest and in transit. The OS images deployed on GKE are Google certified, ensuring a secure environment for running containerized applications.

GKE is a managed service provided by GCP for running Kubernetes. It simplifies the creation and management of Kubernetes clusters, offering advanced features for load balancing, auto scaling, and more. GKE clusters consist of a control plane and nodes, with the control plane responsible for managing the cluster and the nodes running the containerized workloads. GKE provides high availability options, autoscaling capabilities, and a secure environment for running containerized applications.

Clusters in Google Kubernetes Engine (GKE) can be accessed without a public IP on the internet, ensuring secure access control. This is achieved through the use of identity and access management (IAM) and role-based access controls (RBAC). GKE also provides trusted networking capabilities, allowing you to connect to and isolate clusters using a global Virtual Private Cloud (VPC). Additionally, global load balancing enables the deployment of public services behind a single global Anycast IP, simplifying network configuration.

To enhance security, GKE offers several features. Cloud Armor provides easy protection against Layer 7 and DDoS attacks, safeguarding your applications. Networking policies allow you to control the communication between pods within your cluster, ensuring secure and controlled data flow.

GKE also includes tools to verify, enforce, and improve the security of your infrastructure. Binary authorization ensures that only properly signed containers are deployed to production, preventing the execution of unauthorized or malicious code. Vulnerability scanning of container images identifies security vulnerabilities early in the continuous integration/continuous deployment (CI/CD) pipeline. Moreover, the base images used in GKE are managed, automatically receiving patches and updates to address security vulnerabilities.

If you are interested in getting started with containers quickly, GKE is a suitable choice. Visit cloud.google.com/kubernetesengine to explore GKE and its capabilities.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP BASIC CONCEPTS
TOPIC: CLOUD CDN

Cloud CDN is a content delivery network provided by Google's Global Edge Network. It is designed to accelerate the delivery of web and video content by bringing the content as close to the user as possible. This helps reduce latency, cost, and load on backend servers, making it easier to scale to millions of users.

When a user makes a request to a website or app, the request is routed to the closest Google Edge Node, of which there are 120 globally. From there, the request goes to the global HTTP(S) load balancer and then to the backend or origin. With Cloud CDN enabled, the content is served directly from cache.

Cache is a group of servers that store and manage cacheable content, such as JavaScript, CSS, images, and videos. Cloud CDN can automatically cache this content by using recommended cache modes to cache all static content. If more control is needed, Cloud CDN can be directed to cache content by setting HTTP headers on responses. It is also possible to force all content to be cached, ignoring the private, no-store, or no-cache directives in cache control response headers.

When a request is received by Cloud CDN, it looks for the cached content using the cache key, typically the URI. If a cached response is found, it is retrieved from cache and sent to the user, resulting in a cache hit. This saves time and resources by avoiding the need for the origin server to process the request. If the content is not found in cache, it is considered a cache miss. In this case, Cloud CDN may attempt to retrieve the content from a nearby cache using cache-to-cache fill. If the content is not available in any nearby cache, the request is sent to the origin server.

The maximum lifetime of an object in cache is defined by the TTLs (time to live values) set by cache directives from each HTTP response or cache modes. When the TTL expires, the content is evicted from cache.

To use Cloud CDN, it can be set up through the GCloud Command Line interface, Cloud Console, or the APIs. It leverages Google Cloud global external HTTP(S) load balancers for routing, health checking, and Anycast support. Enabling Cloud CDN is as simple as checking a box while setting up the backends or origins.

Cloud CDN also supports hybrid architectures, allowing integration with on-premises or other cloud services. It can be used in conjunction with Google Cloud Storage for easy content management and caching.

From a security perspective, data is encrypted at rest and in transit from Google Cloud load balancing to the backend, ensuring an end-to-end encrypted experience. URLs and cookies can be programmatically signed to limit video segment access to authorized users only. The signature is validated at the CDN Edge, blocking unauthorized requests.

Cloud CDN is a powerful tool for improving performance and reducing serving costs for regularly accessed content. By automatically caching static content, it brings the content closer to the user, resulting in faster and more efficient delivery.

Cloud CDN (Content Delivery Network) is a service provided by Google Cloud Platform (GCP) that helps to deliver content to users quickly and efficiently. It works by caching content in multiple locations around the world, allowing users to access the content from a location that is geographically closer to them. This reduces latency and improves the overall performance of websites and applications.

One of the key benefits of using Cloud CDN is its ability to handle high traffic loads. By distributing content across multiple servers, it can handle large amounts of traffic without affecting the performance or availability of the content. This is particularly useful for websites and applications that experience sudden spikes in traffic, such as during product launches or major events.

Cloud CDN also offers protection against distributed denial of service (DDoS) attacks. By distributing content across multiple locations, it can absorb and mitigate the impact of such attacks, ensuring that the content remains accessible to users.

To use Cloud CDN, you need to configure your GCP project and enable the service for your content. Once enabled, Cloud CDN will automatically cache your content and serve it from the nearest location to the user. This improves the user experience by reducing the time it takes to load the content.

In addition to caching static content, Cloud CDN also supports dynamic content caching. This means that it can cache content that is generated dynamically, such as personalized web pages or API responses. By caching dynamic content, Cloud CDN can further improve the performance of your website or application.

To monitor the performance of your content delivery, Cloud CDN provides detailed logs and metrics. These can help you identify any performance issues and optimize the delivery of your content.

Cloud CDN is a powerful service provided by Google Cloud Platform that improves the performance and availability of your content. By caching content in multiple locations and handling high traffic loads, it ensures that your content is delivered quickly and efficiently to users around the world.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP BASIC CONCEPTS
TOPIC: CLOUD OPERATIONS

Cloud Operations is a suite of products offered by Google Cloud Platform (GCP) that allows users to monitor, troubleshoot, and operate their services at scale. It is designed to enable DevOps, SREs, and IT operations teams to utilize Google's Site Reliability Engineering (SRE) best practices. Cloud Operations provides integrated capabilities for monitoring, logging, and advanced observability services.

One of the key components of Cloud Operations is Cloud Logging, which is a fully managed and highly scalable service. It aggregates log data from all infrastructure and applications into a single location. It automatically collects log data from Google Cloud Services, and users can also feed custom logs through Cloud Logging agent, open-source Fluentd, or the API. With Cloud Logging, users have complete control over how and where to store these logs, including options such as keeping them in Cloud Logging, exporting them to Cloud Storage for archival, BigQuery for analytics, or streaming them via Cloud Pub/Sub to a third-party destination. The Log Viewer tool provides powerful capabilities to filter logs, convert them to log-based metrics for monitoring, alerting, analyzing, and visualizing.

Another important component of Cloud Operations is Cloud Monitoring. This service provides observability across applications and infrastructure, regardless of whether they are on Google Cloud, on-premises, or on other clouds. Cloud Monitoring supports a variety of metrics integrations and allows users to define custom metrics unique to their use case. Users can analyze these metrics on the fly using the Metrics Explorer and Monitoring Query Language, identifying correlations and easily adding corresponding charts to a dashboard. Cloud Monitoring also provides out-of-the-box or custom-built dashboards to get a consolidated view of the health of infrastructure, services, or applications, making it easy to spot anomalies. Additionally, Cloud Monitoring offers alerting capabilities, allowing users to create policies to alert on performance metrics, uptime checks, and service-level indicators.

Cloud Operations also includes advanced observability features such as Trace, Debugger, and Profiler. Trace provides visualization and analysis to understand request flow, service topology, and latency issues in applications. Debugger allows users to inspect the state of running applications after deployment without needing to stop or slow them down. Profiler continuously analyzes code performance on each service, helping users improve speed and reduce costs. These features are designed to run in production with minimal performance impact. While Trace tracks relationships and latency between services, Profiler tracks this across individual functions in the code base, and Debugger helps find the root cause from method to the specific problematic piece of code.

Users can access Cloud Operations tools through the Cloud Console or the API. All these tools offer a generous free tier to make it easy for users to get started. From a security perspective, all data is encrypted at rest and in transit. Security-focused audit logs are automatically available in Cloud Logging, providing information about who did what, where, and when. Access Transparency captures the actions taken by Google personnel while offering support, ensuring compliance.

Cloud Operations is designed to help users keep their applications up and running and ensure customer satisfaction. It provides service-level objectives that work across all application types and cloud environments, as well as error reporting to identify bugs in applications. With Cloud Operations, ops teams have out-of-the-box observability to monitor infrastructure and applications.

To get started with Cloud Operations, users can check out the free trial.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP BASIC CONCEPTS
TOPIC: LOAD BALANCING

Cloud Load Balancing is a crucial concept in the field of Cloud Computing. It is a fully-distributed, software-defined solution that aims to balance user traffic to multiple backends, ensuring low latency and avoiding congestion. In this didactic material, we will explore the basic concepts of Cloud Load Balancing, specifically focusing on Google Cloud Platform (GCP).

There are different types of load balancing, depending on the type of traffic you are dealing with - global or regional. Let's understand these options with a use case. Imagine you have a user named Shen in California. You deploy your backend instances in that region and configure a load-balancing virtual IP. As your user base grows to another region, all you need to do is create instances in the additional regions. There is no need to change the virtual IP or the DNS service settings. This scalability allows your application to seamlessly handle increased traffic across different regions.

Cloud Load Balancing uses anycast virtual IPs, providing a single global frontend virtual IP address. It also offers cross-regional failover, fast autoscaling, and can handle millions of queries per second. This is known as external load balancing at layer 7.

In a three-tier application, after the frontend, you have the middleware and the data sources to interact with in order to fulfill a user's request. This is where layer 4 internal load balancing comes into play. Layer 4 internal load balancing is designed for TCP/UDP traffic behind RFC 1918 VIP, where the client IP is preserved. It leverages software-defined networking controls and data plane for load balancing.

Now let's dive into the data model for Cloud Load Balancing. For global HTTPS load balancing, you have global anycast virtual IPs (IPv4 or IPv6) associated with the forwarding rule. The forwarding rule directs traffic to a target proxy, which terminates the client's session. The URL map configured provides layer 7 routing and directs the client request to the appropriate backend service. Backend services can be managed instance groups or network endpoint groups for containerized workloads. This is also where service capacity and health is determined, and Cloud CDN can be enabled to cache content for improved performance. Firewall rules can be set up to control traffic to and from the backend.

Security is of paramount importance in load balancing. Google Cloud Platform offers best practices such as running SSL everywhere. With HTTPS and SSL proxy load balancing, you can use Google-managed certificates, where Google takes care of the provisioning and managing the SSL certificate lifecycle for you. Cloud Load Balancing also supports multiple SSL certificates if you want to serve multiple domains using the same load-balancing IP address and port. Additionally, Google's global load-balancing infrastructure absorbs and dissipates layer 3, 4 volumetric attacks. Cloud Armor can be used to protect against layer 3 to 7 application-level attacks, while Identity Aware Proxy and firewalls can authenticate and authorize access to your backends.

When choosing the right load-balancing option, consider factors such as internal versus external, global versus regional, and the type of traffic you are dealing with (HTTPS, TLS, UDP). Based on these factors, you can make an informed decision about which load-balancing option is right for your specific use case.

Cloud Load Balancing is a vital component of cloud computing, particularly in the Google Cloud Platform. It offers fully-distributed, software-defined solutions to balance user traffic across multiple backends, ensuring low latency and avoiding congestion. By understanding the different types of load balancing, the data model, and security considerations, you can make informed decisions to optimize performance, security, and cost for your backend systems.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP BASIC CONCEPTS
TOPIC: HIGH PERFORMANCE COMPUTING

High Performance Computing (HPC) is an aggregation of computing power used to solve complex problems that are either too large for standard computers or would take an excessive amount of time. It is also known as supercomputing. HPC enables the simulation or analysis of massive amounts of data that would otherwise be impossible with standard computers. However, a common challenge with HPC is that it often exceeds the capabilities of infrastructure resources, resulting in long wait times for results and slowing down research and innovation.

A high-performance computing system can be thought of as a cluster of computers, with each computer in the cluster referred to as a node. Each node in the cluster consists of an operating system, a processor with multiple cores, storage, and networking capabilities to facilitate communication between units. By utilizing a cluster with multiple nodes, problems can be solved much faster. For example, a smaller cluster may consist of 16 nodes with 64 cores (four cores per processor), significantly improving performance.

A supercomputer is a larger version of a cluster, capable of running HPC jobs across a massive number of cores in a short amount of time. For instance, a job that would take three months to run on an on-premises cluster could be completed in just 16 hours in the cloud, with little to no additional cost. By incorporating Google Cloud into the HPC environment, users can take advantage of economies of scale, gaining access to the largest compute and storage hardware, global presence, robust networking, and intelligent automated management capabilities.

Building an HPC environment on Google Cloud involves three key components: compute, storage, and networking. Compute Engine provides customizable virtual machines that can be scaled up or down as needed. Users can choose from a range of machine types, such as the compute-optimized C2 machines for most HPC applications, or the general-purpose N1, N2, or N2D machines for larger memory requirements. Custom machine types are also available for specific workload needs, ensuring optimal performance. Preemptable VMs are another cost-effective option for short-lived compute instances.

The storage system is crucial for the performance of many HPC applications. Google Cloud offers several storage options, including Cloud Storage for scalable object storage, Persistent Disk for durable and high-performance block storage, and Filestore for high-scale file sharing on Compute Engine VMs.

Networking is an essential aspect of HPC on Google Cloud. Google's privately managed global network infrastructure ensures that data and applications are secure and minimally exposed to the public internet. Users can utilize VPC networks to enable connectivity from Compute Engine VM instances and configure firewalls for applications. Placement policies allow users to control the placement of VMs in data centers, optimizing communication between nodes and reducing latency.

To set up an HPC workload on Google Cloud, users should first determine the compute, storage, and networking requirements for their code. They can then create an HPC cluster using Compute Engine instances connected to the desired storage option. Google Cloud supports various job schedulers, simplifying the process of autoscaling VMs based on job requirements or shutting down a cluster once a job is complete to save costs. Results can be visualized using tools like BigQuery or AI Platform for post-processing. Ongoing performance monitoring and cluster adjustments are essential for optimal results.

Security is a critical consideration for any HPC workload. Google Cloud's secure infrastructure provides advanced antimalware and threat detection to protect data, applications, and users.

High Performance Computing plays a vital role in driving research, development, and innovation across various industries. It is used for tasks such as rendering visual effects in movies, sequencing the human genome, risk analysis in financial services, and designing the next generation of cars.

To learn more about HPC on Google Cloud and get started, visit cloud.google.com/hpc.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP OVERVIEW
TOPIC: GCP COMPUTE ENGINE OVERVIEW

Google Cloud Platform (GCP) offers a wide range of compute products that cater to different types of workloads. In this overview, we will explore three key compute products: Google Compute Engine, Google Cloud Functions, and Google Kubernetes Engine.

Google Compute Engine allows you to create virtual machine (VM) instances from scratch. By specifying a region, machine type, operating system (OS) image, and other optional parameters, you can provision, start, and connect to a VM. During the configuration process, a cost estimate is provided. Noteworthy features include the ability to add GPUs or TPUs (Tensor Processing Units) to your instance and the availability of various OS images, including Linux distros and MS Windows. Additionally, you can use custom images if needed. Once the instance is created, you can SSH into it directly from your browser. You also have the option to create a new VM from a saved template or choose from pre-configured solutions available on the Marketplace. Compute Engine offers advanced features such as fine-grained security access control, HTTPS connectivity, live migration of running applications, and preemptible VMs.

Google Cloud Functions provides a serverless computing environment. With Cloud Functions, you can focus solely on your code while Google handles the underlying infrastructure. For example, you can write a simple snippet of code to listen to image file uploads into a storage bucket and automatically generate thumbnails for each image. Cloud Functions support various triggers, including database changes, pub/sub messages, and Compute Engine instance state changes. They can also be invoked via standard HTTP requests. Deploying Cloud Functions is easy, and they can be deployed in any region within a project. Integration with other GCP services and APIs is seamless, as authentication is automatically handled. One of the key advantages of Cloud Functions is that you only pay for the code that is running, making it cost-effective for temporal workloads.

For larger-scale applications, Google App Engine offers serverless benefits with more developer configurations. App Engine allows you to scale your applications on-demand and provides features such as services, versioning, and traffic splitting. It is an excellent choice if you need more flexibility than Cloud Functions but still want the benefits of serverless computing.

Containers are an essential part of cloud computing, and Google Kubernetes Engine (GKE) simplifies container deployment. GKE is a fully-managed version of Kubernetes, an open-source container orchestration system. With GKE, you can deploy containerized applications with ease. It guarantees uptime, provides rich dashboard metrics, and automates operations from auto-scaling to node repairs and Kubernetes version upgrades. You can describe the compute, memory, and storage resources your application containers require, and GKE will provision and manage the underlying cloud resources automatically. GKE supports persistent storage, allowing you to run stateful workloads such as databases. Moreover, your Kubernetes workloads are portable across different Kubernetes implementations, from your local development environment to GKE or other cloud/on-premises installations.

GCP offers a comprehensive suite of compute products. Google Compute Engine is ideal for running Linux and Windows applications, while Google Cloud Functions provides a serverless environment for code-focused development. Google App Engine offers serverless benefits with more developer configurations, and Google Kubernetes Engine simplifies container deployment and management. Whether you need VM instances, serverless functions, or container orchestration, GCP provides a fast and reliable underlying infrastructure.

Google Cloud Platform (GCP) offers a range of products and services for various cloud computing needs. In this overview, we will focus on GCP Compute Engine. Compute Engine is a virtual machine (VM) service that allows users to run their applications on Google's infrastructure.

With Compute Engine, users can create and manage VM instances, which are virtual representations of computers that run on Google's data centers. These instances can be customized to meet specific requirements, such as choosing the operating system, machine type, and storage options. Compute Engine offers a variety of machine types, ranging from general-purpose to high-performance options, allowing users to select the most suitable configuration for their workloads.

One of the key advantages of Compute Engine is its scalability. Users can easily scale their resources up or down based on demand, allowing for efficient resource utilization and cost optimization. Additionally, Compute Engine provides automatic load balancing, ensuring that traffic is distributed evenly across multiple instances, improving performance and reliability.

Compute Engine also offers various networking capabilities. Users can create virtual private clouds (VPCs) to isolate their resources and control network access. They can also set up firewall rules to manage incoming and outgoing traffic. Additionally, Compute Engine integrates with other GCP services, such as Cloud Storage, BigQuery, and Cloud SQL, enabling seamless data transfer and processing.

To get started with Compute Engine, users can use the Google Cloud Console, which provides a web-based interface for managing resources. Alternatively, they can use the command-line interface (CLI) or APIs for programmatic control. Compute Engine also supports integration with popular DevOps tools, such as Jenkins and Kubernetes, facilitating continuous integration and deployment workflows.

GCP Compute Engine is a powerful and flexible service that allows users to run their applications on Google's infrastructure. With its scalability, networking capabilities, and integration with other GCP services, Compute Engine provides a comprehensive solution for cloud computing needs.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP OVERVIEW
TOPIC: GCP MACHINE LEARNING OVERVIEW

Google Cloud Platform (GCP) offers a range of machine-learning products that simplify the process of building, maintaining, and deploying machine learning models. With GCP, developers can leverage high-quality pre-trained models via APIs, which can be easily integrated into applications regardless of the programming language used.

One of the key APIs provided by GCP is the Vision API. By invoking this API, developers can analyze images and extract various information such as labels, dominant colors, and text. The Vision API can also identify entities like landmarks, celebrities, logos, and news events. Additionally, it offers content moderation capabilities for user-generated content.

GCP also provides pre-trained models for text-to-speech, speech-to-text, natural language processing, and translation. These models can be accessed through user-friendly APIs, allowing developers to incorporate advanced language processing capabilities into their applications.

While pre-trained models are convenient, they may not always meet specific business needs. This is where Cloud AutoML comes in. Cloud AutoML enables developers with limited machine learning expertise to train high-quality models using their own data. Leveraging Google's transfer learning and neural architecture search technology, Cloud AutoML offers a simple graphical user interface for training, evaluating, improving, and deploying custom machine-learning models.

For more advanced use cases, GCP offers the Cloud Deep Learning VM Image. These pre-configured Google Compute Engine instances come with the latest versions of popular machine learning frameworks such as TensorFlow, PyTorch, and scikit-learn. With a single click, developers can add cloud CPU and GPU support, making it easier to train models using their own data sets.

To simplify the training and prediction process, GCP provides Cloud ML Engine. This fully managed service allows developers and data scientists to build models using frameworks like scikit-learn, XGBoost, Keras, and TensorFlow. Cloud ML Engine can train models at large scale on a managed cluster, and it includes a unique feature called HyperTune, which automatically tunes deep-learning hyperparameters to achieve better results faster. The service also offers online predictions through a secure web endpoint, adjusting to the request rate of ML-enabled applications.

Whether you are new to machine learning or an expert, GCP offers a range of tools to suit your needs. From pre-trained models accessible via simple API calls, to customizing models with Cloud AutoML, to training and serving custom TensorFlow models using Compute Engine or Cloud ML Engine, GCP provides a comprehensive set of solutions for machine learning tasks.

To further explore these products, you can take advantage of the free codelabs provided by GCP. Additionally, you can refer to the compute and big data overview episodes for a deeper understanding of GCP's capabilities. Stay tuned for an upcoming video on big data storage and processing.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP OVERVIEW
TOPIC: GCP SERVERLESS OVERVIEW

Serverless computing is a popular approach in cloud computing that allows developers to focus on the business logic of their applications without having to worry about managing servers or infrastructure. Google Cloud Platform (GCP) offers various serverless products to help developers achieve this.

One such product is Functions as a Service (FaaS), which allows developers to write a piece of code, known as a function, that is triggered by an event or an incoming HTTP request. This function can interact with databases and other services before generating another event or sending a response. Google Cloud Functions supports multiple programming languages, such as Python, Node.js, Go, Java, and more. Developers can deploy their functions along with their dependencies directly in the cloud and configure the event that triggers their execution. These events can include HTTP requests, file uploads, database changes, messages posted to a queue, and more. Functions can also be assigned deploy-time environment variables, deployed to multiple regions, and configured with specific security constraints.

Cloud Functions are billed based on the number of invocations, compute time, and outgoing network usage. The first 2 million invocations every month are free. This makes Cloud Functions an easy and cost-effective way to access various powerful GCP services, such as machine learning APIs and storage solutions, enabling the implementation of anything from glue code to fully-fledged microservices-based applications.

For developers who desire more freedom in the languages and frameworks they use, as well as the ability to deploy Docker container images instead of source code, Cloud Run is a suitable option. Cloud Run offers a true serverless experience for stateless HTTP container images. Developers can build their container image, upload it to Cloud Registry, and create a Cloud Run service using that container. Cloud Run takes care of provisioning and managing servers, automatically scaling up and down based on incoming traffic and even scaling down to zero. Developers only pay for the resources their app uses, down to the nearest 100th millisecond. Additionally, Cloud Run can be used with Kubernetes Engine clusters, providing the same easy experience and benefits.

Whether running on GKE or not, Cloud Run supports deploying multiple services in a single GCP project, either in multiple regions or in specific namespaces when running in a GKE cluster. Each service has a unique endpoint, and each deployment creates a revision. Requests are automatically routed to the latest healthy service revision. Each revision scales to handle incoming requests, and the concurrency setting allows developers to set the maximum number of parallel requests a container instance can handle. Cloud Run combines the flexibility of modern container-based development with the benefits of a fully serverless environment, ensuring autoscaling to meet application needs.

For developers looking to deploy source code while preserving serverless benefits, Google App Engine is a managed platform within GCP that has been around for over 10 years. App Engine allows developers to choose their preferred programming language and deploy their applications using the "gcloud app deploy" command.

Google Cloud Platform offers a range of serverless products, including Functions as a Service (FaaS), Cloud Run, and Google App Engine. These products allow developers to focus on their application's business logic without the need to manage servers or infrastructure. Each product provides different levels of flexibility and ease of use, catering to various developer preferences and application requirements.

Google Cloud Platform (GCP) offers various serverless solutions for cloud computing. One of these solutions is the second-generation App Engine runtimes, which provide an idiomatic experience for developers. These runtimes support multiple languages, such as Java, Node, PHP, Go, and Python, and allow the use of any language API and framework. With read/write file system access and isolation provided by gVisor, App Engine offers a powerful open-source sandbox technology.

App Engine allows developers to build applications using multiple services, each of which can use different languages and be scaled independently. Additionally, each service can have multiple versions active at the same time. This makes it easy to set up staged rollouts or A/B testing across different versions with traffic splitting.

App Engine also provides out-of-the-box tooling for app performance management. Developers can perform live debugging of production apps, trace requests flowing across the system, and even profile the CPU and heap of their app. With these features, Google App Engine is a mature serverless platform that offers advanced capabilities for building modern applications.

In addition to App Engine, developers can leverage Cloud Functions and Cloud Run to build more advanced applications. Cloud Pub/Sub and Cloud Tasks are popular solutions for integrating different parts of an application or combining multiple functions. Cloud Pub/Sub is a simple, reliable, and scalable event system that supports many-to-many asynchronous messaging. It offers at-least-once delivery and is globally available without the need to manage infrastructure. Developers can publish and consume hundreds of millions of messages per second.

Cloud Tasks, on the other hand, provides a dispatch system for managing the execution of large numbers of distributed tasks. It is ideal for one-to-one asynchronous messaging and comes with rate limit controls. Lastly, Cloud Scheduler is a fully managed cron job service that allows developers to schedule tasks invoked through HTTPS endpoints, Cloud Pub/Sub topics, or App Engine applications. It is remarkably simple to use yet incredibly powerful.

All of these serverless solutions provided by GCP are fully managed and monitored. Logging and error reporting are built-in features, making it easier for developers to manage their applications. With GCP serverless, developers can submit their code in the form of a function, an application, or a container image, and Google will handle the execution.

To learn more about these serverless solutions and explore GCP products, you can take free codelabs linked in the description. Stay tuned for upcoming episodes and overviews. If you found this material helpful, please like, subscribe, comment, and share. We look forward to bringing you more "GCP Essentials" content.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP OVERVIEW
TOPIC: GCP DATA AND STORAGE OVERVIEW

Google Cloud Platform (GCP) offers a range of data processing and storage products that provide efficient and seamless solutions for storing and analyzing data. One of the key services is Google Cloud Storage, a fully-managed object storage service that allows users to store objects or files. It offers different storage classes, including multi-regional, for optimal user latency, and coldline storage for lower frequency access. With a single unified API, users can easily interact with cloud storage and seamlessly move objects across storage classes. Cloud storage also provides strong consistency and is designed for 11 lines of durability.

In the realm of big data, GCP offers BigQuery, a powerful tool for processing large datasets. With BigQuery, users can run SQL queries on massive amounts of data to gain valuable insights. Users can upload their own datasets or use sample data from public datasets. The beauty of BigQuery is that users do not need to provision a cluster or storage capacity. Simply crafting a standard SQL query allows BigQuery to process potentially gigantic amounts of data within seconds and produce results that can be saved in various formats. For example, using the public data set of GitHub commits, users can ask BigQuery to provide a list of the top Google employees fixing issues on GitHub or determine which programming languages make developers the most happy.

To import or stream data to BigQuery, users can utilize an API or export data produced by other GCP products and services, such as logs to BigQuery. For users with existing Apache Hadoop or Spark workloads, Cloud Dataproc is a fully-managed environment that can spin up a cluster in less than 90 seconds, providing a seamless transition to GCP.

Google Cloud also offers solutions for relational databases. Cloud SQL is a managed service that supports popular relational databases such as MySQL and PostgreSQL. Users can choose between these databases, select a machine size with ample RAM and storage, and Cloud SQL takes care of encryption at rest and in transit, private IP addresses, data replication between multiple zones with automatic failover, automated backups, and point-in-time recovery. Additionally, Cloud Spanner is a horizontally scalable, strongly consistent relational database as a service, which defies the CAP theorem and is the foundation for many Google services.

In the NoSQL department, Cloud Firestore is a highly scalable, strongly consistent database that offers real-time updates and offline support for mobile developers in native mode, as well as a datastore mode for backend developers looking for a schema-less documents database. Cloud Bigtable is a petabyte-scale, fully managed, noSQL database service that is ideal for large analytical and operational workloads, providing high throughput and consistent sub-10 millisecond latency.

Apart from these services, GCP offers other data-related products that are worth considering. Pub/Sub allows for producing and consuming messages globally across all GCP zones and regions. Filestore is designed for applications that require a system interface. Cloud Memorystore for Redis is a fully managed, in-memory data storage service that is ideal for building application caches with sub-millisecond data access. Cloud Dataflow enables users to build and execute unified batch and streaming pipelines using the Apache Beam programming model. Dataprep helps data scientists spend less time cleaning up data and more time processing it. Finally, Data Studio provides interactive dashboards and engaging reports.

To explore any of these GCP products, users can take advantage of the free codelabs provided by Google. Additionally, they can check out the compute overview and machine learning episodes, and look forward to an upcoming video on DevOps and tooling.

Google Cloud Platform offers a comprehensive suite of data processing and storage products that cater to various needs and use cases. From object storage to big data analysis, relational and NoSQL databases, and other innovative data-related services, GCP provides scalable, efficient, and fully-managed solutions for businesses and developers.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP OVERVIEW
TOPIC: GCP HANDS-ON

If you've seen previous GCP Essentials material, you should now have a good understanding of what GCP (Google Cloud Platform) has to offer. In this didactic material, we will discuss the various ways you can experience GCP through hands-on code labs, tutorials, online courses, and solutions, all at little or no cost.

To start your GCP journey, visit the Google Cloud home page at cloud.google.com. This is where you can find product documentation and spend time learning about specific products and APIs. Make sure to check out the Documentation Quick Start, which offers step-by-step tutorials on tasks like creating a Linux VM, storing and sharing files, deploying Docker container images, running label detection on photographs, and deploying applications on App Engine. These tutorials provide a great introduction to GCP.

Interactive tutorials are also available directly in the GCP console. You can select from a variety of product tutorials and follow the step-by-step instructions to navigate the console user interface. This hands-on experience will help you become familiar with GCP's features and functionalities.

For self-paced hands-on experience, Google code labs are available at g.co/codelabs. These labs cover a wide range of GCP products, with over 200 free labs to choose from. You can sort the labs by category, duration, and publish dates. Each lab typically costs $1 or $2, which can be covered by the $300 free trial credits. It is important to follow the CodeLab's clean-up instructions to properly delete any resources that may incur costs.

Another option for hands-on learning is Qwiklabs. If you are preparing for a certification or want to take a series of related labs called Quests, Qwiklabs provides a personalized platform to track your progress. Simply sign up for a free account, purchase credits, and start your quest. Each lab in Qwiklabs comes with a temporary GCP account and project ID for the duration of the lab. Qwiklabs is also used by Coursera to deliver its GCP courses and specializations.

While it is valuable to gain hands-on experience with individual GCP products, understanding how these products work together is equally important. This is where Solutions come into play. Solutions are detailed technical articles that often include source code examples. They are grouped into categories like modernize infrastructure, migrate workloads, hybrid cloud, HPC, big data analytics, machine learning, IoT, continuous delivery, serverless API management, and more. You can also browse solutions by industry or vertical, such as retail, energy, financial services, gaming, and others. These solutions are based on customer best practices and common use cases, providing you with insights and code to accelerate your GCP development.

There are plenty of hands-on materials available to help you explore and learn GCP. Whether you choose tutorials, code labs, or solutions, the resources are waiting for you to dive in and get your hands dirty. So don't hesitate, start your GCP learning journey today!

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP OVERVIEW
TOPIC: GCP CONTINUOUS LEARNING

Having the right resources of information and using the right communication channels can be the difference between a regular and a highly productive Google Cloud Platform (GCP) user. In this material, we will explore various prominent resources and less obvious ones that can help you in your GCP journey.

It all starts with the Google Cloud homepage at cloud.google.com. This page serves as a reference for product descriptions and documentation. Additionally, there are landing pages for the most popular languages supported by GCP, such as Go, Python, Java, Node.js, PHP, .NET, Ruby, and Kotlin. These landing pages provide language-specific guidance on deploying web apps, using GCP's APIs and libraries, and more.

Blog posts are another valuable resource for staying up to date with GCP. The main Google Cloud Blog at cloud.google.com/blog covers a wide range of topics, including product updates, features, partner and customer stories. Additionally, the GCP Medium publication features articles curated by practitioners for practitioners. It includes a weekly recap of GCP news and provides a platform for sharing experiences with the community.

For those who prefer audio content, there are two weekly podcasts worth subscribing to - the GCP podcast and the Kubernetes podcast. These podcasts cover news, interviews with Google engineers, partners, customers, and community members.

The GCP YouTube channel offers a wealth of video content, regularly publishing videos grouped into playlists based on topics, products, and events. Other related YouTube channels include Firebase, TensorFlow, Google Developers, G Suite, and Apigee.

Social media platforms like Twitter, Facebook, and LinkedIn also play a role in the GCP community. Following Google Cloud Platform @GCPCloud and other active accounts like Firebase, G Suite Developers, Google Maps Platform, Google Open Source, and Apigee can provide a way to engage with the GCP community and get the team's attention.

The GCP community extends beyond online platforms. User groups and GCP meet-ups are great opportunities to meet like-minded individuals, share experiences, best practices, and even find job opportunities. If there isn't a meet-up group nearby, creating one is encouraged.

Attending conferences, such as Google Cloud Next, Summits, or industry events like OSCON, KubeCon, and Devoxx, provides opportunities to learn from technical sessions, meet Cloud engineers, and ask questions. Additionally, there are over 600 community-led DevFest events where GCP content can be found.

When seeking help, peers can be a valuable asset. Stack Overflow is a popular platform for getting answers, and Google engineers actively maintain and monitor GCP-related tags on the platform. Formal support options are also available, with different levels of support depending on your needs.

By leveraging the resources mentioned in this material, you can enhance your GCP learning experience, stay up to date with the latest developments, connect with the community, and get the support you need.

In order to support your learning journey with Google Cloud Platform (GCP), there are several resources and tools available to enhance your understanding and optimize your experience.

One valuable resource is the Support section within the GCP console. By utilizing a portion of your $300 trial credit, you can access support directly from the console. This feature allows you to address any questions or concerns you may have while exploring GCP.

Visual representation is often an effective way to communicate complex concepts. Cloud.google.com/icons provides a collection of visually appealing icons and diagrams that can be used to create GCP architecture representations. These resources include vector graphics, PNGs, slides, and templates for popular tools like Lucidchart and Draw.io. By utilizing these icons and diagrams, you can effectively communicate your GCP architecture to your colleagues.

For quick reference and easy access to important information, the Google Cloud Four Words or Less Cheat Sheet is a valuable tool. This cheat sheet is available in various formats, including print-friendly versions, and is maintained on GitHub. It is a popular resource among GCP users and provides concise information that can be easily understood and applied.

In addition to these resources, it is important to explore and utilize the various options available to you within GCP. Choose the resources and tools that align with your learning style and preferences. If you come across any missing or desired resources, feel free to share your feedback in the comments section.

Remember to engage with the GCP Essentials material by liking, subscribing, commenting, and sharing. Stay tuned for more informative videos to further enhance your GCP knowledge.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP OVERVIEW
TOPIC: RUNNING CONTAINERS ON GCP

Containers have become a popular choice for developers due to their ability to package applications and their dependencies into portable and easy-to-move packages. Google Cloud Platform (GCP) offers three ways to run containers: Google Kubernetes Engine (GKE), Cloud Run, and Google Compute Engine (GCE).

GKE is a fully managed Kubernetes service provided by Google. It takes care of scheduling, scaling, and monitoring containers, making it easy to deploy code to production. GKE clusters are secure, highly available, and run on Google Cloud's high-speed network. They can be fine-tuned for specific locations and machine types, including optional GPUs or TPUs. GKE is also a key component of Anthos, Google Cloud's enterprise hybrid and multi-cloud platform, allowing migration of existing VMs into containers and seamless workload movement between on-premises and cloud environments.

Cloud Run combines the benefits of containers and serverless computing. It eliminates the need to provision or manage infrastructure, automatically scaling stateless containers. Creating a Cloud Run service only requires selecting a location, giving it a name, and setting authentication requirements. Cloud Run supports multiple requests per container and works with any language, library, binary, or Docker image. It offers true pay-for-usage, the ability to scale to zero, and out-of-the-box monitoring, logging, and error reporting. Cloud Run is built using the Knative open-source project, enabling private hosting environments and deployment on Cloud Run for Anthos or GCP.

GCE allows running containers within a familiar virtual machine environment. It leverages existing workflows and tools without requiring extensive knowledge of cloud-native technologies. When creating a GCE virtual machine, the container section allows specifying the image to use. The recommended option is the Container-Optimized OS, an operating system optimized for running Docker containers and maintained by Google. This OS image comes with a pre-installed Docker Runtime, ensuring a secure container runtime with a smaller attack surface. GCE supports scalable services using managed instance groups, offering auto scaling, auto healing, rolling updates, multi-zone deployment, and load balancing for compute instances.

Container images can be stored in Google Container Registry (GCR), a private-by-default container registry running on GCP. GCR provides consistent uptime across multiple regions and allows pushing, pulling, and managing container images from various systems, including VM instances and personal hardware. Access to images can be controlled, ensuring only authorized users can view and download them. Container Registry enables convenient deployment to all three runtimes discussed: Cloud Run, Container Engine (GKE), and Compute Engine (GCE).

Google Cloud Platform offers multiple options for running containers. GKE provides a fully managed Kubernetes service, Cloud Run combines containers and serverless computing, and GCE allows running containers within familiar virtual machine environments. Google Container Registry ensures secure and controlled storage of container images.

Container Registry is a feature within Google Cloud Platform (GCP) that allows for the automatic building of containers based on code or tag changes to a repository. It integrates seamlessly with popular continuous delivery systems like Cloud Build, Spinnaker, or Jenkins. By scanning the stored images in the registry, Container Analysis identifies any known vulnerabilities, providing you with the necessary information to review and address these issues before deployment.

When it comes to running containers on GCP, Google Cloud offers three robust options. The first is a fully managed Kubernetes environment, which provides a scalable and reliable solution for deploying and managing containerized applications. Kubernetes automates many aspects of container orchestration, making it easier to handle complex deployments.

The second option is a serverless platform, which allows you to focus solely on your application code without worrying about infrastructure management. With serverless computing, GCP takes care of scaling and resource allocation, enabling you to run your containers in a highly efficient and cost-effective manner.

Lastly, GCP provides a range of free code labs that you can try to explore these containerization products further. These code labs offer hands-on experience and guidance, allowing you to get familiar with the different features and functionalities available.

Container Registry, in conjunction with GCP's various containerization options, provides a comprehensive solution for running containers. Whether you prefer a managed Kubernetes environment or a serverless platform, Google Cloud has you covered. By leveraging these tools, you can deploy and manage your containerized workloads with ease.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP OVERVIEW
TOPIC: GCP AND FIREBASE WITH PROJECTS AND STORAGE

Firebase is a platform developed by Google for mobile and web app development. It provides a range of cloud services that enhance client apps, including authentication, analytics, crash reporting, A-B testing, and in-app messaging. However, Firebase also offers options for managing data and business logic in the cloud, enabling developers to build better apps. These options include a NoSQL database, serverless functions, machine learning APIs, and blob storage.

One important aspect to note is the relationship between Firebase and Google Cloud Platform (GCP). When you create a Firebase project, it is essentially a GCP project in every aspect. This means that resource grouping, identity management, and billing are all the same. Firebase allows Android and web developers to leverage Google Cloud services without having to deal with the complexities of GCP. It provides a way to start using cloud services before transitioning to GCP when necessary. This is also beneficial for users looking to build mobile or web apps on top of an existing GCP infrastructure.

Although the Firebase console and cloud console have different interfaces, they can both be used to access the same project. If you have an existing GCP project, you can open it in the Firebase console to add Firebase-specific functionality. Similarly, if you have an existing Firebase project, you can open it in the cloud console with its identifier and manipulate all project resources.

It is important to exercise caution when deleting projects, as deleting a Firebase project also deletes the associated Google Cloud project and all its resources.

Now let's focus on the three main products that Firebase and GCP have in common: Cloud Storage, Cloud Functions, and Cloud Firestore.

Cloud Storage is a highly scalable blob storage system. It is simple to use and offers powerful features. Firebase developers often use Cloud Storage for managing user-generated content, such as images. The Firebase SDKs for Android, iOS, Web, Unity, and C++ make it easy and secure to upload and download objects directly from the app. Each new Firebase project comes with a default Cloud Storage bucket, which is commonly used by Firebase developers and does not require explicit referencing.

Data stored in Cloud Storage using Firebase can be accessed and processed in GCP, and vice versa. For example, a Firebase-powered mobile application can upload pictures to Cloud Storage, and a Cloud Scheduler-initiated task can manipulate those pictures in various ways. The files can also be used with other GCP big data products, such as BigQuery, Dataflow, and machine learning products.

It is important to note that bucket access control is different and orthogonal. Cloud IAM is used to control access to buckets and objects from GCP services, while Firebase security rules control access only from mobile applications that use the Firebase SDKs.

In the next episode, we will discuss the remaining two products that are common to both Firebase and GCP: Cloud Functions and Cloud Firestore.

Firebase and GCP have a lot in common and are designed to complement each other. Firebase allows developers to leverage Google Cloud services while providing a simplified interface and functionality specifically tailored for mobile and web app development.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP OVERVIEW
TOPIC: GCP AND FIREBASE WITH FUNCTIONS AND FIRESTORE

Cloud Computing - Google Cloud Platform (GCP) Overview - GCP and Firebase with Functions and Firestore

In this didactic material, we will explore two important products offered by Google Cloud Platform (GCP) and Firebase - Cloud Functions and Cloud Firestore.

Cloud Functions is a Google Cloud product that is also accessible through Firebase. It is an implementation of the Functions as a Service (FaaS) paradigm, where events in a system trigger the execution of small pieces of code known as functions. These events can include HTTP calls or file uploads to a bucket. Cloud Functions can be invoked via the Firebase SDK with user tokens and device instance IDs directly propagated to the functions. It is important to note that Firebase offers the option to wrap functions into callable functions, which can be invoked with user tokens and device instance IDs. The supported languages for Cloud Functions are Node.js, TypeScript, Go, Python, and Java.

To deploy functions, you can use either the Firebase CLI or the GCP command line interface, G Cloud. Both tools have their own functionalities, and it is recommended to choose the one that works best for you and stick to it. With Firebase, you need to install Firebase command line tools using NPM, as deploying from the Firebase console is not possible. On the other hand, GCP allows you to deploy functions directly from the console, integrating with GCP's private source repositories or typing the function source code in line. Firebase command line tools provide an API for strongly typed handling of trigger events and the ability to deploy multiple functions at once. Cloud Functions created with a Firebase project can also be managed using the Cloud Console, which offers additional features such as monitoring graphs, a tab for function testing, and the ability to set features like retry on failure, memory allocation, and timeouts.

Cloud Firestore, on the other hand, is Google's state-of-the-art NoSQL document database. It is schema-less and allows you to store documents containing attributes in a hierarchy of collections. Cloud Firestore comes in two flavors - datastore mode and native mode. For Firebase and GCP users, the native mode is the common choice. It can automatically scale to millions of concurrent clients and offers near real-time notifications, enabling synchronization of data across devices. It also has built-in offline support, allowing access and changes to data even when the client is offline. Data stored in Cloud Firestore can be accessed using Firebase SDKs, and both the Firebase and GCP consoles can be used to view, edit data, and monitor database access usage. With Cloud Firestore, you can query the database directly from your mobile or web clients without the need for an intermediary server. Therefore, the Firebase console has an additional tab for security access roles. If you are a Firebase user shipping an app that accesses Firestore, it is recommended to use Firebase Authentication and carefully consider security access rules. On the GCP side, Firestore is typically accessed using a service account, and there are no equivalent security rules. Server-side code is considered trusted, while client code from mobile apps is not.

Supported languages for Firestore SDKs include Python, Node.js, Java, C#, .NET, Go, PHP, and Ruby. Firebase mobile SDKs also include web, Android, and iOS support. These mobile SDKs include local caching as a unique feature to help implement offline capability. It is important to note that GCP and Firebase share a common project and billing infrastructure, as well as common services such as Cloud Storage, Cloud Functions, and Cloud Firestore.

Cloud Functions and Cloud Firestore are powerful tools offered by Google Cloud Platform and Firebase. Cloud Functions allow the execution of small pieces of code in response to various events, while Cloud Firestore is a state-of-the-art NoSQL document database that offers real-time synchronization and offline support. Both platforms provide various SDKs and console functionalities to manage and access data.

Cloud Computing - Google Cloud Platform (GCP) Overview - GCP and Firebase with Functions and Firestore

Google Cloud Platform (GCP) offers a wide range of services for cloud computing, including Firebase. Whether you are a beginner or an experienced user, understanding the capabilities and possibilities of GCP is essential. In this didactic material, we will provide an overview of GCP and its integration with Firebase, focusing on functions and Firestore.

GCP provides a comprehensive suite of cloud services that enable organizations to build, deploy, and scale applications and services. It offers infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS) solutions. With GCP, users can access a variety of computing resources, storage options, and development tools.

Firebase, a mobile and web application development platform, is seamlessly integrated with GCP. It offers a set of features and services that simplify the development process, including authentication, real-time database, storage, and hosting. Firebase allows developers to build high-quality apps quickly and efficiently.

One of the key features of Firebase is Cloud Functions, which enables developers to run serverless code in response to events. With Cloud Functions, you can write code snippets that are executed in the cloud, triggered by events such as database changes, user authentication, or HTTP requests. This allows for the creation of dynamic and scalable applications without the need for managing servers.

Firestore is a NoSQL document database offered by Firebase. It provides a flexible and scalable solution for storing and querying data. Firestore organizes data in collections and documents, allowing for efficient retrieval and manipulation. It supports real-time updates, enabling applications to respond to changes in data instantly.

By combining GCP and Firebase, users can leverage the power of both platforms to build robust and scalable applications. GCP offers a wide range of services, while Firebase provides a streamlined development experience. Together, they enable developers to create innovative solutions with ease.

Google Cloud Platform (GCP) offers a comprehensive set of cloud services, and Firebase provides a powerful development platform. By integrating GCP and Firebase, developers can take advantage of functions and Firestore to build dynamic and scalable applications. Understanding the capabilities of GCP and its integration with Firebase is crucial for anyone looking to leverage these technologies.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP OVERVIEW
TOPIC: GCP LOGGING

One of the key features that makes Google Cloud production ready and ops-friendly is its wide range of management, monitoring, and alerting tools. While these tools are not a substitute for DevOps or SRE practices, they play a crucial role in ensuring the smooth operation of Google Cloud. In this material, we will focus on one specific tool: Cloud Logging.

Cloud Logging is a fully managed service that allows users to collect, read, and parse logs across a distributed infrastructure involving multiple Google Cloud products. It provides search, monitoring, and alerting capabilities, making it easier for users to analyze and manage log data. Additionally, Cloud Logging comes with an API that enables the ingestion of custom log data from any source.

One of the advantages of Cloud Logging is its ease of use. As a fully managed service, there is no need to provision hard drives or resize partitions. It can handle the ingestion of application and system log data from thousands of sources simultaneously. Furthermore, Cloud Logging allows users to analyze log data in real-time without the need to synchronize server pods or manage time zones.

Logs in Cloud Logging are composed of entries created by various sources, including Google Cloud Services, third-party applications, and user code. Each log entry contains a payload, which can be a simple string or structured data. Examples of log entries include details of a compute engine instance starting up, a new file being uploaded to a bucket, or a call made to a machine learning API. The name of the monitored resource is included in each log entry, indicating its origin.

To view and query logging data, users can utilize the Logs Viewer in the console. The Logs Viewer enables users to search for log entries based on the resource, log level, and timestamp. Alternatively, these queries can also be accessed through the logging API or the command line. It is worth noting that logs are stored for free up to the first gigabyte for every project. After that, there is a charge of $0.50 per additional gigabyte. Users can also set up alerting policies based on log ingestion limits.

In addition to the Logs Viewer and logging API, Cloud Logging allows users to export logs to other storage systems such as Cloud Storage, BigQuery, or Pub/Sub. This feature is useful for archival purposes or for advanced analytics.

Cloud Logging is a powerful tool that enables users to collect, analyze, and manage logs across the Google Cloud platform. It provides search, monitoring, and alerting capabilities, and supports real-time analysis of log data. With its ease of use and integration with other Google Cloud services, Cloud Logging is an essential component for managing and monitoring a distributed infrastructure.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP OVERVIEW
TOPIC: GCP ERROR REPORTING

Error reporting is an essential tool in cloud computing to capture and manage errors efficiently. Google Cloud Platform (GCP) offers a powerful error reporting tool that helps users identify and categorize errors in their applications. In this didactic material, we will explore GCP error reporting and its features.

When developing applications, errors can occur at the application level, even in production. It can be time-consuming and frustrating to manually search through logs to find relevant error information. GCP's error reporting tool simplifies this process by aggregating and displaying errors produced in running Cloud Services. It automatically groups error and critical level errors from your application and can notify you when a new error group appears.

GCP error reporting supports multiple programming languages and is supported out-of-the-box by various GCP services such as Cloud Functions, App Engine, Cloud Run, Compute Engine, and GKE. Any application errors that use basic formatting or call the error reporting API can be surfaced using this tool.

Errors are grouped and de-duplicated by analyzing their stack traces. This means that you will only see one entry per error type, making it easier to identify and manage errors. Each error entry provides a summary that includes information on when the application started producing the error, how often it occurred, and when it last occurred.

One useful feature of GCP error reporting is the ability to set a resolution status for each error. By default, errors are marked as "open," but you can change the status to "acknowledged," "resolved," or "muted." Additionally, errors can be linked to an issue in a bug tracking system, enabling efficient collaboration and issue resolution.

One of the advantages of GCP error reporting is its seamless integration with Google Cloud's serverless products. There is zero setup required for serverless products, and for other products, the setup process is straightforward.

To ensure developers are promptly notified of errors, GCP error reporting offers real-time notifications via email or the Google Cloud mobile app. This allows developers to stay updated on error occurrences and take immediate action.

In upcoming episodes, we will explore further tools available in GCP to better understand and resolve errors. Additionally, we will introduce a unique solution using debug log points to address the challenge of restarting or redeploying applications for incorporating new logging statements.

If you found this material helpful, please like, subscribe, comment, and share. Stay tuned for more educational content on Google Cloud Essentials.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP OVERVIEW
TOPIC: GCP DEBUGGING

Google Cloud Platform (GCP) provides developers with a range of infrastructure and tools to aid in the development and management of applications and services. In this material, we will explore the capabilities of GCP for tracing, profiling, and debugging in production environments.

Cloud Trace is a distributed tracing system offered by GCP. It allows developers to collect latency data across multiple Google Cloud products and supported languages. With Cloud Trace, you can easily identify where time is spent during the execution of specific tasks in your application. The system provides a graphical interface in the console that displays recent traces, including their URIs, latency, and timestamps. Additionally, it offers insights into common application problems and presents a heat map of request duration over time, aiding in the identification of requests that require further investigation. Cloud Trace requires minimal setup and is seamlessly integrated with most applications.

OpenTelemetry, formerly known as OpenCensus, is a project developed by Google to simplify the process of capturing distributed traces from applications that are not natively integrated with Cloud Trace. OpenTelemetry provides libraries that can automatically capture traces and application metrics from your applications. It also offers an API for manual instrumentation. This project aims to make tracing in GCP more accessible and user-friendly.

When you identify a bottleneck in your application that requires code modification, Cloud Profiler comes into play. Cloud Profiler allows you to gather CPU and memory allocation data from your production applications with little to no overhead. This data is then attributed back to the application source code, helping you identify resource-consuming areas and optimize performance. Profiling data is displayed using flame graphs, which provide a compact and readable representation of the collected information. The Cloud Profiler user interface allows you to interact with the data, such as accessing the Top Functions list to identify the most expensive functions and enabling the focused view for a more detailed analysis. This tool empowers developers to improve the efficiency of their applications.

Cloud Debugger is a powerful tool for inspecting the state of live-running applications without stopping or slowing them down. It offers two unique features: snapshots and logpoints. With snapshots, you can capture the call stack and inspect local variables, providing valuable insights into the application's execution. Logpoints enable you to inject logging statements without the need to restart the application. By adding logpoints at specific code locations, you can gather relevant information for debugging purposes. These logpoints are written to the standard output and can be used with any logging backend. Cloud Debugger is particularly useful for troubleshooting hard-to-reproduce issues and eliminating the need for frequent application restarts or redeployments.

GCP offers a comprehensive set of tools for tracing, profiling, and debugging in production environments. Cloud Trace enables the collection of latency data and provides insights into application performance. OpenTelemetry simplifies the process of capturing distributed traces. Cloud Profiler allows for efficient performance analysis and optimization. Cloud Debugger offers powerful capabilities for inspecting live-running applications without disruptions. By leveraging these tools, developers can diagnose and fix issues that may arise in their cloud applications, even in production environments.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP OVERVIEW
TOPIC: GCP CODE AND BUILD TOOLS

Google Cloud Platform (GCP) offers a range of tools and plugins to facilitate development and building processes. In this material, we will explore the various GCP tools available for developing and building applications, including IDE plugins, continuous integration, and continuous delivery.

For developers using Visual Studio (VS) code or IntelliJ, Google provides plugins that allow for rapid iteration, debugging, and deployment of code to runtime environments such as Google Kubernetes Engine (GKE) and other Kubernetes implementations. These plugins, known as Google Cloud Code, utilize popular tools like Skaffold, Minikube, Jib, and Kubectl to provide continuous feedback within the IDE. With Cloud Code, developers can also remotely debug their applications while leveraging IDE debugging features.

Getting started with Cloud Code is made easy with the built-in template, which enables the creation of Kubernetes applications that work seamlessly within seconds. Additionally, Cloud Code supports one-click deployment to local Kubernetes clusters via Skaffold, ensuring a tight development interloop. Developers can also choose to deploy applications to Cloud Run using a stored template and monitor the status of live application resources.

Cloud Code allows for effortless switching between different build profiles, enabling the creation of containers using local Docker installations or tools like Cloud Build. The plugins also provide features such as auto-completion, linting, and inline documentation for Kubernetes config files. Cloud Code offers the ability to generate diffs between local and remote config files, stream logs from deployments, pods, and containers, and inspect a cluster's resources using the built-in Kubernetes explorer.

In addition to container-based applications, Cloud Code also provides access to various Google Cloud libraries within the IDE, making them readily available to developers. The platform also supports different types of development artifacts without the need to build container images, thanks to growing support for build packs.

While building and deploying directly from the development environment can be productive, it may not align with DevOps best practices and can lead to errors due to different build environments. To address this, GCP offers Cloud Build, a fully managed CI/CD platform that runs builds on Google Cloud Platform infrastructure. Cloud Build requires a build config file, expressed in JSON or YAML, which describes the tasks to be performed during the build process. Alternatively, a Docker file can be used.

Builds in Cloud Build can be configured to fetch dependencies, run unit tests, perform static analysis, integration tasks, and create artifacts using tools like Docker, Gradle, Maven, and Bazel. These artifacts can then be deployed to preferred runtimes or artifact repositories.

Cloud Build executes builds as a series of steps, each executed within a Docker container provided by Cloud Build, the community, or the user. Google provides pre-built images called builders that can be referenced in build steps to execute tasks. Builds can be initiated using the G Cloud CLI, the Cloud Build API, or through pre-configured source repositories hosted on platforms like GitHub or Bitbucket. Cloud Build also provides a GitHub app that automatically triggers builds on GitHub events such as pushes and pull requests, allowing developers to view build results on GitHub and the Cloud Console with seamless authentication.

Cloud Build is not limited to application source code; it can also be used in conjunction with infrastructure-as-code tools like HashiCorp's Terraform or third-party developer tools like JFrog's Artifactory.

With the versatility of Cloud Build and the availability of plugins for popular IDEs, developers have the necessary tools to streamline their development and building processes on the Google Cloud Platform.

Google Cloud Platform (GCP) is a suite of cloud computing services provided by Google. In this didactic material, we will provide an overview of GCP, specifically focusing on GCP code and build tools.

GCP offers a wide range of services for developing, deploying, and managing applications in the cloud. These services include computing power, storage, machine learning, data analytics, and networking capabilities. GCP

is designed to be scalable, reliable, and secure, making it suitable for both small-scale projects and large enterprise applications.

One of the key aspects of GCP is its code and build tools. These tools enable developers to write, test, and deploy their applications efficiently. GCP supports multiple programming languages, including Java, Python, and Go, allowing developers to choose the language that best suits their needs.

GCP provides a variety of development environments and tools to streamline the development process. For example, Cloud Shell is a web-based command-line interface that allows developers to manage their GCP resources directly from the browser. It provides pre-installed tools and libraries, making it easy to get started with GCP development.

Another important tool in GCP is Cloud Source Repositories, which provides a version control system for managing source code. It integrates seamlessly with other GCP services, such as Cloud Build and App Engine, enabling developers to easily build, test, and deploy their applications.

GCP also offers Cloud Build, a fully managed continuous integration and continuous delivery (CI/CD) platform. With Cloud Build, developers can automate the build, test, and deployment processes, ensuring that their applications are always up-to-date and reliable.

In addition to these tools, GCP provides a rich set of APIs and SDKs that allow developers to integrate their applications with other GCP services. This enables developers to leverage the full power of GCP and build highly scalable and feature-rich applications.

To summarize, GCP is a comprehensive cloud computing platform that offers a wide range of services for developing, deploying, and managing applications. Its code and build tools provide developers with the necessary tools and environments to write, test, and deploy their applications efficiently. By leveraging GCP's capabilities, developers can build scalable, reliable, and secure applications in the cloud.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GETTING STARTED WITH GCP
TOPIC: CLOUD SQL

Cloud SQL is a fully-managed database service provided by Google Cloud Platform. It offers an easy way to set up, maintain, manage, and administer relational databases in the cloud. Cloud SQL provides high performance, scalability, and convenience, making it a reliable database infrastructure for applications running anywhere.

To get started with Cloud SQL, you need to follow a few steps. First, go to the Cloud SQL Instances page in the Google Cloud Platform console and select your project. Then, click on "Create Instance". Choose MySQL and select Second Generation. Enter "My Instance" as the instance ID and set a password for the root user. You can use default values for the other fields and click on "Create".

After creating the instance, you will be taken back to the instances list. Your new instance will appear grayed out while it initializes and starts up. Once it is ready, you can connect to your instance using the MySQL client in the Cloud Shell. In the Google Cloud Platform console, click on the Cloud Shell icon in the upper right corner. At the Cloud Shell prompt, connect to your Cloud SQL instance and enter your root password. You will then see the MySQL prompt, indicating that you are successfully connected.

Now that you are connected to your Cloud SQL instance, you can perform various operations on the database. For example, you can create a SQL database on your Cloud SQL instance by typing "create database guestbook". You can also insert sample data into the guestbook database and retrieve the data by selecting "select star from entries".

By following these steps, you have successfully created a Cloud SQL instance and performed operations on the database. Cloud SQL provides a reliable and convenient solution for managing relational databases in the cloud.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GETTING STARTED WITH GCP
TOPIC: DATASTORE

This Quick Start guide will walk you through the basic operations in Google Cloud Platform's Datastore using the Google Cloud Platform Console. Datastore allows you to store, query, update, and delete data.

To get started, go to the Datastore Entities page in the Google Cloud Platform Console. This page is where you can perform various operations on your data.

To create a new entity, click on the "Create Entity" button. On the Create Entity page, leave the Namespace as Default and set the Kind as Task.

Under the Properties section, you can add properties to your entity. Click on the "Add Property" button to add a new property. For example, you can add a property called "description" of type String, with the value "Learn Google Cloud Datastore". Click "Done" to set the property.

You can continue adding properties by clicking on the "Add Property" button again. For instance, you can add a property called "Created" of type Date and Time, with the current time. Click "Done" to set this property.

Lastly, let's add a third property called "Done" of type Boolean, with a value of False. Click "Done" to set the property, and then click "Create".

Now, if you go back to the console, you will see that the task entity you just created is displayed.

Congratulations! You have successfully stored data in Cloud Datastore.

Now that your Datastore is up and running, let's run a query. Click on "Query by GQL" to run a query using the GQL language.

Enter the query "Select * from task" (note that "task" is case-sensitive) and click "Run Query". You will see that the results display the task entity you just created.

You can also add a query filter to restrict the results to entities that meet specific criteria. For example, you can run a query like "Select * from task where done equals false" (note that "done" is case-sensitive). Click "Run Query" to see the results. In this case, the query will only return the task entity you just created because its "done" value is false.

Well done! You have successfully stored and queried data in Cloud Datastore.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GETTING STARTED WITH GCP
TOPIC: CLOUD SPANNER

Cloud Spanner is a powerful database that combines the features of a relational database with horizontal scalability. It allows for faster deployment and reduced administrative overhead. In this Quick Start guide, we will learn how to perform basic operations in Cloud Spanner using the Google Cloud Platform Console.

The first step in using Cloud Spanner is to create an instance. An instance is a set of resources that are used by the databases within Cloud Spanner. To create an instance, click on "Create Instance" in the Console. Provide a name and ID for the instance, and choose a regional configuration from the dropdown menu. The instance configuration determines the geographic location where your data will be stored and replicated. Once you have configured the instance, click "Create".

After the instance has been created, it will appear on the Spanner Instances page. Next, we need to create a database. Click on "Create Database" and enter a name for the database. For now, we can skip the step of defining the database schema. Click "Create" to create the database.

Once the database has been created, we can proceed to create a table schema. To do this, click on "Create Table" and switch to the "Edit as Text" mode. Enter the table schema using the Cloud Spanner Data Definition Language (DDL). Click "Create Table" to create the table.

Now that we have a table, we can start inserting, editing, and deleting data. To insert data, click on "Data" and then "Insert". Enter the desired data values and click "Save". You can add multiple rows by repeating this process. Similarly, you can edit data by selecting a row, clicking "Edit", and modifying the values. To delete data, select a row, click "Delete", and confirm the deletion.

In addition to these basic operations, Cloud Spanner also provides a query editor for running SQL statements. To run a query, go to the Tables page and click on "Query". You can then execute prepopulated queries by clicking "Run Query" and view the results.

Congratulations! You have successfully created a Cloud Spanner database and performed basic operations using the Google Cloud Platform Console.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GETTING STARTED WITH GCP
TOPIC: CLOUD SHELL

To get started with Google Cloud Platform (GCP) and Cloud Shell, follow these steps:

1. Click the "Activate Google Cloud Shell" button located at the top right of the console window. This will open a Cloud Shell session within a new frame at the bottom of the console, displaying a command line prompt. It may take a few seconds to initialize.

2. Once your Cloud Shell session is ready, you can perform various tasks. For example, you can navigate to your home directory and use VI to view your bashrc configuration.

3. In this Quick Start, we will preview and deploy an App Engine application. Begin by cloning the sample app and running it locally in the Cloud Shell session using the App Engine development server.

4. To preview the app, click the "Web Preview" button and select the desired port number from the displayed menu. Cloud Shell will open the preview URL in a new browser window using its proxy service.

5. When you are finished previewing the App Engine app, type Control-C to stop the development server.

6. Now that you have previewed the app, it's time to deploy it using the command "gcloud app deploy". This deployment process may take a few minutes to complete.

7. Once the deployment is finished, you can open your application in a web browser. The URL will be in the format "your_project_ID.appspot.com".

8. If the application has not finished deploying, you may encounter an error message in the web browser. In that case, simply wait until the deployment is complete and refresh the page.

Congratulations! You have successfully previewed and deployed an App Engine app using just your browser, thanks to the power of Cloud Shell.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GETTING STARTED WITH GCP
TOPIC: CLOUD VPC

Welcome to the quickstart tutorial for Google Cloud VPC. In this tutorial, we will guide you through the process of creating a custom network and an automatic VPC network using Google Cloud Platform.

To get started, go to the VPC network page in your Google Cloud console. Once there, select "Create VPC Network." In the name field, enter "auto-network1." For the subnet creation mode, choose "Automatic." Finally, click on the "Create" button. This will initiate the creation of the auto network.

Next, let's create a custom network. Click on "Create VPC Network" again. This time, enter "custom-network1" as the name. To add a subnet, click on the "Add Subnet" button. For the name, enter "subnet-us-central-192." Choose "us-central1" as the region. In the IP address field, enter "192.168.1.0/24." Click on the "Add Subnet" button once more.

For the second subnet, enter "subnet-europe-west-192" as the name. Select "europe-west1" as the region. Set the IP address to "192.168.5.0/24." Click on the "Add Subnet" button again.

Finally, for the third subnet, enter "subnet-asia-east-192" as the name. Choose "asia-east1" as the region. Set the IP address to "192.168.7.0/24." Once you have entered all the necessary information, click on the "Create" button.

Congratulations! You have successfully created a custom network and an automatic VPC network using Google Cloud Platform.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GETTING STARTED WITH GCP
TOPIC: PERSISTENT DISKS

Welcome to the Quickstart for Google Cloud Persistent Disks. In this guide, we will walk you through the steps to create and mount a persistent disk in Google Cloud Platform (GCP).

To get started, go to the VM Instance page and select your desired instance. Once you are on the VM Instance Details page, click on the Edit button. Scroll down to the Additional Disk section and click on the Add Item button.

In the new dialog that appears, enter "new-disk" as the name for the disk. Change the disk type to "Standard persistent disk" and set the size to 500 GB. Click on the Create button to create the disk.

Now, go back to the VM Instance Details page and scroll down to the bottom. Select the Save button to save the changes you made.

To confirm that the persistent disk has been created successfully, you will need to connect to the instance using SSH. Click on the SSH button to open a new SSH terminal. It may take some time to connect, so please be patient.

Once connected, type in the following command to list the block devices: sudo lsblk

Next, format the attached disk using the following command: sudo mkfs.ext4 -m 0 -F -E lazy_itable_init=0, lazy_journal_init=0, discard /dev/sdb

Please note that the formatting process may take some time. Once it is done, you can create a mount directory by running the command: sudo mkdir -p /mnt/disk/mymountdir

To mount the disk, use the following command: sudo mount -o discard,defaults /dev/sdb /mnt/disk/mymountdir

Finally, change the permissions on the disk by running the command: sudo chmod a+w /mnt/disk/mymountdir

Congratulations! You have successfully completed the Quickstart for Google Cloud Persistent Disks. You can now use the mounted disk for your desired purposes.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GETTING STARTED WITH GCP
TOPIC: BIGTABLE USING CLOUD SHELL

Welcome to the Quickstart for Cloud Bigtable. In this guide, we will walk you through the steps to get started with Cloud Bigtable using Google Cloud Platform's Cloud Shell.

To begin, navigate to the Bigtable page and select "Create Instance". In the "Instance Name" field, enter "Quickstart Instance". The instance ID will be auto-populated for you. For this Quickstart, choose "Development" as the instance type. Under "Zone", select "us-east-1c". Click "Create" to create the instance.

Once the instance is created, open Google Cloud Shell. After it initializes, run the command "gcloud components update" followed by "gcloud components install cbt" to update and install the necessary components.

Next, insert the project and instance information into the CBT RC file. This will allow us to interact with Bigtable using the cbt command-line tool.

Now, let's create our first Bigtable. Run the command "cbt createtable my-table" to create a table named "my-table". To confirm its creation, type "cbt ls" to list all the tables.

To further organize our table, let's create a new family called "cf1". Run the command "cbt createfamily my-table cf1" to create the family. To confirm its creation, type "cbt ls my-table" to list the families within the table.

Now, let's set a key-value pair in our table using the "cbt set" command. For example, you can run "cbt set my-table row1 cf1:column1=value1" to set the value "value1" for the key "row1" and column "column1" in the "cf1" family.

To verify that the key-value pair was stored properly, use the "cbt read" command. For example, you can run "cbt read my-table" to read all the data in the table.

Congratulations! You have successfully completed the Quickstart for Cloud Bigtable. You have learned how to create a Bigtable instance, create tables and families, and interact with data using the cbt command-line tool.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GETTING STARTED WITH GCP
TOPIC: APP ENGINE PYTHON

To get started with Google Cloud Platform's App Engine Python, follow these steps:

1. Install the Google Cloud SDK and the App Engine Python components locally.
2. In the Cloud console, create a new GCP project and an App Engine application. Choose the region closest to you.
3. Clone the Hello World Python app from GitHub. You can find everything you need in the Python doc sample directory.
4. The minimal Python file included in the directory handles the response to the HTTP request.
5. Start the app locally and test it by visiting localhost:8080. You should see the "Hello, World" message.
6. Keep the development server running to automatically detect and reload any code changes you make.
7. To deploy the app to App Engine, use the command "gcloud app deploy." It will upload all the relevant files to GCP.
8. Once the deployment is finished, you can access the app in your browser using the command "gcloud app browse."
9. Congratulations! Your App Engine app is now live on appspot.com.

By following these steps, you have successfully deployed your first App Engine app using Google Cloud Platform.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GETTING STARTED WITH GCP
TOPIC: CLOUD STORAGE

To get started with Google Cloud Platform (GCP) Cloud Storage, follow these steps:

1. Open the Cloud Storage browser in the Google Cloud Platform Console.
2. Click on "Create bucket" to create a new bucket.
3. Give the bucket a globally unique name. Remember that this name will be publicly visible, so avoid including sensitive information.
4. Choose the "Regional" storage class for your bucket.
5. Select the "us-east1" location for your bucket.
6. Click on "Create" to create your bucket.

Now that you have created a bucket, you can upload files into it:

1. Click on "Upload files" and select the file you want to store in the bucket.
2. Wait for the upload to complete.
3. Once the upload is finished, you will see the file name, size, type, and last modified date displayed in the bucket.

If you want to share the file and make it publicly accessible, follow these steps:

1. Click on the dropdown menu associated with the file you want to share.
2. Select "Edit permissions" from the dropdown menu.
3. Click on the "Add item" button.
4. In the new row that appears, enter "User" for the Entity column.
5. Enter "allUsers" in the Name column.
6. Enter "Reader" in the Access column.
7. Click on "Save" to save the changes.

After saving the changes, you will see a link icon for the object. Clicking on it will reveal the object's public URL. Now your file is online and accessible to everyone.

Congratulations! You have successfully created a bucket in GCP Cloud Storage, uploaded a file, and made it publicly accessible.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GETTING STARTED WITH GCP
TOPIC: COMPUTE ENGINE

Compute Engine is a service offered by Google Cloud Platform that allows users to create and manage virtual machines (VMs) in the cloud. In order to create a VM instance, you need to access the Compute Engine section in the Google Cloud Platform console. Once there, navigate to the VM Instances page and click on the "Create Instance" button.

When creating a new instance, most of the fields will be pre-populated to streamline the process. However, you have the flexibility to adjust any settings according to your requirements. To begin, provide a name for your instance. Then, in the Boot Disk section, click on the "Change" button to configure your boot disk.

The first tab you'll encounter is the OS Images tab. Here, you can choose the desired operating system image for your VM. In this case, select Debian 9 and click "Select" to proceed. In the Firewall section, click on "Allow HTTP Traffic" to enable HTTP access to your VM.

Once you have completed the necessary configurations, click on the "Create" button to initiate the creation of your instance. You can monitor the progress and check the status of your instance on the VM Instances page. Once the instance is ready, it will be listed with a green status icon.

To connect to your instance, click on the SSH button. This will open a new window and establish a connection to your VM. Congratulations! You now have a terminal window that allows you to execute shell commands and interact directly with your Linux instance.

Remember to delete any instances that you no longer need to avoid unnecessary charges. This quick start guide has provided you with the necessary steps to get started with Compute Engine on the Google Cloud Platform.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GETTING STARTED WITH GCP
TOPIC: CLOUD PUB/SUB

To get started with Cloud Pub/Sub on Google Cloud Platform (GCP), you need to ensure that the Pub/Sub API is enabled for your project. Once that is done, you can follow the steps below to create a topic, add a subscription, publish a message, and receive the message.

1. Go to the Pub/Sub page in the GCP console.
2. Click on "Create a Topic".
3. Enter a unique name for your topic, for example, "My Topic".
4. Congratulations! You have just created a Cloud Pub/Sub topic.

To add a subscription to the topic you've created:

1. Use the Display menu for the topic and click on "New Subscription".
2. Type a name for the subscription, for example, "MySub". Remember that this is case-sensitive, so capitalize the "M" and the "S".
3. Leave the delivery type as "Pull" and click on "Create".

To publish a message to the topic:

1. In the Overflow menu for the topic, click on "Publish Message".
2. Enter "Hello World" in the message field.
3. Click on "Publish".

To receive the message you just published:

1. Your subscription needs to perform a pull operation.
2. One way to do this is through the G Cloud command line tool.
3. You can use the Google Cloud SDK locally, but here we'll use the in-console cloud shell to run the following G Cloud command from the guide.
4. The message you sent will appear in the data field of the command output.

Congratulations! You have successfully created a Cloud Pub/Sub topic, added a subscription, published a message, and received the message using a pull request to the subscription.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GETTING STARTED WITH GCP
TOPIC: DEPLOYMENT MANAGER

To deploy a virtual machine using Cloud Deployment Manager on Google Cloud Platform (GCP), follow these steps:

1. Open Cloud Shell by clicking the pencil icon at the top.
2. In the File menu, choose New File and name it vm.yaml.
3. Paste the configuration file from the Quick Start into the vm.yaml file.
4. In the config file, define the virtual machine instance by specifying the machine type, image family, zone, disk, and IP.
5. Replace the placeholder variables in the yaml file with your preferred project and VM image family.
6. Use the gcloud command to deploy the resources, referencing the yaml file you just created.
7. Once the deployment is successful, you will have a new instance deployed.
8. To check on your deployment, use the described command in gcloud, which will provide information about the deployment, including any warnings or errors.
9. You can also navigate to the Cloud Console Web UI and go to the Deployment Manager section to see the deployed instance.
10. Clicking on the deployment will provide more information about the included resources, and you can access details about the VM from there.

Congratulations! You have completed your first deployment using Cloud Deployment Manager on Google Cloud Platform.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GETTING STARTED WITH GCP
TOPIC: RESOURCE ACCESS CONTROL

To get started with resource access control on the Google Cloud Platform (GCP), you will need to follow a few steps. First, open the IAM (Identity and Access Management) page on the GCP console from the top left navigation menu of an existing GCP project. Once on the IAM page, click on the "Add" button located at the top.

In the dialog box that appears, enter the email address of the person you want to grant access to, making sure they are not already in the access list shown. After entering the email address, you will be able to choose the role you want to assign to them. For example, you can select the "Storage Admin" role, which will give the account control over Cloud storage resources.

After adding the new person and assigning them a role, you will need to switch to the account you just added. Initially, the account will have no permissions to view storage resources. However, permissions take a few seconds to propagate. After a quick reload, you will be able to see the photo you stored earlier because you have the Storage Admin role.

As a Storage Admin, you will have the ability to modify or delete the file. However, if you remove the member from the IAM list, the account will lose its permissions. This means that if you reload the storage browser page after removing the member, you will encounter a permissions error because the account is no longer allowed to view storage resources.

To manage resource access control on GCP, you need to open the IAM page, add a new person, assign them a role, and then switch to that account to access and manage resources. Remember that removing a member from the IAM list will result in the loss of permissions for that account.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GETTING STARTED WITH GCP
TOPIC: TEXT PARSING AND ANALYSIS WITH PYTHON

To get started with the Cloud Natural Language API for Python, you need a Google Cloud Platform (GCP) project. In your project, enable the Google Natural Language API and create a service account with a private key in JSON format. These credentials will be used to access Google Cloud APIs, so it is important to keep them secure and not include them in your code or public repositories.

To access the credentials from your project, set up an environment variable. The process may vary depending on your development environment. For example, if you are using PyCharm, you can set the environment variable in the "Edit Configurations" section under "Run". If you are not using PyCharm, you can set the environment variable in the terminal.

Next, prepare your environment for Python development. Install the client library either from the terminal or your integrated development environment (IDE). If you are using Cloud Shell in the GCP console, you can skip this step as the required dependencies are already included.

Once you have set up your environment, create a new file or open an existing one. Import the Google Cloud Client Library and instantiate a client. Provide the text you want to analyze, and create a document object with the text and the type of text (in this case, plain text).

To analyze the sentiment of the text, call the `analyze_sentiment` function, passing the document as a parameter. This function will return an `analyze_sentiment` response, which has three properties: `document_sentiment` (the overall sentiment of the input document), `language` (the language of the text), and `sentences` (the sentiment for each sentence in the document).

To extract the sentiment score and magnitude, assign the `document_sentiment` to a variable and log its `score` and `magnitude`. The score represents the sentiment from -1 (negative) to 1 (positive), while the magnitude represents the absolute magnitude of the sentiment regardless of its score.

Finally, run the code to send your first request to the Natural Language API and analyze the sentiment of the provided text.

The Cloud Natural Language API offers more functionalities, such as syntax analysis and text annotation. To learn more about the client libraries and explore these features, consult the natural language basics, work through the sentiment analysis tutorial, and refer to the Cloud documentation.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GETTING STARTED WITH GCP
TOPIC: TEXT PARSING AND ANALYSIS FOR NODE.JS

To get started with the Cloud Natural Language API for Node.js, you'll need a Google Cloud Platform (GCP) project. You can set up a project in the GCP console. Once you have a project, enable the Google Natural Language API for that project. After enabling the API, create a service account and download the private key as a JSON file. These credentials will be used to access Google Cloud APIs, so make sure to keep this file secure and avoid including it in public repositories.

To access the credentials from your project, go to the terminal and set an environment variable to the path of your service account. Next, prepare your environment for Node.js development and install the client library. If you are using Cloud Shell in the console, you can skip this step as the required dependencies are already included.

Now, create a new project file or open an existing file. Import the Google Cloud client library and instantiate a client. Provide the text you want to analyze. For example, you can use a classic example. Create a document with a field called "content" set to the text you want to analyze and a field called "type" which specifies the type of text (in our case, plain text).

Call the "analyzeSentiment" function, passing the document as a parameter. This function returns a promise that resolves to an array. The first element of the array is an object representing the sentiment analysis response. The sentiment analysis response has three properties: "document sentiment" (the overall sentiment of the input document), "language" (the language of the text), and "sentences" (an array of sentiment for each sentence in the text).

Define a constant for the sentiment of the document and log the sentiment score and magnitude. The sentiment score is a value between -1 and 1, where negative values indicate negative sentiment and positive values indicate positive sentiment. The magnitude is a number between 0 and infinity, representing the absolute magnitude of the sentiment regardless of the score being positive or negative.

Finally, include a catch block to handle any errors that may occur. Save the file and run the code. If everything is set up correctly, you should see the original text, the sentiment score, and the magnitude displayed in the terminal. If not, review the code for any syntax errors.

Congratulations! You have successfully sent your first request to the Cloud Natural Language API for text parsing and analysis using Node.js. Keep in mind that the Cloud Natural Language API offers more functionalities, such as analyzing syntax and annotating text. To learn more about the client libraries, consult the natural language basics or work through the sentiment analysis tutorial available in the Cloud documentation.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GETTING STARTED WITH GCP
TOPIC: TEXT PARSING AND ANALYSIS FOR GO

To get started with the Cloud Natural Language API for Go, you'll need a Google Cloud Platform (GCP) project. You can set up a GCP project in the console. Once you have a project, enable the Google Natural Language API for that project. After enabling the API, create a service account and download the private key as JSON. These credentials are required to access Google Cloud APIs, so it is important to keep them secure and out of your code and public repositories.

To access the credentials from your project, you can set an environment variable to the path of your service account in the terminal. Next, you need to install the client library. If you are using Cloud Shell in the console, you can skip this step as the required dependencies are already included.

Once you have the necessary setup, you can start writing your code. Create a new file or open an existing file in your preferred Integrated Development Environment (IDE). Import the Google Cloud client library as well as any other packages you need.

In your code, instantiate a language client. If there is an error in instantiating the client, handle it accordingly. For this example, we will simply log the error.

Provide the text you want to analyze. In this example, we will use the classic example of "Hello, world!".

Call the AnalyzeSentiment function, passing the context, a document object, and the type of encoding. The document consists of a source of content, which in this case is the text "Hello, world!", and a type of content, which is plain text. Remember to include error handling, and log any errors if they occur.

If there are no errors, the AnalyzeSentiment function will return an analyze sentiment response. This response has three properties: document sentiment, which represents the overall sentiment of the input document; language, which indicates the language of the text; and sentences, which provides the sentiment for each sentence in the document. In this example, we will focus on the document sentiment.

Log whether the sentiment score is positive or negative. The score is a sentiment score ranging from -1 for negative sentiment to 1 for positive sentiment.

Finally, run the code and observe the results. Congratulations! You have just sent your first request to the Natural Language API.

Please note that the Cloud Natural Language API offers more capabilities, such as analyzing syntax and annotating text. To learn more about the client libraries, you can consult the natural language basics or work through the sentiment analysis tutorial available in the Cloud documentation.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GETTING STARTED WITH GCP
TOPIC: CONVERTING SPEECH TO TEXT WITH NODE.JS

To get started with the Speech API for Node.js, you'll need a Google Cloud Platform (GCP) project. You can set up a GCP project in the console. Once you have a project, enable the Speech API for that project. Next, create a service account and download the private key. This key will be used to access Google Cloud APIs, so it's important to keep it secure and out of your code and public repositories.

To securely access the credential from your project, you need to set an environment variable to the path of your service account. You can do this by going to the terminal and setting the environment variable.

Before you start coding, make sure you have prepared your Node.js development environment. Install the client library for the Speech API. If you are using Cloud Shell on the console, you can skip this step as the required dependencies are already included.

For this example, you will need an audio file that contains speech to transcribe. There is a sample audio file available for download (linked in the notes below). Move this audio file to your project's resources folder and rename it to something generic for example purposes. Note that you don't actually have to change the name of your file, but you should remember to write it correctly when prompted to write the file name in your code.

Now, create a new project file or open an existing one. Open the Google Cloud client library and the file system module. Start with an async function called main. In this function, instantiate a speech client. Provide the file name of the audio file you want to transcribe. Use the file system module to read the local audio file and convert it to Base64 encoding.

Create a document called "audio" with a field called "content" and set it to the Base64 string of the audio. Provide some details about the configuration in an object called "config". You must include the type of encoding, the sample rate in Hertz, and the language of the speech in the audio. There are also optional fields that you can explore in the API documentation.

Instantiate an object called "request" which is made up of the "audio" and "config" objects. Call the "recognize" function, passing the "request" object. This function returns a promise that resolves to contain the result, which can have zero or more sequential speech recognition result messages. Each speech recognition result has a property called "alternatives", which may contain one or more recognition hypotheses. An alternative has three properties: "transcript", "confidence", and "words". The "transcript" property represents the words spoken by the user. The "confidence" property is a number between 0.0 and 1.0, where a higher number indicates a greater likelihood that the recognized words are correct. The "words" property is an array of word info objects, but we won't go into detail about this in this quick start.

To print out the transcription, get the first alternative from each result and join them in a single string. Log that transcription. Run the main function and include a catch block to handle any errors.

Save your file and run the code. If everything worked properly, you should see the transcription of the text. If not, review the code for any syntax errors.

Congratulations! You have just sent your first request to the Speech to Text API. To learn more about the client libraries or to consult the speech basics, check out the Cloud documentation.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GETTING STARTED WITH GCP
TOPIC: TRANSLATING SPEECH USING CURL

To get started with the Cloud Translation API for cURL, you need a Google Cloud Platform (GCP) project. You can set up a GCP project in the Console. Once you have a project, enable the Google Cloud Translation API for that project. Additionally, create a service account and download the private key as a JSON file. These credentials will be used to access Google Cloud APIs, so it is crucial to keep the JSON file secure and not include it in your code or public repositories.

To securely access the credentials from your project, open the terminal and set an environment variable to the path of your service account. If you do not have the Cloud SDK installed on your machine, make sure to install and initialize it. Instructions for installing and initializing the Cloud SDK can be found in the linked resources below.

To make a request to the translation.googleapis.com endpoint, you can use the curl command. The curl command should include JSON, specifying the text to be translated, the language to translate from, and the language to translate to. The source and target languages are identified using the iso-639-1 codes. In this case, the source language is English (en) and the target language is Spanish (es). The format of the query should be noted as text for plain text.

After running the curl command, you should receive a response confirming the success of your request. Congratulations! You have successfully made your first request with the Cloud Translation API for cURL.

If you want to learn more about the client libraries or need to consult translation basics, you can refer to the comprehensive Cloud documentation.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GETTING STARTED WITH GCP
TOPIC: SECURING APP ENGINE APPS

To secure your App Engine apps on Google Cloud Platform (GCP), follow these steps:

1. Open your existing GCP project and click on the Cloud Shell icon in the blue menu bar at the top.
2. This will open the Cloud Shell frame at the bottom of your window.
3. In the Cloud Shell, type the command "gcloud projects list" to see a list of your projects.
4. Choose one of your projects to use for this quick start and note down its project ID.
5. Set your project ID by typing the command "gcloud config set project [project ID]".
6. Next, obtain the code for the sample App Engine app from GitHub using the link provided in the Quick Start.
7. Go into the newly created directory and deploy the app using the command "gcloud app deploy".
8. Your App Engine app is now running. You can check it in your browser to ensure that it is working correctly.
9. After authenticating, the app will greet you by name.
10. To add Identity Aware Proxy (IAP) in front of your app, go to the Navigation menu, then select Security, and finally click on Identity Aware Proxy. Alternatively, you can click on the link provided in the Quick Start.
11. Under "All Web Services", locate your App Engine app that you just deployed and select it.
12. On the right side panel, click on "Add Member".
13. Add the email address of a person or group that you want to authorize and choose "IAP-secured web app user" for the role under Cloud IAP.
14. Save your changes.
15. Use the IAP slider next to the App Engine app line to turn on IAP for this app.
16. When browsing the app, the authorized account will still see the same welcome message. However, another account will receive an access denied error, as intended.
17. Congratulations! Your application is now protected by Identity Aware Proxy and can only be accessed by authorized individuals.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GETTING STARTED WITH GCP
TOPIC: SETTING UP BIGQUERY SANDBOX

To analyze data using BigQuery without a credit card, you can set up the BigQuery sandbox environment. This didactic material will guide you through the process of setting up your own BigQuery sandbox.

To begin, go to the BigQuery Web UI at console.cloud.google.com/bigquery. If you already have a Google account, log in. Otherwise, create a new account. If this is your first time logging into Google Cloud, select your country and accept the terms of service. Click "Agree and Continue".

To use the BigQuery sandbox, you need to create a project. Enter a project name and click "Create". Once the project is created, you will be redirected to the BigQuery Web UI. Look for the sandbox banner indicator in the upper left-hand corner of the console.

Congratulations! You are now ready to start using the BigQuery sandbox. In this project, you can load or query data without an attached billing account. The usage will be limited to the same limits as the free tier, which can be found at cloud.google.com/free. Keep in mind that any tables, views, partitions, and partition tables will automatically expire after 60 days.

If you need to overcome the limitations of the sandbox, you can upgrade your project by enabling billing and adjusting the expiration time for your resources.

Happy analyzing!

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GETTING STARTED WITH GCP
TOPIC: CLI FOR GCP

In this quickstart, we will guide you through the process of installing the Google Cloud SDK, initializing it, and running core G Cloud commands from the command line. Before we begin, please ensure that you have a Google Cloud Platform project available and that you have Python and the Google Cloud SDK installed on your system.

To set up the SDK, we will start by using the "gcloud init" command. This command will prompt you to log in and, if you choose to do so, it will open a web browser for you to log into your Google user account. This step is necessary to grant permission access to Google Cloud resources.

After logging in, you will be taken back to the command line. At this point, you can select a cloud platform project to use. If you have the Google Compute Engine API enabled, you will also have the option to choose a default Compute Engine zone.

Once you have completed these steps, the "gcloud init" command will confirm that you have successfully set up the SDK.

To view information about your SDK installation, you can run the following command:

```
1. gcloud info
```

This command will provide you with a summary of details about your Cloud SDK installation, including the installed SDK components, the active user account, and more.

If you want to see the accounts whose credentials are stored locally, you can use the following command:

```
1. gcloud auth list
```

This will display a list of accounts whose credentials are stored on your system.

To see the properties of your active SDK config, you can use the command:

```
1. gcloud config list
```

This will show you the properties of your active SDK configuration.

Finally, if you need help or want to learn more about specific gcloud commands or other topics, you can use the command:

```
1. gcloud help
```

This will provide you with information about gcloud commands and other related topics.

By following these steps, you will be able to install the Google Cloud SDK, initialize it, and run core G Cloud commands from the command line.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GETTING STARTED WITH GCP
TOPIC: PRIVATE CONTAINER REGISTRY/STORAGE

Container Registry is a private container image registry provided by Google Cloud. It allows users to build, store, and manage Docker images securely. In this quickstart, we will learn how to create a Docker image, push it to Container Registry, and pull it back to our local machine.

Before we begin, there are a few prerequisites. First, make sure you have a Google Cloud project selected. Second, ensure that the Container Registry API is enabled for your project. Lastly, you will need to have the Cloud SDK and Docker installed on your system.

To start, we need to create a Docker image of a small Python web app. This can be done by setting up a directory with three files: a Dockerfile, requirements.txt, and app.py. The Dockerfile provides instructions on how to package the application, requirements.txt lists the application and its dependencies, and app.py contains the actual Python code.

Once the directory is set up, we can run the Docker Build command to create the Docker image on our local machine. Next, we need to configure Docker to use the gcloud command line tool for authentication. This can be done by running the gcloud auth command.

To associate the Docker image with a registry name and push it to a specific location, we use the Docker Tag command. Once the image is tagged, we can upload it to Container Registry using the Docker Push command.

To view the images hosted by Container Registry, you can either use the Google Cloud console or visit the image's registry name in your web browser.

If you want to pull the image from Container Registry back to your local machine, you can use the Docker Pull command.

Finally, if you wish to delete the Docker image from Container Registry, you can do so using the gcloud container images Delete command.

This quickstart has shown you how to build a Docker image, push it to Container Registry, and pull it back to your local machine. Container Registry is a powerful tool for managing container images in a private and secure manner.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GETTING STARTED WITH GCP
TOPIC: BUILD AND PACKAGE CONTAINER ARTIFACTS

This didactic material will guide you through the process of building and packaging container artifacts using Google Cloud Platform's Cloud Build. We will cover two methods: building an image using a Docker file and building an image using a cloudbuild.yaml configuration file.

Before we begin, ensure that you have a Google Cloud project selected and that you have installed and initialized the Cloud SDK. To authorize a gcloud command line tool to access your project, run the command "gcloud auth login" in the command line. Connect the gcloud tool with your project by using the command "gcloud config set project".

To build an image using a Docker file, we will first create a script called quickstart.sh for our container to execute. Then, we will create a Docker file. Ensure that quickstart.sh is executable in the command line. In the directory where the Docker file is located, use the command "gcloud build submit" with the tag flag set to the project name and image name you want to create. If the Cloud Build API is not enabled, you will be prompted to enable it.

By following these steps, you have successfully built a Docker image using a Docker file and pushed it to Container Registry. Now, let's explore building the same image using a cloudbuild.yaml configuration file.

In the same directory that contains quickstart.sh and the Docker file, create a file named cloudbuild.yaml. This build configuration file instructs Cloud Build to perform tasks based on your specifications. Once you have built the cloudbuild.yaml file, use the command "gcloud builds submit" with the config cloudbuild.yaml flag to build your image and push it to Container Registry.

After building and pushing the image to Container Registry, you can use the Google Cloud Console to view the build details on the Cloud Build page. On this page, you will find the build history. By clicking on a specific image, you can see the details of that build.

Congratulations! You have successfully built a container image and pushed it to Container Registry using Google Cloud Platform's Cloud Build. This image can now be used with other parts of the Cloud Build process.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GETTING STARTED WITH GCP
TOPIC: CLOUD FUNCTIONS QUICKSTART

Welcome to the quickstart tutorial for Cloud Functions. In this tutorial, we will learn how to deploy a Cloud Function from a Google Cloud project.

To begin, ensure that you have a Google Cloud project with building enabled. The first step is to enable the Cloud Functions API. To do this, go to the Navigation menu, click on API and Services, and then select Enable APIs and Services. Search for Cloud Functions and select it from the results. On the Cloud Functions API page, click Enable to enable it for your project.

Next, go to the Navigation menu and click on Cloud Functions, which can be found under the Compute section. Click on "Create Function" to start creating a new function. Give your function a name of your choice.

For this demonstration, we will keep the HTTP trigger selected. The source code can be edited using the Inline Editor. Cloud Functions support multiple runtimes, and for this tutorial, we will use Node.js 8.

By default, a basic "hello, world" function is provided that responds to an HTTP request. You can click on the expanded Editor window icon to get a better look at the code. In this case, we don't need to make any changes, so click OK to go back.

In the "Function to Execute" field, specify the function in your source code that you want to execute. In this case, we only have the "hello, world" function, so we will select that.

Click "Create" to deploy your Cloud Function. The Cloud Functions Overview page will provide an overview of all your Cloud Functions, including their region, trigger, runtime, and the time of their last deployment, as well as permissions.

Once the deployment is complete, click on the menu of your function and select "Test Function". On the testing page, you can test your function by sending a triggering event in JSON format. The test function responds by outputting the value for the "message" key in the JSON object.

To test the function, click on "Test the Function" and check the output. As expected, the function will output the message that was provided in the JSON object. Below the output, you will see the log associated with the function, including the start time, end time, and status code.

For a more detailed view of the logs, click on "See All Logs for This Function Execution". Here, you will find the same log entries along with additional information if needed.

And that's it! You have successfully deployed a Cloud Function using Google Cloud Platform.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GETTING STARTED WITH GCP
TOPIC: MANAGED KUBERNETES QUICKSTART

Welcome to the Quick Start Tutorial for Managed Kubernetes. In this tutorial, we will learn how to deploy a containerized application using Google Cloud Platform's Managed Kubernetes service.

To get started, we need to enable the Kubernetes Engine API on our project. This can be done by navigating to the API and Services section in the Google Cloud Console and enabling the Kubernetes Engine API.

Once the API is enabled, we will activate the Cloud Shell in our console. The Cloud Shell provides us with an interactive command-line interface where we can run our commands. We can set the default zone for our commands by using the "gcloud config set compute zone" command followed by the desired zone. For this tutorial, let's use the "us-east1" zone.

With the default zone set, we can now create a Kubernetes cluster. A cluster consists of a cluster master and multiple worker nodes. The cluster master manages the cluster, while the worker nodes run the Kubernetes processes. We can create a cluster using the "gcloud container clusters create" command followed by a cluster name.

Once the cluster is created, we need authentication credentials to interact with it. We can obtain these credentials by using the "gcloud container clusters get credentials" command followed by the cluster name. This command will configure the command-line interface for Kubernetes, called kubectl, to use the newly created cluster for subsequent commands.

Now that we have our cluster set up, it's time to deploy our containerized application. We can do this by using the "kubectl create deployment" command followed by a name for the deployment. In this tutorial, let's name our deployment "hello-server". We can also specify the Google Container Registry URL for our application using the "--image" flag.

After deploying our application, we need to expose it to the internet so that we can access it. We can create a Kubernetes service for this purpose by using the "kubectl expose deployment" command followed by the deployment name. In this tutorial, our deployment name is "hello-server". We also need to specify the type of service, which is "LoadBalancer", and the ports to be used.

Once the service is created, a load balancer will be initialized and assigned an external IP address. We can obtain this IP address by using the "kubectl get service" command followed by the deployment name. This IP address can then be used to access our application in a web browser.

And there you have it! We have successfully deployed a containerized web application to Kubernetes using Google Cloud Platform's Managed Kubernetes service.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GETTING STARTED WITH GCP
TOPIC: BIGQUERY WEB UI QUICKSTART

Welcome to the quick start tutorial for the BigQuery web UI. In this tutorial, we will learn how to use the BigQuery web UI to run queries and analyze data.

To begin, navigate to the BigQuery section under Big Data in the navigation menu. BigQuery provides access to various public data sets that we can use for our analysis. For this tutorial, we will use a data set containing USA names between the years 1910 and 2013.

The BigQuery web UI provides a query editor where we can write our SQL queries. Before running a query, we can validate it by clicking on the green check mark icon in the lower right side of the query editor. This query validator will check if our query is valid without actually running it. If there are any errors, it will provide us with the necessary information to correct them.

Once our query is validated, we can click on the Run button to execute it. The results of the query will be displayed below the query editor. We can view the results as a table or in JSON format. Additionally, we have the option to save the results for future reference.

Now, let's explore how to load our own data into BigQuery. For this demonstration, we will use a text file containing the most common US baby names of the year 2014. The file is in CSV format with three columns.

To load this data into BigQuery, we need to create a data set. In the navigation panel, under the Resources section, click on your project name. Then, in the details panel on the right, click on Create Dataset. Enter "babynames" as the data set ID and select "United States" as the data location. Leave all other fields with their default values and click on Create Dataset.

Once the data set is created, we can proceed to load the data into a new table. In the Resources section, we will see our "babynames" data set listed under our project. Click on it and in the details panel, click on Create Table. Change the source from "Empty Table" to "Upload" and browse for the CSV file containing our data. Make sure to select the correct file format, which is CSV. Give the table a name, such as "names_2014", and toggle the "Edit As Text" option to define the schema for the table. In this case, the schema consists of three columns: name (string), gender (string), and count (integer). Finally, click on Create Table to load the data.

Once the data is loaded into the table, we can run queries on it. Click on "Compose New Query" to open the query editor. Let's find out the top five male names in our table. Run the query and the results will be displayed below the query window.

And there you have it! That's a quick start on using the BigQuery web UI to analyze data.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GETTING STARTED WITH GCP
TOPIC: CLOUD ENDPOINTS QUICKSTART

Welcome to the Quickstart tutorial for Cloud Endpoints. In this tutorial, we will learn how to get started using Cloud Endpoints by activating the Cloud Shell in our console.

To begin, we need to download the project code to our instance. In the scripts directory, you will find a couple of scripts that we will be using. Let's start by running deploy_api.sh. This script will deploy an API included in the project files to Cloud Endpoints.

Next, we will deploy the API backend by executing deploy_app.sh. This will create an App Engine flexible environment to host the API Backend and deploy our sample API to App Engine.

To see our API in action, we can run query_api.sh. This command takes a three-letter airport code as an argument. For example, if we enter EWR, our API will return Newark Liberty International Airport.

Once we have our API deployed, we can also track API activity and gain insight into our users and usage with Stackdriver logging. To enable this, we need to first enable the Service Control API on our project.

After enabling the Service Control API, we can generate some activity to work with. The generate_traffic.sh script will generate requests to our API for five minutes, giving us plenty of data to check out metrics for.

Once traffic has been generated, we can check out the Endpoints Services page. Here, we have visibility into the number of requests per second, 500 errors, and latency. Additionally, we can view the logs for the methods of our API through a link at the bottom of the page.

Every request to our API is logged with details such as the timestamp of the request, the method called, and the HTTP response code.

And that's it! With Cloud Endpoints, it is easy to deploy your very own API and gain insights into its usage.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GETTING STARTED WITH GCP
TOPIC: IMAGE RECOGNITION AND CLASSIFICATION WITH CLOUD VISION

To get started with image recognition and classification using Cloud Vision on Google Cloud Platform (GCP), follow these steps:

1. Set up your project and create a Google Cloud Storage bucket. In the Cloud Console, select or create a Cloud project and ensure that billing is enabled. Enable the Cloud Vision API.

2. Create a Cloud Storage bucket in the Cloud Console. Go to the Cloud Storage browser page, click "Create Bucket," and assign a unique name to the bucket. Avoid including sensitive information in the bucket name, as it is globally visible. Choose the location for storing the bucket data and select the default storage class as "Standard." Click "Create" to create the bucket.

3. Upload a demo image to your Cloud Storage bucket. Download the demo image provided and open the Cloud Console storage browser. Select the bucket you just created and click "Upload Files." Choose the demo image JPEG file from your local machine and upload it to the bucket. Once uploaded, make sure to share the image publicly in the Cloud Storage browser.

4. Open the interactive API Explorer template provided in the guide. Replace "cloud-samples-data/vision" in the "image.source.imageUri" field with the name of your Cloud Storage bucket where you uploaded the demo image JPEG file. Click "Execute" to send the request to the service.

5. The JSON response will appear, providing the results of the image annotation request. Congratulations! You have successfully made your first image annotation request using the Cloud Vision API.

To explore specific features, view example annotations, or obtain annotations for individual files or images, refer to the Cloud documentation.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GETTING STARTED WITH GCP
TOPIC: RUNNING A QUERY WITH BIGQUERY WEB UI

BigQuery is a fully managed cloud data warehouse that offers fast SQL analytics for large datasets. In this material, you will learn how to use the BigQuery web UI to run a query. Specifically, you will query the USA Name Data public dataset to determine the most common names in the US from 1910 to 2013.

To get started, navigate to the BigQuery UI at console.cloud.google.com/bigquery. If you are new to BigQuery, you can set up a new project in the BigQuery sandbox by following the instructions in the description below the material. No credit card is required. If you are already a BigQuery or Google Cloud platform user, you can select an existing project.

Once you are in the Query Editor, you can write and run SQL queries directly. If the Query Editor is not currently displayed, click "Compose New Query" at the top right of the window to summon the editor.

Before writing a query, you need to navigate to the USA Names public dataset. In the Resources section of the left-hand navigation, click "Add Data" and "Pin a Project". Type "bigquery-public-data" and click "Pin". This project contains several public datasets. Expand the project and scroll down to expand the USA Names dataset. Click on the "1910 to Current" table to review the table schema. The schema provides the structure of the table and a list of available columns for querying. You can also view table details and preview the data.

To start the query, click "Query Table" and a preloaded query statement will appear in the Query Editor. For this quick start, you can copy and paste the query text provided in the description below the material into the Query Editor. Then, click the green checkmark on the right-hand side of the window to view the query validator. The validator will indicate if the query is valid or not, and it will also show the amount of data the query will process when you run it.

After validating the query, click "Run". The Query Results page will display below the query window. At the top of the Query Results page, you will see the time elapsed and the data processed by the query. Below, a table will show the query results with a header row containing the name of each selected column.

You have the option to save the query for future access and run it at a later time. Additionally, you can save the query results in various formats for further analysis or follow the direct link to explore the results in Data Studio.

Now you are ready to analyze the data using BigQuery's web UI. Happy analyzing!

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GETTING STARTED WITH GCP
TOPIC: LOADING LOCAL DATA INTO BIGQUERY USING THE WEB UI

To load local data into BigQuery using the web UI, you must first have a Google Cloud Platform project with billing enabled. This process can be done in two ways: directly from your local machine or by using Google Cloud Storage as an intermediary.

If you choose to load the data directly from your computer using the BigQuery web UI, there is a limit of 10 megabytes. However, this video will focus on loading data into BigQuery via Google Cloud Storage, which does not have this limit.

To get started, you need to identify the data set to which you want to add the new table. If you already have an existing data set, make a note of its location. If you need to create a new data set, you can do so by selecting the project name in the left-hand navigation menu and clicking the "Create Data Set" button.

Name the data set, choose the location, and then click "Create Data Set". The new data set will appear in the left-hand navigation menu.

Next, navigate to Cloud Storage by typing "storage" into the search bar in the console. Click the "Create Bucket" button and name your bucket. Choose a location for the bucket that is in the same region as the location of the BigQuery data set. It is recommended to choose the exact same location as your destination data set in BigQuery to save on egress charges.

Choose the "Standard Storage Class" and click "Create". You can then upload the file you want to load into BigQuery by clicking the "Upload Files" button or dragging the file from your desktop onto the Google Cloud Storage browser. Once the upload is complete, navigate back to the BigQuery web UI.

Select your data set and click "Create Table". Choose "Google Cloud Storage" as your source and browse for the file you uploaded. The file format should update automatically. Provide a table name and optionally provide the schema details or choose to auto detect the schema.

You can also choose to partition and cluster your data, which is explained in a linked video. If your CSV file has a header row, you can specify the number of header rows to skip in the advanced options.

Click "Create Table" and a load job will be created. Once the job finishes, you can view the table schema, details, and a preview of the data. Your table is now ready to be queried. Click "Query Table" and you can start editing the preloaded query statement in the Query Editor.

Before diving into the data, consider whether you want to delete the original file you loaded into Google Cloud Storage or keep it as a backup. Remember that Cloud Storage and BigQuery storage have separate pricing, so review the pricing documentation for more information.

Thank you for watching and happy analyzing!

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GETTING STARTED WITH GCP
TOPIC: SETTING UP COST CONTROLS FOR BIGQUERY

BigQuery is a powerful tool for processing large amounts of data. However, it's important to have cost controls in place to prevent excessive spending on queries. In this material, we will guide you through the process of setting up cost controls for BigQuery.

You have the option to set custom quotas to manage costs at either the project level or the user level. Project level custom quotas limit the total usage of all users within a project, while user level custom quotas are applied to each individual user or service account within a project. You can use either of these options, or both together. If you choose to use both, usage will be counted against both quotas and will adhere to the stricter limit.

To set up custom quotas, start by navigating to the Quotas menu in the IAM and Admin console. Make sure you have the correct project selected. Filter the quotas for the BigQuery API service. Check the box for "Query Usage Per Day" and/or "Query Usage Per Day Per User" and click on "Edit Quotas".

Next, enter your email and phone number. These contact details may be used when processing certain quota requests. Set your daily quotas in tebibytes. It's important to note that the quotas are in tebibytes, so you may need to make necessary conversions if needed. Once you have entered your quotas, click "Done" and then "Submit Request".

Review the changes you have made and click "Confirm". It may take a few minutes for the quota changes to take effect. Once the quotas are set, if either the project level or user level custom quotas are exceeded, BigQuery will return an error. Daily quotas reset at midnight Pacific Time.

For more information on custom quotas, please refer to the documentation and FAQs provided in the links below. Happy analyzing!

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GETTING STARTED WITH GCP
TOPIC: LOCATING AND QUERYING PUBLIC DATASETS

BigQuery is a fully managed data warehouse that allows for fast SQL analytics over large datasets. It offers over 100 publicly available datasets for analysis, covering various data types such as historical weather and taxi trips in New York City. These datasets, including the US census data, can be joined without the need for importing. They are not only used for vital decision-making in enterprises but also serve as a great starting point for data analysis in BigQuery.

To get started, navigate to the Google Cloud console at console.cloud.google.com. If you are new to Google Cloud and BigQuery, refer to the material provided in the description to set up a new project in the BigQuery sandbox, which does not require a credit card.

Once in the console, open the navigation window and select the Marketplace. In the left-hand menu, you can filter the datasets. Each tile represents a public dataset, such as the American Community Survey. This ongoing survey collects social, economic, housing, and demographic data from over 3.5 million US households annually. The data is used in governmental funding decisions and strategic decision-making in private businesses.

Clicking on a dataset tile provides more details, including a description, sample queries, and metadata like the last update and update frequency. To access the dataset, click the "View Data Set" button. This will open a new console window and bring you to the data set in the BigQuery web UI. You can explore the tables available within the dataset and scroll through other public datasets.

Within a table, you can see the schema or columns available, along with details like size, number of rows, and a preview of the first few rows of data. Clicking "Query Table" will open the Query Editor with a template that already references the selected table. Alternatively, you can choose to run one of the sample queries provided on the dataset's details page.

For example, you can query how the rent cost as a share of median income has changed in King County between 2011 and 2017. Click "Run" in the Query Editor, and within seconds, you will have a table showing the changes for each zip code in the county.

If you prefer to analyze data without writing SQL, you can highlight the table you wish to analyze in the left-hand nav, click "Export," and choose "Explore in Data Studio," which is Google's data visualization tool.

BigQuery offers pay-as-you-go pricing for storage and querying data. There is also a free tier option available to quickly get started. To stay within the free tier, you can use the BigQuery sandbox. More information on how to get started in the BigQuery sandbox is provided in the description.

Happy analyzing!

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GETTING STARTED WITH GCP
TOPIC: COPYING DATASETS IN BIGQUERY

If you have a dataset in BigQuery that you wish to copy within a region or from one region to another, you can do so without needing to extract, move, and reload the data. In this material, you will learn two ways to copy a BigQuery dataset using the cloud console.

Both options require three preparation steps. First, review the required permissions in the documentation page to ensure you have the roles needed for the source dataset, destination dataset, and for creating transfers. Second, you must create the destination dataset where you would like the copy to live. In this tutorial, we will create the destination dataset in the same project. However, this is not required. Third, you must enable the BigQuery Data Transfer Service in the same project as the source dataset.

To copy a dataset using the Copy Dataset icon, select the dataset name of the source dataset that you want to copy. Click the Copy Dataset icon. In the Copy Dataset dialog, select the project ID and destination dataset ID. Optionally, check the Overwrite Destination Table box, if you want to refresh or overwrite the data and schema of the tables in the destination dataset. Click Copy. A Permissions window may appear to give the BigQuery Data Transfer Service permission to manage your dataset copy. If so, click to allow. You can see the progress and view details of the dataset copy under Transfers. Back in the BigQuery console, you will see the tables populate under your destination dataset, once the transfer completes. Consider deleting the old dataset to avoid additional storage costs, if that makes sense for your use case.

To copy a dataset using the transfers UI, click Transfers in the left-hand nav. Click CREATE TRANSFER. In the source dropdown, choose Dataset Copy. Give the transfer a name. In the Schedule Options section, you can choose when the transfer will execute and also set the transfer to repeat at regular intervals. Now, choose your destination dataset in the dropdown. You must copy in the name of your source dataset and project ID. Check the Overwrite Destination Table box, if you want to refresh all data in the destination dataset. You can also receive email notifications if and when a transfer fails. Click Save. You can see progress and view details of the dataset in the transfers UI. Once the transfer completes, you will see the tables populate under your destination dataset. Back in the transfers UI, make sure you use the three dots to delete or disable the recurring transfer, to avoid ongoing charges.

There you have it - two different ways to copy datasets, depending on your needs. At general availability, data copied between regions is billed at the same rates as pricing for compute engine network egress between regions. BigQuery sends compressed data for copying across regions, so the gigabytes build may actually be less than the size of your dataset. All standard BigQuery usage charges for storage and querying will apply on the copied data.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GETTING STARTED WITH GCP
TOPIC: QUERYING CLOUDSQL FROM BIGQUERY

Cloud Computing - Google Cloud Platform - Getting started with GCP - Querying CloudSQL from BigQuery

Data is often scattered in many places, such as storing customer tables in BigQuery while sales tables live in Cloud SQL. To perform analysis, it is important to be able to join these tables together. Cloud SQL Federation from BigQuery allows you to analyze data residing in Cloud SQL in real time without the need to copy or move data. It supports both MySQL and Postgres instances.

To get started with querying CloudSQL from BigQuery, you need to complete the initial setup and run an example query. Here are the steps:

Step 1: Enable the BigQuery connection service
- Start in the Cloud console and select the project that includes the Cloud SQL instance you want to query.
- Enable the BigQuery connection API in the APIs and Services section.

Step 2: Configure public IP connectivity for your Cloud SQL instance
- Navigate to Cloud SQL using the search bar.
- Open the details page of your Cloud SQL instance.
- Select the connections tab and make sure the public IP checkbox is marked.

Step 3: Set up the Cloud SQL database connection in BigQuery
- Navigate to BigQuery using the search bar.
- Click Add Data and select External Data Source.
- Provide the details needed to establish the connection resource, including the database type (MySQL in this case), connection ID, name, location, Cloud SQL instance name, database name, username, and password.
- Click Create Connection.

Step 4: Grant permissions to connection users
- Select the connection in the left-hand navigation and click Share Connection.
- Enter the user's address and select BigQuery Connection User or BigQuery Connection Admin role.

Once the initial setup is complete, you can write queries over tables in the connected database. To query Cloud SQL from BigQuery, you need to use the external query function. The syntax of this function requires the connection ID and a string of the query in the external database's SQL dialect.

For example, you can join the employees and salaries tables in the Cloud SQL database to analyze the average salary by year of hire. The query uses the external query function and the connection ID to select the necessary fields and join the tables.

Another example is joining a BigQuery table with a table in your Cloud SQL database. You can use a snapshot of the salaries table loaded into BigQuery and join it with the Cloud SQL employees table. The query calculates the average salary for employees in each hire year.

After running the query, you will see the results. Keep in mind that the cost of your queries will follow standard BigQuery pricing.

Cloud Computing - Google Cloud Platform - Getting started with GCP - Querying CloudSQL from BigQuery

In this material, we will explore how to query CloudSQL from BigQuery in Google Cloud Platform (GCP). This process allows you to leverage the power of BigQuery to analyze data stored in CloudSQL, providing a seamless and efficient workflow.

To begin, it is important to note that there are different pricing options available for this service. You can choose to pay on demand, based on the amount of data processed, or opt for a flat rate model if it is applicable to your organization.

Setting up a Cloud SQL connection and querying the connection using the external query function is a straightforward process. By establishing the connection, you can access and analyze the data stored in CloudSQL with BigQuery's powerful querying capabilities.

For further guidance and detailed instructions on performing federated queries with Cloud SQL, please refer to the documentation provided in the link below. The documentation will provide you with comprehensive information and step-by-step guidance to help you successfully execute your queries.

Happy analyzing!

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GETTING STARTED WITH GCP
TOPIC: MAKING DATA PUBLIC IN CLOUD STORAGE

To make your data in a bucket public to everyone on the internet, follow these steps:

1. Open the Cloud Storage Browser in Google Cloud Console.
2. In the bucket, you can see all the files that live in it.
3. To make a single file public, click the Actions menu (the three little bars to the side of the file).
4. Select Edit Permissions from the drop-down menu.
5. Add a new entity with the name "Public", then "allUsers", and set the role as "Reader". Click Save.
6. Now you have a link that you can share with anyone who wants to view this specific file in the bucket.

If you want to make all the images in a folder public, follow these steps:

1. Click the Edit Permissions button.
2. Add a new member called "all_users".
3. Set their role as "Storage Object Viewer" in the Cloud Storage section.

By following these steps, you can make every image in your bucket public and accessible to anyone on the internet.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GETTING STARTED WITH GCP
TOPIC: USING OBJECT VERSIONING

In this material, we will learn how to use object versioning with Google Cloud Storage and explore some examples of working with versioned objects.

To get started, let's assume we have a storage bucket already set up called "tiny_homes" and it contains a few photos of our favorite tiny homes. We will be using the Google Cloud Console for this demonstration.

To enable versioning for the "tiny_homes" bucket, we need to use the command "gsutil versioning set on gs://tiny_homes". This command instructs Cloud Storage to create a new version of an object each time the live version of the object is overwritten or deleted.

If we ever need to stop versioning, we can simply use the command "gsutil versioning set off gs://tiny_homes".

To check if versioning is enabled for a bucket, we can use the command "gsutil versioning get gs://tiny_homes". If the output shows "enabled", it means that versioning is enabled for that bucket.

Now, let's explore working with versioned objects in our "tiny_homes" bucket. We have uploaded multiple versions of tiny home pictures to this bucket. However, only the most recent version is visible by default. To see all the versions that have existed, we can use the command "gsutil ls -a gs://tiny_homes".

If we want to access a specific version that is not the most recent, we can simply append the generation number to the object's URL. For example, if we want to access the second version of an object, we would use the URL "gs://tiny_homes/object_name#generation_number".

With object versioning, we have the flexibility to manipulate past or present versions of our storage objects.

Object versioning in Google Cloud Storage allows us to keep track of changes made to our objects and easily access previous versions if needed. By enabling versioning, Cloud Storage automatically creates a new version whenever an object is modified or deleted. We can also retrieve specific versions by appending the generation number to the object's URL.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP NETWORKING
TOPIC: VIRTUAL PRIVATE CLOUD (VPC)

A Virtual Private Cloud (VPC) is a private, isolated virtual network partition that provides managed networking functionality for resources on Google Cloud Platform (GCP). It can be thought of as a virtual version of a traditional physical network, allowing for private communication between virtual machines (VMs) and the cloud. This includes features such as firewall rules and routing to protect against external access and limit public exposure.

In a traditional VPC, the scope is typically bound to a specific geographic region. However, if you want to connect workloads across regions, you would need to establish a VPN connection using public IPs. This approach can become complex and costly as the number of regions increases, requiring the management of multiple VPNs, VPN gateways, routers, and BGP sessions.

To address these challenges, Google Cloud offers the Global VPC. This allows a single VPC to span multiple regions without relying on the public internet for communication. It provides private gateways for on-prem hardware, global scope between regions, sharable configuration between projects, near-real-time logging, and a suite of support services such as shared VPC, Cloud Router, firewall support, VPC peering, and VPN.

The Global VPC eliminates the need for VPNs by leveraging Google's global underlying network infrastructure. This network, which powers services like search, YouTube, and Gmail, dynamically advertises routes across the VPC, enabling VMs to communicate across regions seamlessly.

Using Google Cloud's VPC is beneficial for globally distributed multi-tier applications, connecting GCP-hosted or externally hosted databases to Google's machine learning services, and disaster recovery with application replication. It simplifies network management with a single global VPC, regional segmentation, and a single security policy applied globally. This results in fewer VPNs, routers, and network constructs to manage and troubleshoot.

To set up a VPC using Google Cloud, you can follow these steps:

1. Go to the VPC Networks page in the Google Cloud Platform console.
2. Create a new VPC network by providing a name for the network.
3. Choose the subnet creation mode, which can be automatic or custom.
4. If using a hybrid setup with an existing on-prem network, consider removing the default VPC to avoid overlapping IP ranges.

By following these steps, you can establish a VPC for your existing on-prem configuration using Google Cloud's VPC.

In Cloud Computing, specifically in Google Cloud Platform (GCP), networking plays a crucial role in establishing connectivity between various resources. One of the key components in GCP networking is the Virtual Private Cloud (VPC). In this didactic material, we will explore the concept of VPC and its configuration options.

When setting up a VPC, we have the option to choose between Auto mode and Custom mode. Auto mode is recommended for its simplicity and ease of setup. In Auto mode, the VPC automatically creates a subnet, which is essentially a block of IP addresses for a specific region. The subnet IP addresses are predefined and ensure that there is no overlap with IP ranges in your on-premises network. On the other hand, Custom mode allows for more control over IP range configuration, enabling avoidance of overlapping IP ranges with your on-premises network.

In the Firewall Rules section of VPC configuration, we can select predefined firewall rules that address common use cases for connectivity to Virtual Machines (VMs). These rules provide a level of security and control over inbound and outbound traffic. Alternatively, we can create our own custom firewall rules or choose not to use any rules, although this may compromise security.

In terms of routing, we can choose between dynamic routing mode and static routing mode for the VPC network.

Dynamic routing mode is recommended for its ability to adapt to changes in the network topology. This ensures efficient routing of traffic within the VPC and to external networks. Static routing mode, on the other hand, requires manual configuration of routes and is suitable for simpler network setups.

To test the functionality of our VPC network, we can create a new instance in a specific region and then perform a ping test to determine its assigned IP address. By navigating to the Compute Engine tab and creating an instance, we can specify the VPC network and subnet for the instance. Once the instance is provisioned, we can verify that its internal IP address matches the IP range of the selected subnet.

Google Cloud Platform allows for the mapping of on-premises network topologies to the cloud, enabling seamless integration and connectivity between the two environments. By optimizing network configurations, businesses can effectively utilize their available bandwidth and ensure efficient data transfer.

Virtual Private Cloud (VPC) is a fundamental component of Google Cloud Platform (GCP) networking. By understanding the different configuration options and best practices for VPC setup, organizations can establish secure and efficient network connections within GCP and with their on-premises networks.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP NETWORKING
TOPIC: GOOGLE CLOUD INTERCONNECT

Cloud Interconnect is a feature provided by Google Cloud Platform (GCP) that allows for high-speed direct connections between on-premise systems and resources in the cloud. This enables faster data migration and facilitates the creation of optimal hybrid environments.

When migrating to the cloud, it is often more practical to migrate portions of an on-premise system at a time. However, it is still necessary for the on-premise systems to communicate with the newly created cloud resources. While it is possible to reach cloud resources using Virtual Private Cloud (VPC) firewalls and IP addresses, a direct connection to Google's network can offer higher reliability and performance.

In a scenario where a significant spike in compute resources is expected, such as during a shopping holiday like Black Friday, having compute resources in a GCP-VPC allows for direct communication with on-premise systems. However, this communication would have to go through the public internet, which can introduce security and performance challenges. The public internet is not the most performant, and the additional overhead of a VPN can further degrade performance.

To address these challenges, Google Cloud Interconnect provides a reliable and secure way to connect on-premise workloads to the public cloud. It allows for the extension of the on-premise private network into Google Cloud over a dedicated link. This is particularly important for industries that frequently work between on-premise and cloud environments, such as data migration, replication, disaster recovery, and high-performance computing.

Google Cloud Interconnect offers several options to suit specific needs, but the focus here is on Dedicated Interconnect. Dedicated Interconnect enables direct physical connections between the on-premise network and Google's network. This is achieved by setting up a cross-connect between the on-premise router and the Google network at a co-location facility. A Border Gateway Protocol (BGP) session is then configured over the interconnect to route traffic between the networks.

The immediate benefit of Dedicated Interconnect is an enterprise-grade connection to the Google VPC with a dedicated 10-gigabit per second circuit. It also allows for connectivity beyond Google's existing network locations, enabling scalability and cost savings on egress traffic from the VPC network to the on-premise network. This is particularly useful for transferring large amounts of data, as it can be more cost-effective than purchasing additional bandwidth over the public internet.

Furthermore, Dedicated Interconnect reduces disruptions and drops in connectivity, providing a predictable user experience. Traffic between the on-premise and VPC networks does not traverse the public internet, resulting in fewer potential points of failure. Additionally, VPC's internal IP addresses can be directly accessed from the on-premise network with peering, eliminating the need for additional network devices or VPN tunnels.

To set up Cloud Interconnect, the first step is to ensure that a VPC is set up for the cloud environment. Once that is in place, the decision needs to be made between using a dedicated connection or a partner connection. Dedicated Interconnect is ideal for situations requiring more than a 10-gigabit connection, while a partner connection can be used for lower speed needs. If physical proximity to Google's network is not possible, Partner Interconnect can be used.

Google Cloud Interconnect provides a secure, fast, and reliable way to connect on-premise systems to the public cloud. It offers a dedicated connection between the on-premise network and the Google VPC, allowing for high-speed data transfer and reducing disruptions. By extending the on-premise private network into the cloud, organizations can create optimal hybrid environments and leverage the benefits of both on-premise and cloud resources.

To set up a dedicated interconnect connection between your on-premises network and Google Cloud Platform, follow these steps:

1. Go to the Cloud Interconnect Physical Connections tab in the Google Cloud Platform console.

2. Select Setup Connection and then select Dedicated Interconnect.
3. Click on Continue and then select Order new Dedicated Interconnect.
4. Specify the details for your name, location, and capacity of the interconnect. The capacity is determined by the number of 10-gigabit per second connections you can order.
5. Select Next to skip over the redundancy information for now. If you need redundancy, refer to the documentation linked in the description.
6. Specify your contact information and review your order.
7. Select Place Order and review the Order Confirmation page for the next steps.
8. Once Google finishes allocating resources, you will receive a confirmation email and LOA-CFAs (Letter of Authorization and Connecting Facility Assignment) that you will need to send to your vendor.
9. Your vendor will provision the cross-connects between the Google Peering Edge and your on-premises network.
10. Google will extensively test your access before you can use the Interconnect directly. Follow the resource linked in the description for the specific steps related to your setup.
11. After the testing is complete, configure your VLAN attachments.
12. Click Finish Setup and select Add VLAN Attachment.
13. Give the attachment a name and select or create a cloud router to associate with it. The cloud router must be in the VPC network you want to connect to.
14. Once you finish adding the VLAN attachments, select Create.
15. The attachment takes a few moments to create.
16. Click Configure to add a BGP (Border Gateway Protocol) session to your cloud router's interface.
17. Provide a name for the BGP session, the public or private ASN (Autonomous System Number) of your on-premises router, and an optional advertised route priority.
18. The Cloud router and on-premises BGP-IP addresses are already allocated by the VLAN attachment.
19. The BGP sessions will be inactive until you configure BGP on your on-premises router.
20. If you're setting up redundancy with a duplicate interconnect, repeat these steps for the second interconnect and specify a different Cloud router.
21. For more information, refer to the documentation.
22. Note that there are some nuances between using Cloud VPN, VPC, and Dedicated Interconnect with your own VPN. The official documentation covers these nuances in detail.
23. Once you have configured the dedicated connection, the next step is to protect your cloud instances by configuring firewall rules.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP NETWORKING
TOPIC: FIREWALL RULES

Moving from on-prem to the cloud can bring a ton of nifty features for your company and applications. But one of the biggest challenges, and certainly the scariest, is how this movement can potentially expose your systems to new vulnerabilities. And without taking the right precautions, you can run into the risk of exposing your system in very problematic ways.

Firewall rules are extremely important for a number of reasons. They allow you to isolate your internal network and instances from unwanted access. They allow you to monitor inbound and outbound activity coming from your network for suspicious activity, blocking items that are considered dangerous based on a set of security rules. They establish the first line of defense against attacks, viruses, and malware, and help create a secure network.

In traditional on-prem systems, multiple servers on a single internal network are supported through the use of a cluster of firewalls coupled with a load balancer. A large drawback of this traditional architecture is that it doesn't scale well. In an on-prem environment, a firewall is generally a dedicated piece of hardware that has an upper limit in terms of capacity, and this makes a firewall a choke point. To support dynamic scaling, you'll need to habitually run down to the server room and replace the hardware with ones that can handle increased load. Of course, this creates its own challenge. When the traffic goes back to normal, you've now got a big piece of expensive hardware going unused. This is where Google Cloud platform's distributed firewalls can make a difference. Google's global network has a federation of firewalls that can operate and scale as your systems need them. So you only end up paying for what you use rather than making commitments for long-term expectations. This gives you the same power of your on-site perimeter network, which blocks all incoming traffic by default, but allows you to scale without lifting a finger.

Each VPC network functions as a distributed firewall. A distributed firewall means that, by default, it will handle filtering traffic. But you need to adjust it to handle your access needs, like applying firewall rules to tagged instances. In this example, when a request comes in from a Compute Engine System labeled with the red tag, it hits the applicable firewall rule before being allowed to communicate with the blue tag. While firewall rules are defined at the network level, connections are allowed or denied on a per-instance basis. You can think of the GCP firewall rules as existing not only between your instances and other networks, but between individual instances within the same network.

In Google Cloud, a firewall rule is made up of four things: an action either to allow or deny traffic, the type of protocol to which it applies (such as TCP, UDP, and ICMP), either a source or a destination for which the rule applies, and the ports on which the rule applies. Each of these parameters means that firewall rules can help control traffic to and from your Google Cloud VMs accordingly.

Let's look at what this looks like. I have two existing servers here that are trying to use iPerf to test network speed. Here are my two SSH sessions with these VMs with iPerf setup. But note that I have to use a specific port for it, and since that's not part of the standard firewall rules, it doesn't work. The only default firewall rules created are allow egress and deny ingress traffic, and for Linux instances, allow SSH/TCP traffic on port 22. We're going to create a new firewall rule that allows access for iPerf.

Go to the VPC Network tab and click Firewall Rules. You can see there are a bunch of default firewall rules created for the default network. We need to create one for our custom VPC that our instances are sitting in. So click Add Firewall Rule. Create a rule iPerf access. Change the network to VPC 1. Leave it as Ingress and change it to Allow. The target tag will be iPerf access.

Firewall rules are a crucial component of networking in Google Cloud Platform (GCP). They play a vital role in allowing or blocking traffic between various entities, such as cloud instances and on-premise networks. In this didactic material, we will discuss the importance of firewall rules and how they facilitate the transition from on-premises to the cloud.

When configuring firewall rules in GCP, you need to specify the source IP range, protocol, and port. For example, if you want to run iPerf on port 5001, you would set the source IP range to the public internet and the protocol

to TCP. By configuring these parameters, you can control which traffic is allowed to flow to and from your cloud instances.

To add a firewall rule, navigate to the VM Instances page and select the desired instance. Then, add the appropriate access tag, such as the iPerf access tag, to enable traffic for specific applications or services. This process should be repeated for all relevant instances.

Once the firewall rules are in place, you can test the connectivity of your applications or services. For instance, by running iPerf again, you can verify that the firewall rules are working as intended. This ensures that the necessary traffic is allowed to pass through the firewall.

Firewalls are not only essential for securing your cloud instances but also for establishing connectivity between your on-premise network and your cloud network. They act as the gateway through which traffic can flow between these two environments. By properly configuring firewall rules, you can establish secure and efficient communication channels.

If you are interested in exploring more complex use cases or need detailed documentation, you can refer to the resources provided below. These resources will provide you with in-depth knowledge and guidance on configuring firewall rules for various scenarios.

As you continue your journey towards migrating to Google Cloud, stay tuned for the next episode, where we will discuss configuring IPs. Optimizing your network is crucial for maximizing your bandwidth and ensuring smooth operations in the cloud.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP NETWORKING
TOPIC: IP ADDRESSES

One of the key components of on-prem systems is the use of IP addresses to manage traffic. However, when migrating to the cloud, it is important to understand that IP address logic will change. Planning ahead for IP address changes is crucial to ensure that workloads and their interaction with network traffic are not affected.

In the context of Google Cloud, there are two types of IP addresses: private and public. Private IP addresses are used for communication within a Virtual Private Cloud (VPC) network, while public IP addresses are used for communication with the internet or other VPC networks.

When migrating to the cloud, you cannot simply move your existing IP addresses along with your services. Cloud providers have a limited pool of IP addresses and often reuse previously-assigned IPs. As a result, your services will receive dynamically-assigned internal and public IP addresses, which are ephemeral by default. This means that if you restart your instances, you will lose those IPs.

To address this challenge, there are two solutions available. The first solution is to define firewall rules using tags instead of IP addresses. This allows for more flexibility as the IP addresses change. The second solution is to forward traffic through a managed load balancer with a static IP. While both solutions work, they come with the downside of breaking any dependencies on hardcoded IPs, requiring manual changes and maintenance.

Thankfully, Google Cloud offers a way to reserve static IPs for your services. This means that even if your VM is shut down, you can retain the same internal and public IPs when you spin it back up. This is particularly useful if you are dependent on a specific IP address for your service and want to prevent others from using it. Additionally, you can even promote a previously ephemeral IP to be a static one, saving time and effort.

In Compute Engine, each VM can be assigned one internal and one public IP address. Internal IPs are assigned by default from your subnet range, but you can reserve a static internal IP later if needed. Public IPs, on the other hand, are assigned randomly from a pool unless you assign a reserved static public IP or choose not to assign a public IP at all for security purposes. It is also possible to create a custom public IP on Google Cloud Platform, but further details can be found in the documentation.

When migrating to the cloud, it is important to plan ahead for IP address changes. Google Cloud provides the option to reserve static IPs for your services, ensuring that you can retain the same IPs even when instances are shut down. By understanding and utilizing the different types of IP addresses available, you can effectively manage your network traffic in the cloud.

In the context of Google Cloud Platform (GCP) networking, IP addresses play a crucial role in enabling communication between various resources. While ephemeral public IPs are automatically assigned to instances, there are cases where static IPs are required to ensure consistency and persistence. Fortunately, GCP allows you to create static IP addresses effortlessly.

To create a static IP address, navigate to the Network tab in GCP and access the External IP Addresses section. From there, you can choose to create a new static IP or promote an existing ephemeral IP. It is important to note that when creating a static IP, it is recommended to select the same region as the instance, unless global forwarding is being utilized, as static IPs are regional resources.

Once the static IP address is created, it can be assigned to the desired instance directly from the interface. This assignment enables the instance to have a new static public IP address. It is crucial to ensure that the appropriate firewall rules are in place to allow the desired traffic, such as HTTP traffic on port 80, for the instance associated with the static IP.

Understanding how IP addresses are impacted by different actions within your architecture is essential. This understanding highlights the significance of utilizing remappable IP addresses, particularly for front-end servers in the cloud. Remappable IP addresses provide flexibility and adaptability, allowing for efficient management of network resources.

In some cases, maintaining higher security may necessitate avoiding the use of public IP addresses. However, there may still be a need to fetch updates from the public internet. To address this requirement, the next episode will delve into alternative solutions. By optimizing your network and utilizing the appropriate IP addressing strategies, you can effectively free up bandwidth and enhance overall performance.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP NETWORKING
TOPIC: NETWORK ADDRESS TRANSLATION (NAT)

Cloud Networking: Protecting Internal Endpoints with Cloud NAT

As a Cloud Developer, it is essential to ensure that not everything in the cloud is publicly accessible. In this material, we will discuss how to protect internal endpoints using Cloud NAT in the context of Google Cloud Platform (GCP) networking.

When working with a cloud application that has internal services, it is often necessary to restrict inbound communication while allowing outbound traffic. Traditionally, this could be achieved through a VPN service, which secures and authorizes connections. However, using public IPs in this setup can leave you vulnerable to malicious actors.

Another option is to use a bastion host, which acts as an external endpoint allowing clients to SSH from the public internet. This setup keeps your apps from being publicly accessible but only addresses inbound communication.

In scenarios where you have a multi-tiered application setup in the cloud and an update server on-premise, you may want to allow instances outbound access to the internet without having an external IP address. This is where Network Address Translation (NAT) comes into play.

NAT allows multiple VMs in a subnet to reach the internet using a single public IP address. Traditionally, setting up a NAT gateway required reserving static IP addresses, creating compute instance groups as NAT gateways, creating health checks, and adding default routes. Additionally, traditional NATs introduce potential choke points in the network path, affecting performance and availability.

Fortunately, Google Cloud NAT offers a software-defined networking (SDN) solution that avoids these issues. It is a fully managed service that allows Google Cloud VM instances without external IP addresses and private GKE clusters to connect to the internet. Unlike traditional NATs, it doesn't require custom routing and simplifies management.

With Google Cloud NAT, each internal instance is assigned a unique set of NAT IPs and port ranges. This eliminates choke points, improves scalability, performance, and availability. Additionally, external resources cannot directly access private instances behind Cloud NAT, enhancing VPC isolation and security.

Cloud NAT seamlessly scales with the number of instances and network traffic volume. It provides the same bandwidth as instances with external IP addresses.

To set up Cloud NAT, follow these steps:

1. Set up your VMs in the same VPC and subnet.
2. Configure the web server to be private without an external IP address.
3. Set up a bastion host with an external IP address to allow inbound traffic to the web server.
4. Configure firewall rules to only allow SSH access to the web server through the bastion host.
5. Access the bastion host using SSH and then SSH into the web server.
6. Verify that the web server does not have access to the public internet.

To route egress traffic from the web server to the internet using Cloud NAT, follow these steps:

1. Go to the Google Cloud NAT page and click "Get Started."
2. Enter a gateway name and select the VPC network for your instances.
3. Set the region for the NAT gateway, which should match the region of your instances.
4. Create a Cloud Router in the same region and give it a name.
5. Leave the NAT IP addresses as automatic, which allocates IP addresses based on usage.
6. Click "Create" to create the Cloud NAT gateway.

After setting up Cloud NAT, you will be able to access external resources from the web server without an external IP address.

It is important to note that Cloud NAT does not set up inbound NAT, meaning instances outside your VPC cannot initiate new connections to your cloud instances with NAT. However, Cloud NAT is an excellent managed service for tasks like fetching periodic updates from external servers in another network.

As your cloud environment grows, centralizing control and simplifying your network topology through services like Cloud NAT will save you time and effort in the long run.

Optimizing your network is crucial for maximizing the efficiency and performance of your cloud computing infrastructure. In this didactic material, we will explore the concept of Network Address Translation (NAT) in the context of Google Cloud Platform (GCP) networking.

NAT is a technique used to translate IP addresses between different networks. It allows multiple devices within a network to share a single IP address, conserving IP address space and providing an additional layer of security. By using NAT, you can connect your private network to the internet using a single public IP address.

In GCP, NAT is implemented using Cloud NAT, a fully managed service that provides outbound internet connectivity for virtual machine instances (VMs) running in private subnets. Cloud NAT allows your VMs to communicate with the internet without exposing their private IP addresses.

To understand how Cloud NAT works, let's consider a scenario where you have multiple VMs running in a private subnet within a VPC (Virtual Private Cloud) network. These VMs need to access resources on the internet, but you want to hide their private IP addresses.

Cloud NAT acts as an intermediary between your VMs and the internet. When a VM sends a request to access a resource on the internet, the request is first routed to the Cloud NAT service. Cloud NAT then translates the source IP address of the request to the public IP address assigned to the NAT service. The translated request is then forwarded to the internet.

When the response from the internet is received, Cloud NAT performs the reverse translation, replacing the public IP address with the private IP address of the VM that made the request. The response is then forwarded back to the VM.

By using Cloud NAT, you can simplify your networking configuration and reduce the number of public IP addresses required. It also provides a level of abstraction and security by hiding the private IP addresses of your VMs from the internet.

Network Address Translation (NAT) is an essential technique in cloud networking, and Cloud NAT in Google Cloud Platform (GCP) provides a managed solution for outbound internet connectivity. By leveraging Cloud NAT, you can optimize your network, conserve IP address space, and enhance the security of your infrastructure.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP NETWORKING
TOPIC: SHARED VPC

Cloud Networking: Shared VPC

As cloud applications scale, organizations often face the challenge of maintaining control over network resources while allowing teams to quickly spin up the resources they need. To address this issue, Google created shared VPCs, which offer the best of both worlds. Large organizations with multiple cloud projects can share resources while maintaining logical separation between groups or departments.

Shared VPC allows an organization to connect resources from multiple projects to a common VPC network. This enables secure and efficient communication between resources using internal IPs from the shared network. To set up shared VPC, a project is designated as the host project, and one or more service projects are attached to it. The VPC networks in the host project are called shared VPC networks.

With shared VPC, network administrators can centrally manage the creation of routes, firewalls, subnet IP ranges, VPN connections, and more for the entire organization. At the same time, developers can own billing, quotas, and IAM permissions and autonomously operate their development projects.

For example, let's consider an e-commerce company with an externally facing website application server. This server uses various internally available services, such as personalization, recommendation, and analytics, which are built by different development teams. In this scenario, a shared VPC network can be set up with a host project and three service projects for each of these services, all on different subnets.

The network and security admin would set up the overall security policies in the host project, such as restricting which VMs can have public IPs and access to the internet. Meanwhile, each development team can spin up VMs in their assigned service project and make fine-grained decisions, like setting up compute resources. VMs on shared networks still receive the same network throughput caps and VM-to-VM latency as when they're not on shared networks.

To set up a shared VPC, the organization's admin or someone with shared VPC admin privileges can follow these steps:

1. Create a custom VPC in the host project with subnets for different purposes (e.g., development and production).
2. Set up firewall rules to allow necessary traffic within the network.
3. Enable the host project to be the host project for shared VPC.
4. Choose the option to share specific subnets in the host project VPC with service projects.
5. Select the desired subnets and attach the service projects.
6. Edit the default permissions given to the service projects if needed.
7. Save the configuration and manage additional host project users and admins if necessary.

Once the shared VPC is set up, developers in the service projects can create VMs and specify the shared network and subnet. They can then configure the VMs according to their needs.

It is important to ensure that sufficient IP space is allowed between subnets when configuring subnet IP ranges in the same or different regions to accommodate future growth. Additionally, GCP allows for the expansion of existing subnets without affecting existing VMs' IP addresses.

Shared VPCs provide a powerful solution for organizations that require both centralized control over network resources and the flexibility for development teams to work autonomously. By leveraging shared VPCs, organizations can optimize their network management and resource allocation while maintaining logical separation between projects.

Shared VPC is a powerful feature offered by Google Cloud Platform (GCP) that enhances the flexibility and manageability of your organization's network. With Shared VPC, you can achieve zero downtime during network maintenance or updates, allowing your operations to run smoothly without interruptions.

By utilizing Shared VPC, you can allocate resources across multiple projects within your organization, enabling seamless communication and collaboration between different teams or departments. This feature eliminates the need for complex network configurations and allows for efficient resource utilization.

One of the key benefits of Shared VPC is the ability to optimize your network and free up bandwidth. By centralizing the management of your network resources, you can ensure efficient utilization of available bandwidth, leading to improved performance and reduced costs.

Shared VPC also provides enhanced security and control over your network. You can define granular access controls and permissions, ensuring that only authorized users or projects can access specific resources. This helps in maintaining the integrity and confidentiality of your data.

To learn more about Shared VPC and its capabilities, you can refer to the official documentation provided by Google Cloud Platform. The documentation offers detailed explanations, step-by-step guides, and best practices for implementing and managing Shared VPC within your organization.

Shared VPC is a valuable feature of Google Cloud Platform that allows for flexible and manageable networking across multiple projects. By optimizing your network and freeing up bandwidth, you can enhance the performance and efficiency of your organization's operations.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP NETWORKING
TOPIC: VPC PEERING

Cloud providers, like Google Cloud, build their networks with a series of data centers positioned strategically around the world. To connect these data centers, providers rely on the public internet, which introduces security and performance concerns. To address these issues, corporations utilize virtual private networks (VPNs) and virtual private clouds (VPCs) to securely access cloud resources.

As cloud footprints and team complexity grow, a phenomenon called VPC islanding occurs. This refers to the increasing complexity of managing multiple VPCs in different regions. However, Google Cloud offers a solution to this problem. By default, all VPCs on Google Cloud are global, meaning there is no need to set up a VPC for each region. This simplifies the process and allows users to build a VPC once and move on.

However, some organizations require more fine-grained control over VPC deployment and isolation. In such cases, VPC peering is necessary. VPC peering is useful when there are multiple network administrative domains. For example, an organization might have separate VPCs for the finance and accounting departments, and each department needs access to the resources in the other department's VPC. VPC peering can also be used to connect two VPCs and reach them from an on-prem network with a single VPN.

VPC peering utilizes a global virtual network backbone to establish connections between networks by allowing them to exchange routes. It does not rely on a gateway or VPN connection, eliminating single points of failure and bandwidth bottlenecks.

To set up VPC peering, you can follow these steps:

1. Identify the VPCs you want to peer.
2. Go to the VPC peering page and click on the "Create Connection" button.
3. Enter a name for the peering connection and select the source and destination networks.
4. Repeat the process to create a second peering connection, reversing the source and destination networks.
5. Once created, the networks will automatically connect to each other, and you will see a confirmation that peering is active.
6. Configure the firewall to enable traffic between the peered networks, allowing access only to specific ports and from trusted source IP addresses.
7. Ensure that there are no overlapping IP ranges between the networks or their peered networks, as this can cause routing issues.

After setting up VPC peering, you can confirm its establishment by testing the communication between deployments in the peered VPCs. This can be done by connecting to instances in each VPC and verifying successful data transfer.

VPC peering provides a secure and efficient way to connect VPCs within an organization or across different administrative domains. It simplifies network management and allows for seamless communication between resources in peered VPCs.

VPC peering is a feature provided by Google Cloud Platform (GCP) that allows communication between Virtual Private Clouds (VPCs) in a secure and efficient manner. With VPC peering, two instances located in different VPCs can communicate with each other using private IP addresses.

The main advantage of using VPC peering is the enhanced security it offers. By establishing a direct connection between VPCs, traffic remains within the private network and does not traverse the public internet. This eliminates the need for additional security measures such as VPN tunnels or firewall rules.

Another benefit of VPC peering is improved performance. Since communication between VPCs occurs over Google's private network infrastructure, latency is minimized, resulting in faster data transfer speeds. This is especially useful for applications that require low latency, such as real-time data processing or video streaming.

VPC peering also enhances manageability by simplifying network administration. Once the peering connection is

established, VPCs can be treated as if they were part of the same network. This means that resources, such as Compute Engine instances or Kubernetes clusters, can be easily accessed and managed across multiple VPCs without the need for complex networking configurations.

To learn more about VPC peering and its configuration in GCP, you can refer to the official documentation provided by Google Cloud Platform. It provides detailed instructions and examples on how to set up and manage VPC peering connections.

VPC peering in Google Cloud Platform offers a secure, high-performance, and manageable solution for establishing communication between VPCs. By utilizing this feature, you can optimize your network by ensuring private and efficient communication between your resources.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP NETWORKING
TOPIC: ROUTING

Cloud Computing - Google Cloud Platform - GCP Networking - Routing

In the field of networking, it is often necessary to establish connections between networks while ensuring that data remains secure and accessible only to authorized users. One example of this is when two companies, Company A and Company B, want to connect their private networks without compromising their existing network configurations.

Traditionally, setting up a router for on-premise networks requires significant effort and time. Physical assembly and interconnectivity between users and applications can take days or even weeks. As the network grows, so does the management and operational costs for the company.

However, with Google Cloud Platform (GCP), software-defined routing is made possible through the use of a scalable Distributed Virtual Routing mechanism. GCP routes define the paths that network traffic takes from a virtual machine (VM) instance to its intended destination, whether it is within the same VPC network or outside of it.

Each route in GCP consists of a destination and a next hop, which are represented by IP addresses or ranges of IP addresses. When traffic is sent, it is directed to the next hop based on the destination IP address. VM instances are equipped with controllers that are constantly updated with the network's routing table, ensuring that each packet is sent to the appropriate next hop based on the routing order.

Adding or deleting a route triggers changes that are propagated to the VM controllers using an eventually consistent design. This design ensures simplicity, centralized control, automation, security, encryption, and high performance.

In GCP VPC networking, there are two types of routes: system-generated routes and custom routes. System-generated routes include default routes and subnet routes. The default route defines the path for traffic between the VPC and Google services and the public internet. The subnet route, on the other hand, defines the path for traffic within the VPC to each subnet.

Custom routes, as the name suggests, are routes that you create yourself. These can be static routes or dynamic routes using Google Cloud Router. Static routes require manual creation and maintenance of a routing table. If there is a topology change in either network, static routes must be manually updated. Additionally, static routes cannot automatically reroute traffic in the event of a link failure. However, static routing is suitable for small networks with stable topologies.

To illustrate the setup of static routes, let's consider a scenario with two VPCs in different regions, US East and US Central. We have VPN gateways on both sides and IPSec tunnels connecting them. However, we are unable to ping the internal IP of a server in the US Central VPC from a server in the US East VPC. To enable traffic to be forwarded into the tunnel, we create two static routes.

By creating a route from the US East VPC to the US Central VPC and another route from the US Central VPC to the US East VPC, we establish the necessary connectivity. These routes specify the destination IP range, the network to which the route applies, the next hop, and other optional parameters like priority and target instance tag.

It is important to note that Google's virtual routing plays a crucial role in connecting subnets within a VPC and even extends connectivity to local networks on-premise. By optimizing your network through effective routing, you can free up bandwidth and ensure efficient data transfer.

GCP's software-defined routing offers a scalable and efficient solution for defining traffic routing between networks. With system-generated and custom routes, you have the flexibility to establish secure connections and manage network traffic effectively. Stay tuned for more insights into networking on Google Cloud Platform.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP NETWORKING
TOPIC: CLOUD ROUTER

Dynamic routing is a crucial aspect of networking in the cloud, as it allows for automatic and efficient traffic flow between different networks. Google Cloud Platform (GCP) offers a powerful solution called Cloud Router, which enables dynamic routing and eliminates the need for manual configuration and management of static routes.

When using static routes, network administrators have to manually define how traffic can move between private networks. However, this approach is vulnerable to component failures or unexpected events like wind or squirrels. Cloud Router provides a more reliable and flexible alternative by leveraging dynamic routing protocols.

In a typical scenario, you may have a Virtual Private Cloud (VPC) setup with multiple virtual machines (VMs) in a subnet. Each VM's traffic is directed through static routes to a cloud VPN gateway, which encrypts traffic between your on-premise network and the cloud. This setup is sufficient if you have a single network on-premise with a firewall and router that know how to send traffic to the cloud VPN gateway.

However, if you have multiple networks on-premise, you would need to manually add static routes in Google Cloud to expand each VM's routing table. Additionally, you would need to reconfigure your VPN on both ends, resulting in dropped connections and disruptions. This manual process becomes even more challenging for larger organizations that frequently create new testing networks.

Dynamic routing with Cloud Router solves these challenges by automatically discovering topology changes and routing traffic accordingly. When you extend your on-premise network to Google Cloud, Cloud Router establishes a peer connection with your on-premise VPN gateway or router. The routers exchange topology information through the Border Gateway Protocol (BGP), enabling automatic propagation of topology changes between your VPC and on-premise network.

Cloud Router offers several advantages over static routes. Firstly, it can automatically reroute traffic if a link fails, ensuring uninterrupted connectivity. Additionally, Cloud Router learns on-premise routes through BGP, which can result in lower latency as the network infrastructure selects the best route to reach the destination. Furthermore, Cloud Router scales with your traffic and eliminates the need to statically manage subnets or make manual changes to routing tables.

Setting up Cloud Router is a straightforward process. You can create a Cloud Router for your VPN setup by navigating to the Cloud Router page in the Google Cloud Console. Here, you can provide a name for the Cloud Router and select the VPC network that contains the instances you want to reach. Choose the region where you want to locate the Cloud Router, as it will advertise all subnets in that region.

To establish BGP sessions, you need to select a Google Autonomous System Number (ASN) that is not used elsewhere in your network. The ASN uniquely identifies each network on the internet for BGP sessions. You can choose to advertise all subnets visible to the Cloud Router or create custom routes based on your requirements.

Finally, you can add a tunnel to each VPN gateway and establish BGP sessions between the two Cloud Routers or with your on-premise router. The documentation provides detailed instructions on how to set up these connections.

Cloud Router acts as the orchestral conductor for traffic between your on-premise network and the cloud, especially when using VPN and Cloud Interconnect. It ensures the continuous functioning of a growing network, even in the face of failures or other unexpected events. By leveraging Cloud Router, you can optimize your network and free up your bandwidth.

Cloud Router is a powerful feature offered by Google Cloud Platform that enables dynamic routing and eliminates the need for manual configuration of static routes. It automatically discovers topology changes and reroutes traffic accordingly, ensuring uninterrupted connectivity. By leveraging the Border Gateway Protocol, Cloud Router provides scalability, adaptability, and lower latency. Setting up Cloud Router is a straightforward process, allowing you to optimize your network and improve overall performance.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP NETWORKING
TOPIC: LOAD BALANCING

Google Cloud Platform (GCP) offers powerful networking capabilities, including load balancing, to ensure fast and reliable performance for applications and services. In this didactic material, we will explore how GCP leverages its global network to achieve high-speed and accurate results.

Google has invested heavily in building one of the largest and fastest fiber optic networks in the world to support Google Cloud and its own services like YouTube and Gmail. This network has grown 15 times in the past six years, passing 600 trillion bits per second across land and sea. With this infrastructure, Google is able to serve over 1 billion users a day.

To understand the technology behind this network, let's start by looking at how the internet works from the perspective of a single photon running through Google's network. Imagine we have created a web e-commerce site on Google Cloud that sells cat fashion trends. Users in Singapore send a request from their devices to purchase cat winter apparel.

The user's request first hits the service provider, which recognizes that the destination is a Google server. It then sends the request to the closest Google front-end server (GFE) available. These GFEs are strategically located at the edge of Google's global network, with multiple points of presence worldwide. This allows information to be served as close to the users as possible.

Once the request reaches the GFE, it is directed to Google Cloud's software-defined global load balancer. This load balancer is responsible for distributing HTTP and HTTP(S) traffic to back-end instances in a scalable way. In the ideal scenario, the request is handled by back-end instances running in the Singapore data center, and the data is sent back to the user.

However, if the closest back-end instance group in Singapore is unavailable or not deployed yet, the request is seamlessly directed to other VMs running in the US-East region. This routing to a different region across the globe happens automatically without the developer's intervention. The global L7 load balancer ensures that traffic is balanced and distributed efficiently, even if back-ends go down.

The L7 load balancer performs a weighted selection from the set of back-ends, skipping any unhealthy or saturated connections. This software-based load balancing is highly flexible and can be easily configured through the user interface. It provides a single Anycast IP that can be used to direct traffic, ensuring high availability and fault tolerance.

To enable the routing of the user's request from Singapore to the US-East region, the GFEs forward the query through a subsea fiber optical cable. These fiber networks connect Google Cloud regions across land, spanning thousands of kilometers. They are hidden underground, along railroad tracks, and even across mountain ranges. The speed of light in fiber allows photons to carry vast amounts of data, such as over 2,000,000 photos and 8,000 YouTube videos every second across the network.

Once the request reaches the US-East region, it is directed to a target proxy associated with the Anycast IP of the L7 load balancer. The target proxy terminates the client session and routes the traffic to the correct back-end service on VMs in the US-East region. This ensures that the user's request is handled efficiently and the data is delivered back to the user.

Google Cloud Platform leverages its global network infrastructure, including load balancing capabilities, to provide fast and reliable performance for applications and services. The combination of software-defined load balancing, subsea fiber networks, and strategically located front-end servers allows Google to serve billions of users worldwide.

Load balancing is a crucial component of networking in cloud computing. It ensures that incoming traffic is distributed evenly across multiple servers, optimizing performance and preventing any single server from becoming overwhelmed. Google Cloud Platform (GCP) offers a robust load balancing feature that allows users to achieve high availability and scalability for their applications.

When a user sends a request to a GCP load balancer, it checks the capacity and health of each back-end server. It then routes the traffic to the most available and healthy server. This process helps to optimize for latency and ensures that users receive a fast and responsive experience.

The Google Front End (GFE) plays a key role in load balancing. It caches the response from the back-end server and forwards it to the user. This caching mechanism reduces the load on the back-end servers and improves overall performance.

To further optimize network performance, GCP offers globally extensive regions that are strategically located close to users. This allows users to choose regions that are nearest to their target audience, reducing latency even when back-end servers are overloaded or unhealthy. Additionally, Google Cloud's content delivery network (CDN) can be utilized to leverage load balancing and cache content closer to users, further improving performance.

The transmission of data between back-end servers and users involves the use of fiber optic cables. Photons, which carry the data, are sent from Google's servers through optical transmission equipment that uses infrared lasers to transmit signals. These signals are then transmitted through hundreds of optical fibers, each with a diameter as small as a human hair. These fibers can transmit signals with terabytes of capacity.

However, as photons travel through fiber optic cables, their intensity diminishes due to attenuation. To combat this, Google places optical amplifiers along its cables, which increase the signal's strength over long distances. This ensures that the photons can reach their destination, even when regions are thousands of kilometers apart.

Google's fiber networks connect to GCP regions across five continents. Over the past decade, Google has invested in fiber infrastructure worldwide through a combination of buying and leasing. This extensive network allows for fast and reliable data transmission between regions.

To take advantage of Google's global private network and avoid suboptimal internet paths, users can utilize the Premium Tier network. This network automatically provides access to Google's private network, ensuring optimal performance and reliability.

Load balancing is a critical aspect of GCP networking. It allows for the efficient distribution of traffic across multiple servers, optimizing performance and ensuring high availability. Google's extensive fiber network and caching mechanisms further enhance network performance, providing users with a fast and reliable experience.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP NETWORKING
TOPIC: LIMITING PUBLIC IPS

Cloud Computing - Google Cloud Platform - GCP Networking - Limiting Public IPs

In the realm of cloud computing, security is of utmost importance. It is crucial to approach security comprehensively, without burdening development or operations teams. Google Cloud offers a solution that can greatly assist in this endeavor.

With the increasing number of endpoints, networks, and attack surfaces, implementing automated and effective security policies across your cloud infrastructure can be challenging. Additionally, administrators need to establish guardrails to ensure that workloads comply with security requirements and industry regulations. Public IP addresses pose a significant risk as they expose your enterprise environment to the internet. Therefore, limiting public IPs is essential for securing your environments.

To achieve this, it is necessary to identify which resources in your network utilize public IPs. This can include virtual machines (VMs), load balancers, and VPN gateways. When deploying production-level systems, developers have numerous ways to open public IP addresses. This is where organization policies come into play. Organization policies provide a centralized means of enforcing restrictions on Google Cloud resources across your entire resource hierarchy.

As the organization policy administrator, you can configure constraints that apply to the organization, folders, or projects. These constraints can be inherited by nested folders and projects or overwritten on a case-by-case basis. By utilizing organization policies, administrators can ensure that resources such as VMs and load balancers adhere to basic security requirements at all times.

In this material, we will demonstrate how you can use organization policies as guardrails to prevent the use of public IPs in your Google Cloud network. This tool is particularly useful for IT and security administrators who aim to enforce their security standards across all cloud deployments.

Let's explore how you can limit public IP exposure for VMs, load balancers, and VPN gateways using organization policies.

Compute Engine instances can be directly exposed to the internet when assigned a public IP or when utilizing protocol forwarding. To prevent Compute Engine instances from obtaining public IPs, ensure that you have the organization policy admin role. Then, search for and edit the "constraints compute VM external IP access" organization policy constraint. This constraint allows you to define the set of Compute Engine VMs that are permitted to use public IPs on your network. By customizing the policy, you can restrict public IP creation to specific instances while preventing it for others in your organization.

To prevent protocol forwarding from being enabled, utilize the "constraints compute restrict protocol forwarding creation for types" organization policy constraint. This constraint limits virtual hosting of public IPs by Compute Engine VM instances in your organization.

For VPN gateways, a public IP address is required to connect your on-premises environment to Google Cloud. To safeguard your VPN gateway, employ the "constraints compute restrict VPN peer IPs" organization policy constraint. This constraint restricts the public IPs that are allowed to initiate IPsec sessions with your VPN gateway.

Next, let's discuss load balancers. Google Cloud offers various internal and external load balancers. To prevent the creation of external load balancer types, use the "constraints compute restrict load balancer creation for types" organization policy constraint. Ensure that you add all external load balancer types to the policy values. Alternatively, you can use "external" to cover all types of external load balancers automatically. This approach guarantees that your infrastructure remains secure even as new load balancer types are introduced.

Finally, let's address restricting GKE (Google Kubernetes Engine) surfaces. GKE enables developers to create and expose their services to the internet effortlessly. However, by implementing the policies for VMs and load

balancers, as previously demonstrated, no new GKE services can be exposed to the internet without the organization administrator's knowledge.

By enabling the mentioned organization policy constraints, developers attempting to create a GKE service with an external load balancer will encounter restrictions. The service will have a pending external IP status, and when checking the service status using "kubectl describe service," an error will occur due to the load balancer organization policy constraint in place.

Implementing organization policies as guardrails is an effective way to limit public IP exposure in your Google Cloud network. By configuring the appropriate constraints, you can ensure that only authorized instances, load balancers, VPN gateways, and GKE services have access to public IPs, bolstering the security of your cloud infrastructure.

When working with Google Cloud Platform (GCP) networking, it is important to consider the concept of limiting public IPs. By implementing Org policies, you can ensure that public IPs are assigned only to the appropriate resources and avoid any potential security risks.

It is worth noting that Org policies are not retroactive, meaning they will only apply to new infrastructure requests after the policy is set. This means that you do not need to worry about breaking any existing workloads when adding these policies to your Org.

By applying Org policies, you can easily and efficiently enforce restrictions on public IPs across your entire Org hierarchy or a subset of resources. This can be done from a single centralized place, providing you with greater control and visibility over your network.

Implementing these policies can bring peace of mind, as you can rest assured that there are no stray resources with public IPs that have been assigned by your teams but should not have them. This helps to minimize the risk of unauthorized access and potential security breaches.

In addition to the security benefits, optimizing your network by limiting public IPs can also free up bandwidth. By reducing the number of public IPs in use, you can allocate more resources to other critical tasks, improving overall network performance.

To learn more about limiting public IPs and implementing Org policies in GCP networking, you can refer to the provided link.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP SERVERLESS WITH CLOUD RUN
TOPIC: INTRODUCTION TO CLOUD RUN

As a developer, you may face the challenge of choosing a cloud solution that allows you to quickly deliver cloud applications such as web apps, mobile APIs, and background jobs. However, this decision often involves trade-offs. On one hand, you can choose to manage your own servers, which requires provisioning and configuring them yourself. Additionally, you need to worry about scaling as traffic patterns change, and there is a risk of overprovisioning resources and paying for more than what you actually need.

On the other hand, you can opt for a traditional serverless solution. While this approach offers simplicity, it may limit the languages and libraries you can use. It might also require code changes and make it harder to move your application.

What if there was a way to enjoy the benefits of both worlds? Introducing Cloud Run, a solution that brings server agility to your containerized apps. With Cloud Run, you can deploy any stateless HTTP container, allowing you to write your code in your preferred language with the framework or binary library that suits your needs.

To get started with Cloud Run, you simply need to specify the language, dependencies, and start script in a Dockerfile. With one command, you can package your application into a container and deploy it to the cloud. This means you no longer have to worry about provisioning or managing servers because Cloud Run takes care of that for you.

One of the key advantages of Cloud Run is its automatic and rapid scaling. It scales up or down based on your incoming traffic and can even scale down to zero when there is no traffic. This ensures that you only pay for the resources your app uses, billed down to the nearest one hundredth millisecond.

Cloud Run is built with Knative, which means you can use it with your own Kubernetes engine cluster as well. With Cloud Run on GKE, you can easily build and deploy apps to your own Kubernetes cluster, enjoying the same user-friendly experience and benefits.

If you're looking to build a great app quickly, Cloud Run is worth considering. Its flexibility, ease of use, and automatic scaling make it a powerful tool for deploying containerized applications in the cloud.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP SERVERLESS WITH CLOUD RUN
TOPIC: CLOUD RUN EXAMPLARY DEPLOYMENT

Cloud Run is a serverless platform provided by Google Cloud Platform (GCP) that allows developers to run any stateless container on a serverless environment. With Cloud Run, developers can focus on writing code and deploying applications while the platform takes care of the underlying infrastructure. Cloud Run offers fast and automatic scaling, which is request-aware, meaning that applications can scale down to zero when not in use, resulting in cost savings as developers only pay for the resources used.

In this example, Stephanie Wong demonstrates how easy it is to deploy a serverless microservice using Cloud Run. She deploys a microservice that transforms Word documents to PDFs. To achieve this, she includes OpenOffice, a 15-year-old binary, inside a container and runs it in a serverless environment.

The deployment process involves accessing the Cloud Run console and navigating to the Deployment page. From there, Stephanie selects or pastes the URL of the container image and clicks Create. This simple process creates a serverless container without the need for provisioning infrastructure in advance, creating YAML files, or managing servers. Cloud Run imports the image, ensures it starts, and generates a stable and secure HTTPS endpoint.

The deployed microservice can then be tested by providing a document to convert, and the microservice returns a PDF. The advantage of Cloud Run is its support for Docker containers, allowing developers to run applications written in any programming language or software in a serverless manner.

The code for this example includes a small Python script that listens for incoming HTTP requests and calls OpenOffice to convert the document. The Docker file defines the base image, installs OpenOffice, and specifies the start command. The container image is then created using Cloud Build and deployed to Cloud Run.

Cloud Run allows for automatic scaling of the microservice to thousands of containers or instances in just a few seconds. It enables the deployment of legacy applications to a microservice environment without any changes to the code. However, for developers who require more control, Cloud Run on GKE (Google Kubernetes Engine) offers options for larger CPU sizes, access to GPUs, more memory, and the ability to run on a Kubernetes Engine cluster.

Cloud Run and Cloud Run on GKE are powered by Knative, an open-source project for running serverless workloads. This means that the same microservice can be deployed to any Kubernetes cluster running on Knative. The process involves exporting the microservice into a file, deploying it to a managed Knative on another cloud provider using the kubectl command, and retrieving the URL endpoint.

Cloud Run provides a serverless experience with no servers to manage, allowing developers to focus on writing code. It offers fast scale-up and scale-down to zero, resulting in cost savings. Developers can use any binary or programming language due to the flexibility of containers. Cloud Run also provides access to the Google Cloud ecosystem and APIs. Whether in a fully-managed environment or on GKE, developers can enjoy a consistent experience.

Cloud Run is a powerful serverless platform offered by Google Cloud Platform. It allows developers to deploy any stateless container in a serverless environment, providing automatic scaling, cost efficiency, and flexibility. With Cloud Run, developers can focus on writing code and let the platform handle the underlying infrastructure.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP SERVERLESS WITH CLOUD RUN
TOPIC: CLOUD RUN DEVELOPMENTS

Cloud Run is a serverless platform offered by Google Cloud Platform (GCP) that allows users to run containers on a fully managed environment. It is designed to be enterprise-ready, developer-friendly, and flexible. With Cloud Run, users can deploy containers in seconds, making it possible to use any programming language or library. The platform auto-scales and eliminates the need for infrastructure management, allowing users to focus on their applications.

One of the key features of Cloud Run is its enhanced developer experience. When deploying to Cloud Run, users get an automatic HTTPS endpoint for their service, and they can easily attach their own custom domain. Cloud Run is also portable, accepting container standards as inputs and based on Knative, an open source project that enables serverless workloads to run anywhere. Additionally, Cloud Run is available for Anthos, allowing users to run the same workloads on their own clusters, whether on Google Cloud, other clouds, or on-premises.

Cloud Run follows a pay-per-use model, where users only pay for the time their containers are processing requests, rounded up to 100 milliseconds. This pricing model provides cost efficiency and scalability.

In terms of enterprise readiness, Cloud Run has transitioned from beta to general availability, offering the support and service-level agreement (SLA) that comes with GCP products. It has a 99.95% availability SLA, ensuring high uptime for applications. Cloud Run is also expanding its global presence, with availability in multiple regions around the world. By the end of the year, Cloud Run plans to operate in even more regions, covering all continents.

One of the requested features that Cloud Run has added is the ability to access resources in a user's Virtual Private Cloud (VPC). This is made possible through serverless VPC connectors, enabling connections to services like Cloud Memorystore Redis or Memcache, as well as private IPs of Compute Engine VMs.

Cloud Run also offers improved deployment control with features like blue/green deployments. This allows users to roll out new revisions gradually and control the traffic distribution. Users can start by deploying a new revision without serving traffic, validate its behavior, and then gradually direct a percentage of traffic to the new revision. This controlled rollout enables better software delivery practices and minimizes the risk of issues affecting all users.

Cloud Run provides a powerful serverless platform for running containers, offering enterprise readiness, developer-friendly features, and flexibility. It enables users to focus on building and deploying their applications quickly and efficiently.

Cloud Run is a serverless platform provided by Google Cloud Platform (GCP) that allows users to deploy and run containerized applications. In this didactic material, we will discuss some key developments related to Cloud Run, including virtual rollouts, support for Google Cloud Artifact Registry, support for Google Cloud load balancing, and developer-friendly features.

Virtual rollouts are an important feature of Cloud Run that allows users to ensure the health of their deployments. By performing virtual rollouts over multiple days, users can verify that everything is functioning properly between each step. This enables them to have confidence in the new revisions they are serving to incoming traffic. If any issues arise during the rollout, users have the ability to roll back to a previous revision with just one click in the user interface or one command with GCloud.

The introduction of support for Google Cloud Artifact Registry is another significant development in Cloud Run. Artifact Registry is the evolution of Google Cloud Container Registry and offers several benefits. Users can now have content or registries in specific regions of their choice, allowing for better data locality. Additionally, users can encrypt their containers in the registry using their own encryption keys. Cloud Artifact Registry also provides a free tier and allows for per repository access control using Cloud IAM.

In terms of enterprise-readiness, Cloud Run now supports Google Cloud load balancing. This feature enables users to deploy the same Cloud Run service in multiple regions and utilize Google Cloud load balancer to expose

a global endpoint. This global endpoint routes requests to the closest region, reducing latency and improving resilience in case of regional outages. Furthermore, Cloud Run can be integrated with Cloud CDN, which allows for caching of requests and reduces the load on the service, leading to improved performance.

Cloud Run's integration with Google Cloud load balancer also provides additional features such as Identity-Aware Proxy and Cloud Armor. Identity-Aware Proxy allows users to add a login screen to their Cloud Run service, granting access to only certain users within their organization. Cloud Armor, on the other hand, acts as a web application firewall, protecting against denial of service attacks and allowing users to control access based on IP ranges or geographies.

Cloud Run is continuously evolving to be more developer-friendly. Users have expressed high satisfaction with Cloud Run, and Google Cloud has been actively delivering developer-focused features to enhance productivity. One such feature is the ability to deploy applications using a YAML file, simplifying the deployment process.

Cloud Run on Google Cloud Platform offers various developments and features that make it a powerful and versatile serverless platform. From virtual rollouts to support for Artifact Registry, load balancing, and developer-friendly features, Cloud Run provides users with the tools and capabilities necessary to deploy, manage, and scale containerized applications.

Cloud Run, a serverless platform offered by Google Cloud Platform (GCP), provides developers with the capability to store the configuration of their Cloud Run services in version control systems using config files. This is made possible because Cloud Run implements the Knative serving API, where the config files serve as Knative serving resource descriptors. This approach, known as GitOps, allows developers to use Git as the source of truth for their resource configuration.

To facilitate this process, GCP has introduced a new command called "gcloud run services replace" which takes a YAML file of the service as input. Additionally, Google Cloud has collaborated with the Cloud Code team to integrate Cloud Run into the Cloud Code IDE plugins for IntelliJ and Visual Studio Code. These plugins assist developers in several ways, such as bootstrapping new Cloud Run apps with sample apps, managing services and revisions, viewing their properties, building source code into containers, and deploying them to Cloud Run. Moreover, developers can run their Cloud Run services locally within a containerized environment that closely resembles the production environment.

In terms of container creation, Google Cloud has released Google Cloud buildpacks. Buildpacks are essentially recipes for creating containers in popular languages like Go, Node.js, Python, Java, and .NET. With buildpacks, developers no longer need to write Dockerfiles manually. By using the "--pack" command in Cloud Build, developers can build their source code into a container without needing a Dockerfile. These buildpacks are open source and adhere to the CNCF open standard, making them widely adopted by various vendors.

When deploying to Cloud Run and building containers from a Git repository, the context of the container is captured and displayed within the user interface. This includes providing a link to the build that was used to create the container and a link to the GitHub repository at the specific commit. This context information aids in troubleshooting if any issues arise with the container.

Furthermore, Cloud Run supports continuous deployment practices. From the Cloud Run user interface, developers can easily configure a continuous deployment pipeline for their Cloud Run service. This involves selecting a Git repository, validating the build configuration, and saving the settings. Cloud Run will then create the service and set up a Cloud Build trigger that automatically builds and deploys the service whenever new commits are pushed to the repository. The user interface also offers a history of builds for convenience.

In addition to these features, Google Cloud introduces new ways to trigger Cloud Run services via events. Developers can now trigger their Cloud Run services when events are emitted from other Google Cloud resources. This functionality is based on Cloud audit logs, which are generated whenever GCP resources are modified. The events received by the Cloud Run service adhere to the cloud events open standard, which is a CNCF project.

Lastly, Google Cloud is launching Cloud workflows, which allow developers to orchestrate their serverless tasks using a rich and declarative programming model. For example, developers can process events with multiple Cloud Run services, chain API calls, automate infrastructure management, and implement retry policies easily

with Cloud workflows. An example provided is a workflow that stops all developer machines every day at 6:00 PM. This workflow involves retrieving a list of dev VMs, extracting their statuses, stopping the running VMs, and sending an email to the VM owners.

Cloud Run on Google Cloud Platform offers various features and integrations to enhance the development and deployment experience. With support for GitOps, Cloud Code IDE plugins, buildpacks, continuous deployment, event triggering, and Cloud workflows, developers have a comprehensive set of tools to build, deploy, and manage their serverless applications on Cloud Run.

Cloud Run, a serverless platform offered by Google Cloud Platform (GCP), has undergone significant developments to enhance its developer-friendliness and flexibility. In this didactic material, we will explore the improvements made to Cloud Run, including increased resource limits, extended request processing time, auto-scaling enhancements, and support for gRPC and server-side streaming.

To begin with, Cloud Run now allows users to allocate up to 4 gigabytes of memory and four virtual CPUs (vCPUs), providing more options for resource allocation. Additionally, the previous limit of 15 minutes for request processing time has been extended to one hour, enabling longer and more complex operations on Cloud Run.

One of the challenges with serverless platforms is cold start time, which refers to the time it takes to start container instances when there is a sudden influx of traffic. To address this, Cloud Run now supports the configuration of a minimum number of instances that remain active at all times, even during periods of low traffic. This feature ensures improved performance and reduces the impact of cold starts. Furthermore, idle instances are more cost-effective than active instances, resulting in potential cost savings.

To facilitate better handling of auto-scaling, Cloud Run now sends a signal before shutting down an instance. This signal, known as SIGTERM, allows users to perform necessary actions, such as closing database connections or sending pending data, within a 10-second window before the instance is terminated.

In response to user feedback, Cloud Run has added support for gRPC, a high-performance remote procedure call (RPC) framework. Users can now expose a gRPC server in their Cloud Run services without any additional configuration. This integration enables seamless communication between gRPC and HTTP requests.

Another significant enhancement is the support for server-side streaming. Previously, response sizes were limited to 32 megabytes, and responses were buffered. With the new update, Cloud Run allows streaming of HTTP and gRPC responses, eliminating the size limitation and buffering. This feature enables the streaming of large data, such as video files, directly to clients.

These developments have expanded the capabilities of Cloud Run, making it more enterprise-ready and developer-friendly than ever before. By removing previous limits and introducing new features, Google Cloud Platform aims to provide users with greater flexibility and improved performance. The continuous efforts to enhance Cloud Run demonstrate Google's commitment to empowering developers and enabling them to build innovative solutions.

Cloud Run on Google Cloud Platform offers a powerful serverless environment with various improvements to meet the needs of developers. The increased resource limits, extended request processing time, auto-scaling enhancements, and support for gRPC and server-side streaming contribute to a more efficient and flexible development experience.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP LABS
TOPIC: ACCESS CONTROL WITH CLOUD IAM

Identity Access Management (IAM) is a crucial aspect of cloud computing, especially when it comes to managing access control for Google Cloud Platform (GCP) resources. In this didactic material, we will explore the fundamentals of IAM and its significance in ensuring secure and efficient resource management within an organization.

IAM allows you to determine who can perform specific actions on your Google Cloud resources. At its core, IAM consists of three key components: identity, role, and resource. Think of IAM as an air traffic controller for your business, where identities represent users or entities, roles define the set of permissions, and resources are the objects or services that can be accessed.

Organizational structures and policies can become complex, especially when dealing with projects, workgroups, and dynamic changes in authorization. However, GCP's Cloud IAM provides a clean and universal interface to manage access control across all GCP resources consistently. Whether you are working with App Engine or Google Compute Engine (GCE), Cloud IAM ensures that access control is streamlined and easily manageable.

One of the remarkable aspects of Cloud IAM is that it is offered at no additional charge for GCP customers. You only pay for the use of other services, making it a cost-effective solution for access control management.

Cloud IAM seamlessly integrates with G Suite, allowing you to manage users and groups through the Google admin console. With Cloud IAM, you can create policies that grant permissions to Google Groups, Google-hosted domains, service accounts, or specific Google account holders. It also provides a full audit trail, automatic permission authorization removal, and delegation for administrators.

For established enterprises with complex organizational structures, hundreds of workgroups, and numerous projects, Cloud IAM offers a unified view into security policies across the entire organization. It simplifies compliance processes by providing built-in auditing capabilities, ensuring that your organization meets regulatory requirements.

To further enhance your understanding and practical experience with Cloud IAM, we recommend exploring Qwiklabs. As part of this series, Qwiklabs offers interactive demos and labs that allow you to experiment with the products and use cases discussed. These labs provide hands-on experience and enable you to test different scenarios related to access control management. To access the Qwiklabs environments, please refer to the provided link.

In one of the labs, you will encounter a scenario where User 1 removes project access from User 2. This task demonstrates how permissions can be added or removed for a user. As User 1, you have a Google Cloud Storage bucket, and User 2 currently has access to the file. By removing User 2's permissions, User 2 will no longer be able to access the file. However, User 1 can grant User 2 permission again, allowing them to access the bucket.

IAM plays a vital role in ensuring secure access control for Google Cloud Platform resources. With Cloud IAM, organizations can effectively manage access permissions, streamline security policies, and simplify compliance processes. By leveraging Qwiklabs, users can gain hands-on experience and further enhance their understanding of IAM and its practical implementation.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP LABS
TOPIC: MACHINE LEARNING WITH CLOUD ML ENGINE

Cloud Computing - Google Cloud Platform - GCP labs - Machine learning with Cloud ML Engine

Machine learning and artificial intelligence (AI) are two major buzzwords in today's cloud market. But what exactly is the difference between them? In the context of AI, machine learning can be seen as the algorithms that make AI work, while deep learning is a specific type of machine learning that utilizes multilayer neural networks.

The core activities in machine learning include data gathering, model building, training, evaluation, and parameter tuning. Data is gathered and used to train the model, which is then evaluated for its performance. Through iterative learning from data, algorithms can make predictions on new examples.

Machine learning is behind many everyday experiences such as tagged people in photos, personalized search results, and buying suggestions. It is a powerful tool to jump-start applications that already have a large amount of examples of what the algorithm should do.

Google provides APIs for vision, natural language, translation, and video intelligence, which are great resources for those without training data. These APIs are readily available and serve as a starting point for machine learning projects.

To further explore machine learning, Google offers Qwiklabs, an interactive learning platform. In the Qwiklabs, you can find an introductory walkthrough on training a TensorFlow model both locally and on Cloud Machine Learning Engine (CMLE). The lab also covers deploying the model to the cloud for prediction.

In the lab, you will learn how to create a Cloud ML Engine model, select the exported model to use, set an environment variable for the output path, and deploy the trained model. Once deployed, the model can make predictions and scale to serve multiple concurrent requests.

The lab takes approximately one hour to complete and provides hands-on experience with training and deploying machine learning models using Google Cloud Platform.

In addition to Qwiklabs, Google offers on-demand courses on Coursera to further enhance your knowledge of machine learning.

We hope you found this episode informative and encourage you to explore the resources mentioned, including the on-air webinar series, Qwiklabs, and Google Cloud blogs.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP LABS
TOPIC: SCALABLE STORAGE

Cloud Storage is a storage solution offered by Google Cloud Platform that is specifically designed to handle large volumes of structured and unstructured data. It is a massively scalable storage solution that runs on the same backend infrastructure that powers Google's own applications, such as Gmail, Google Photos, and search indexing.

When choosing a storage option, there are four different types of storage available in Cloud Storage: Multi-Regional, Regional, Nearline, and Coldline. Each type of storage is suited for different use cases and has different pricing structures. For example, if you need to store disaster recovery backups that are rarely accessed, Coldline Storage would be a good option due to its lower cost per gigabyte stored. On the other hand, if you need to support streaming data or applications that require high global uptime, you would want to consider Multi-Regional or Regional Storage.

When choosing a storage option, there are three key factors to consider: availability, minimum storage duration, and pay-per-use pricing. Availability refers to how frequently you need to access the data, while minimum storage duration refers to whether you are archiving monthly backups or storing short-lived data. Pay-per-use pricing allows you to only pay for the storage and access you actually use. Regardless of the storage option you choose, Cloud Storage offers high throughput, low latency, durability, and security.

Data security is a top priority for Google Cloud Storage. All applications and data stored in Cloud Storage benefit from the same security model used by Google to keep its own customers safe. Data encryption takes place on the server side as soon as the data is received, before it is written to disk and stored. Additionally, you can provide your own encryption keys for server-side encryption. Google Cloud Platform and Google's infrastructure are certified for compliance standards and undergo independent security audits.

In Cloud Storage, data is stored in buckets. Buckets are elastic containers that hold your data and associated metadata, and they help you organize and control access to your data. Storage management tasks, such as creating and managing buckets, can be performed using the Google Cloud Platform Console UI or the gsutil Command-line Tool.

To get hands-on experience with Cloud Storage, you can participate in Qwiklabs. In the Qwiklab, you will use the Google Cloud Platform Console UI Tool to create, use, and manage a storage bucket. The lab provides step-by-step instructions and takes approximately 30 minutes to complete. If you prefer using the command-line, you can also perform these operations using the Cloud Storage Quickstart CLI.

Google Cloud Storage is a scalable storage solution that is designed to handle large volumes of structured and unstructured data. It offers different types of storage options to suit various use cases and provides high availability, durability, and security. By using buckets, you can organize and control access to your data. Participating in Qwiklabs allows you to gain hands-on experience with Cloud Storage.

Google Cloud Storage is a service provided by Google Cloud Platform (GCP) that allows users to store and retrieve data in a highly scalable and durable manner. In addition to the GCP labs and on-demand courses available on Coursera, there are various resources available to help you learn more about Google Cloud Storage.

Last week's episode focused on machine learning, where we explored a use case from Google and went through a section of the ML Engine Qwiklab. Machine learning is a branch of artificial intelligence that enables computers to learn and make predictions or decisions without being explicitly programmed. It has applications in various fields, including image recognition, natural language processing, and recommendation systems.

Google Cloud Storage provides a reliable and scalable solution for storing and accessing data in the cloud. It offers multiple storage classes, including Standard, Nearline, and Coldline, each designed for different use cases and cost requirements. Standard storage is suitable for frequently accessed data, while Nearline and Coldline storage are more cost-effective options for less frequently accessed data.

One of the key features of Google Cloud Storage is its durability. Data stored in Google Cloud Storage is automatically replicated across multiple locations, ensuring high availability and protection against data loss. This replication is performed within a region by default and can be extended to multiple regions for additional redundancy.

Google Cloud Storage also offers advanced security features to protect your data. Access control lists (ACLs) and Identity and Access Management (IAM) policies allow you to define fine-grained access permissions for your storage buckets and objects. Additionally, you can enable versioning and object lifecycle management to further control and manage your data.

To interact with Google Cloud Storage, you can use the Google Cloud Console, command-line tools, or the Cloud Storage API. The Cloud Storage API provides a programmatic interface for managing your storage resources, allowing you to create buckets, upload and download objects, and perform various other operations.

Google Cloud Storage is a powerful and scalable storage solution offered by Google Cloud Platform. It provides durability, security, and flexibility for storing and accessing your data in the cloud. By exploring the available resources, such as on-demand courses and labs, you can gain a deeper understanding of Google Cloud Storage and its capabilities.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP LABS
TOPIC: MEANINGFUL INSIGHTS WITH BIGQUERY

Cloud Computing - Google Cloud Platform - GCP labs - Meaningful insights with BigQuery

Cloud computing has revolutionized the way businesses handle and analyze big data. One of the key tools in this field is Google BigQuery, a fully-managed, massive-scale, low-cost enterprise data warehouse running on Google's proven compute, storage, and networking infrastructure.

Traditional approaches to handling complex data require significant investments in system architecture and hardware. Even then, queries can take a long time to run. However, with Google BigQuery, organizations can focus on analyzing data to find meaningful insights using familiar SQL, without the need for infrastructure management or a database administrator.

BigQuery is designed to handle ad hoc queries and aggregating queries across extremely large data sets. It is incredibly fast, capable of scanning terabytes in seconds and petabytes in minutes. This speed enables interactive self-service exploration of massive data sets, leading to better analysis, more creativity, and the discovery of interesting insights.

It's important to note that BigQuery is not meant to replace every enterprise data store. It is not suited for online transaction processing systems or for applying changes as they happen. Additionally, since BigQuery is a cloud-based solution, it is not an on-premise solution.

BigQuery offers dynamic allocation of query and storage resources based on usage patterns. It scales automatically to handle large queries, leveraging the processing power of Google's infrastructure. Sharing and collaboration are easy, allowing you to control access to projects and data according to your business needs. Standard SQL queries make it accessible to anyone, regardless of their technical background.

Data replication across multiple geographies ensures a 99.9% service level agreement (SLA), guaranteeing access to your data at all times. BigQuery also prioritizes data security by encrypting all data at rest and in transit by default.

Pricing for BigQuery is based on the separation of storage and compute concepts. This allows you to scale and pay for each independently. You can choose between a pay-as-you-go model or a flat rate monthly price.

To further explore and practice using BigQuery, you can participate in the Qwik Labs. These labs provide step-by-step instructions on how to load and query data using the BigQuery web UI and the command line tool. Each lab takes approximately 30 minutes to complete.

Google BigQuery is a powerful tool for analyzing big data in the cloud. Its scalability, speed, ease of use, and security features make it an excellent choice for organizations looking to derive meaningful insights from their data.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP LABS
TOPIC: SCALABLE APPS WITH APP ENGINE

App Engine is an integral part of the Google Cloud Platform, providing developers with a solid infrastructure for building and deploying apps. With App Engine, developers can focus on writing and perfecting their code, without having to worry about the complexities of a development platform.

One of the key advantages of App Engine is its scalability. Apps built on App Engine can automatically scale to handle large or small amounts of traffic, ensuring optimal performance at all times. Developers only pay for the capacity they use, making it a cost-effective solution.

Data management is another important consideration for developers, and App Engine offers a range of choices for storing and retrieving data. This flexibility allows developers to choose the most suitable option for their app's requirements.

App Engine also supports a wide range of programming languages out of the box, enabling developers to be productive immediately in a familiar environment. This eliminates the need to learn a new language or framework, saving time and effort.

In addition to these features, App Engine offers several other benefits, including the ability to bring any library or framework into the platform, run multiple app versions and microservices, split traffic between versions, and access diagnostic tools for app monitoring and debugging. App security is also a priority, ensuring that developers can build secure and reliable apps.

To get started with App Engine, you can explore the documentation provided in the description below. The documentation covers various programming languages, allowing you to choose the one that suits your preferences. Additionally, you can try out the Qwiklabs provided for Python, Java, PHP, and Go. These labs will guide you through the process of downloading, testing, and deploying an app, and each lab takes approximately 30 minutes to complete.

App Engine offers developers a powerful and user-friendly platform for building scalable apps in the cloud. With its robust infrastructure, support for multiple programming languages, and extensive features, App Engine is an excellent choice for developers looking to bring their app ideas to life.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP LABS
TOPIC: CONTAINERIZED APPS WITH KUBERNETES ENGINE

Apps are expected to be available 24/7, and developers need to be able to deploy new versions of their apps multiple times a day. In this didactic material, we will introduce Google Kubernetes Engine, a production-ready open-source platform that provides a container-centric management environment. We will also run through a quick demo of a self-paced lab where we deploy a containerized application with Kubernetes Engine.

Containers address the problem of easily and consistently deploying apps in different environments. They allow apps to be broken down into smaller independent pieces that can be deployed or managed dynamically. Containerization also allows for a separation of apps from infrastructure, enabling developers to focus on their apps while IT operations teams handle deployment and management. Containers are lightweight, allowing individual services to be quickly called when needed and available almost immediately.

Kubernetes is a production-ready open-source platform that provides a container-centric management environment. It allows you to interact with your container cluster to deploy and manage your apps, perform administration tasks, set policies, and monitor the health of your deployed workloads. Kubernetes Engine, provided by Google, is the premier managed Kubernetes solution. Leveraging Google's infrastructure, Kubernetes Engine offers a managed environment for deploying, managing, and scaling containerized apps.

With Kubernetes Engine, the compute, memory, and storage resources your application containers require are automatically provisioned and managed. Google's Site Reliability Engineers (SREs) constantly monitor your cluster and its resources to ensure high availability of your services. Kubernetes Engine's auto-scaling feature allows you to handle increased demand while scaling back during quieter periods.

Google has been running everything from Gmail to YouTube to search on containers. With Kubernetes Engine, you can benefit from Google's expertise and experience in building container management systems over the last decade.

To further understand and explore Kubernetes Engine, you can participate in a hands-on lab called Qwiklabs. This lab takes about 30 minutes to complete and demonstrates how to deploy a containerized application with Kubernetes Engine. In the lab, you will create a Kubernetes Engine cluster, authenticate for the cluster, deploy a containerized application, create a Kubernetes service to expose your application to external traffic, and inspect the deployed service. Finally, you can view the deployed application from your web browser using the external IP address with the exposed port.

We hope you found this episode informative, and we would love to hear how you would use Google Kubernetes Engine. Don't forget to explore more resources such as our OnAir webinar series, Qwiklabs, blogs, and on-demand courses on Coursera to enhance your knowledge of Kubernetes Engine.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP LABS
TOPIC: CONNECTING GCP SERVICES WITH CLOUD FUNCTIONS

Cloud Functions is a serverless compute solution offered by Google Cloud Platform (GCP) that allows developers to create event-driven applications. When building apps, developers often rely on various cloud services such as storage, messaging, data analytics, and mobile development. However, seamless integration between these services can be a challenge.

Google Cloud Functions provides a solution to this challenge by allowing developers to connect different services together and extend their behavior simply by adding code. It enables developers to respond to events, such as changes to data in a database or files added to a storage system, by creating triggers and associating Cloud Functions with those triggers.

With Cloud Functions, developers only need to provide the code, as the software and infrastructure are fully managed by Google. This means that the function can scale from a few requests to millions per day without any additional effort from the developer. This serverless architecture allows developers to offload resource-intensive work that wouldn't be practical to run on a user's device.

Cloud Functions can be used in various scenarios. For example, when a user subscribes to a newsletter, a Cloud Function can be triggered to send an email to their inbox. Another example is creating a function that automatically generates a thumbnail of an image uploaded to a storage bucket and saves it in another bucket.

Furthermore, Cloud Functions offer opportunities for integration with third-party services and APIs, enabling developers to leverage additional functionalities and capabilities.

To get started with Cloud Functions, Google provides Qwiklabs, which offers self-paced labs that guide users through creating, deploying, and testing Cloud Functions. These labs can be accessed through the command line or the Google Cloud Platform console. In the labs, users will create Cloud Functions, deploy and test them, and view logs to monitor their functions' performance.

Google Cloud Functions is a serverless compute solution that allows developers to create event-driven applications by connecting different cloud services together. With Cloud Functions, developers can easily respond to events and offload resource-intensive work. Integration with third-party services and APIs is also possible. The Qwiklabs provide a hands-on learning experience for creating, deploying, and testing Cloud Functions.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP LABS
TOPIC: HEALTH MONITORING WITH STACKDRIVER

Stackdriver is a comprehensive monitoring, logging, and diagnostics tool for cloud-powered applications across various platforms. It provides insights into the health, performance, and availability of your apps, enabling you to identify and resolve issues quickly.

One of the key features of Stackdriver is monitoring. It collects metrics, events, and metadata from platforms like GCP, AWS, and common application components. With this data, you can create alerts and dashboards to track the performance of your applications.

Stackdriver also offers logging and error reporting functionalities. Logging allows you to filter, search, and view logs from your code and cloud provider services. You can export these logs for further analysis using services like BigQuery, Cloud Storage, and Pub/Sub. Error Reporting monitors your application's errors, aggregates them, and alerts you to any new issues. It provides insights into the instances of these errors, the versions of your app in which they occurred, and when they were first or last seen.

Stackdriver introduces a new brand called APM (Application Performance Management) for its tools, which includes Stackdriver Trace, Debugger, and Profiler. Stackdriver Trace is a distributed tracing tool that allows you to visualize how requests propagate across your services, helping you identify and improve latency issues. Debugger enables production breakpoints without impacting customer interactions, and it can add log statements to your production code without requiring redeployment. Lastly, Stackdriver Profiler, currently in beta, provides insights into CPU or memory-hungry functions in your code, helping you optimize performance and reduce costs.

In addition to the features mentioned above, the material also introduces a self-paced lab on monitoring a Compute Engine VM instance with Stackdriver. This lab guides you through the process of installing monitoring and logging agents for your VM, creating charts for CPU load and network packets, testing the check and alerting, and viewing logs using Stackdriver Logging.

To learn more about Stackdriver, you can explore Google Cloud's on-demand courses on Coursera, visit their on-air webinar series, Quick Labs, and read their blogs for further information.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP LABS
TOPIC: GOOGLE CLOUD DEPLOYMENT MANAGER

Deployment Manager is an infrastructure deployment service provided by Google Cloud Platform (GCP) that automates the creation and management of GCP resources. As customers use Cloud in larger capacities, simplifying Cloud management becomes increasingly important. Deployment Manager serves this purpose by allowing users to create configuration files that define the resources they need, and then automating the deployment process based on these files.

Think of Deployment Manager as a cookbook that combines all the recipes into one place. Each resource can be seen as a recipe, and by creating configuration files, the deployment process can be repeated consistently. Declarative language can be used in these configuration files, allowing users to specify exactly what they want the configuration to be, while leaving the system to figure out the necessary steps to achieve it.

For more complex architectures and configurations that are intended to be reused, Deployment Manager allows users to break down their configuration into templates. Templates enable users to separate their configuration into different pieces that can be reused across different deployments.

Deployment Manager provides the flexibility to deploy systems in various locations and easily roll out new services. It also allows users to deploy multiple versions of their code simultaneously. Additionally, it offers the ability to modify a deployment by adding or removing resources, as well as updating properties of existing resources.

To further understand Deployment Manager, a self-paced lab is available. In this lab, users will utilize the gcloud Command-Line tool to create a simple configuration file, use that file to deploy resources, and view deployment information in a manifest. The lab can be accessed through the provided link, and completion is estimated to take around 30 minutes.

During the lab, users will create a configuration file (vm.yaml) where they can specify values such as project ID and the VM image to deploy. To deploy the configuration, a gcloud command is run, which will display the status of the deployment. Once the deployment is complete, users can use the describe command to view information about the deployment. The manifest ID output from the describe command can be used to access the deployments manifest.

To stay updated on the latest information about Deployment Manager and other GCP services, users are encouraged to visit the OnAir webinar series, Qwiklabs, and blogs. Additionally, on-demand courses on Coursera provide an opportunity to learn more about Deployment Manager.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP LABS
TOPIC: EVENT DRIVEN PROCESSING WITH CLOUD PUB/SUB

Cloud Pub/Sub is a messaging service provided by Google Cloud Platform that enables the exchange of messages between independent applications. This service allows senders, known as publishers, to decouple from receivers, known as subscribers. Messages are organized into topics, and subscribers can either pull related messages or receive push messages from these topics. It is important to note that neither publishers nor subscribers need to have any knowledge of each other.

Cloud Pub/Sub is particularly useful for distributing event notifications. For example, a service that accepts user sign-ups can send notifications whenever a new user registers, and downstream services can subscribe to receive these notifications. This service is also involved in the architecture of smart home technology, where it supports the streaming of data from residential sensors to cloud-based backend servers.

One of the key features of Cloud Pub/Sub is its ability to handle millions of streaming events per second from anywhere in the world. This enables real-time event response. Cloud Pub/Sub is part of Cloud's Stream Analytics solution, which offers various applications such as sentiment analysis, identifying patterns and trends in data, monitoring campaign performance, and crafting real-time messages.

To get hands-on experience with Cloud Pub/Sub, you can use the Qwik labs available through the GCloud command line tool or the Google Cloud Platform console. These labs cover activities such as setting up a topic, subscribing to a topic, and publishing and consuming messages using a pull subscriber. Each lab takes approximately 30 minutes to complete.

In the lab, you will create a Pub/Sub topic using the Google Cloud Platform console and add a subscription. You can then use the console to publish a message to the topic. For example, you can publish the message "Hello Demo." To view the message, you can pull it from the topic using the subscription.

We hope you found this information about Cloud Pub/Sub useful. Remember, you can apply what you've learned today with a $300 free trial credit to get started on Google Cloud Platform. Additionally, there are other Google Cloud training resources available to further enhance your understanding of Cloud Pub/Sub. Thank you for watching, and we look forward to hearing about how you use or would use Google Cloud Pub/Sub.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP LABS
TOPIC: SLACK BOT WITH NODE.JS ON KUBERNETES

In this didactic material, we will explore the topic of building a Slack bot with Node.js on Kubernetes using Google Cloud Platform (GCP). We will provide a step-by-step guide on how to create a bot that posts messages, and explain the concepts of Kubernetes engine and bot users in Slack.

Kubernetes engine is a managed environment provided by Google Cloud Platform for deploying, managing, and scaling containerized applications. It allows for easy and consistent deployment of apps in different environments by breaking them into smaller, independent pieces called containers. Containerization also enables the separation of apps from infrastructure.

To start, we need to create a cluster on Kubernetes engine. This ensures that a new node is added to the cluster if the pods don't have enough capacity to run. Conversely, if a node in the cluster is underutilized, Kubernetes engine can delete the node. We will also use Kubernetes engine to create a deployment.

Bot users in Slack are similar to regular users, but they are controlled programmatically through APIs instead of interacting with the workspace through mobile or desktop apps. They have profiles, can be messaged or mentioned, post messages and upload files, and can be invited to or kicked out of conversations. Within a Slack channel, bots can perform tasks based on the programming they are given.

In the hands-on lab, we will create a custom bot integration in Slack, build a Node.js image in Docker, upload the Docker image to a private Google Container Registry, and run the Slack bot on Kubernetes engine. The lab requires access to a Slack team where you are authorized to create custom integrations. It will take approximately an hour to complete.

To set up the lab, you will clone the code repository and install Node.js dependencies. Then, you will create a new Slack app, add a new bot user to the app, and obtain an OAuth access token for the bot user. The OAuth access token will be used to edit the Node.js file. After running the bot, you will see that the bot user is online in Slack. You can then send a message to the bot and receive a response.

This hands-on lab provides an opportunity to learn how to build a Slack bot using Botkit, run it on Kubernetes engine, and interact with it in a live Slack channel. By completing the lab, you will gain practical experience in creating custom bot integrations, building and deploying containerized applications, and utilizing Kubernetes engine on Google Cloud Platform.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP LABS
TOPIC: EXPLORING NCAA DATA WITH BIGQUERY

BigQuery is a fully managed, massive scale, low-cost enterprise data warehouse running on top of Google's proven compute, storage, and networking infrastructure. It allows users to focus less on developing infrastructure and more on finding insights from their data. BigQuery is super fast, capable of scanning terabytes in seconds and even petabytes in minutes. This enables interactive self-service exploration of massive datasets, leading to better analysis, more creativity, and the ability to derive more interesting insights.

In this hands-on lab, participants will use BigQuery to explore the NCAA dataset, which includes basketball game data, teams, and players. The dataset covers play-by-play and box scores back to 2009, as well as final scores back to 1996. This lab is based on the NCAA's migration of over 80 years of historical and play-by-play data from 90 championships and 24 sports to the Google Cloud platform. As the official public Cloud provider of the NCAA, Google Cloud is proud to support this data migration.

Additionally, Google Cloud, NCAA, and Kaggle partnered for a competition using the NCAA March Madness Tournament as the common backdrop. With $100,000 up for grabs, participants had the opportunity to strengthen their knowledge of basketball, statistics, data modeling, and cloud technology while competing for the most innovative applications of machine learning.

The lab provides a link to start the quick lab, which will take approximately 45 minutes to complete. Participants will run various queries against the NCAA dataset using BigQuery. Some of the results may be surprising. The lab covers finding the types of basketball plays, identifying teams that scored the most points in a game, determining the top 10 teams with the most cumulative points since 2010, and analyzing which conferences excel at winning tight games.

To further enhance your learning experience, sign up for the $300 free trial credit on Google Cloud Platform (GCP). This credit allows you to apply what you've learned in the lab. Additional training resources are also provided for further exploration.

We hope you enjoy this lab and have fun analyzing the NCAA dataset with Google BigQuery. We would love to hear how BigQuery has helped you gain insights into your own datasets as well.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP LABS
TOPIC: SCALABLE DATABASE SERVICE WITH CLOUD SPANNER

Cloud Spanner is a unique relational database service offered by Google Cloud Platform (GCP) that provides both strong transactional consistency and horizontal scalability. This means that developers no longer have to choose between data consistency and scalability when developing new applications. In this didactic material, we will explore the features and benefits of Cloud Spanner and how it can be applied in a self-based lab.

Cloud Spanner combines the advantages of a relational database structure with horizontal scale and performance. This simplifies application development and database management, allowing for faster app delivery. For globally distributed apps, Cloud Spanner's multi-regional configuration automatically replicates a database across continents, enabling localized reads and minimizing latency. Creating or scaling a globally-replicated database is a straightforward process that only requires a few clicks.

Google itself uses Cloud Spanner for its own mission-critical services and apps that billions of people access every day. This battle-tested database service boasts an industry-leading five nines of availability SLA, meaning it guarantees 99.999% availability. It also offers no planned downtime and enterprise-grade security.

If you are currently using a traditional relational database system that is struggling with scalability or relying on hand-rolled transactions on top of an eventually consistent database, Cloud Spanner could be the solution you are looking for.

To further understand and experience Cloud Spanner, you can participate in the Qwiklabs lab provided. The lab demonstrates how to use the GCP Console to create a Cloud Spanner instance, database, and table. It also covers adding a schema, writing data, modifying it, and running queries. The lab takes approximately 30 minutes to complete and provides hands-on experience with Cloud Spanner's features.

Cloud Spanner is a powerful database service offered by Google Cloud Platform that combines the benefits of a relational database structure with horizontal scalability. Its features simplify application development and database management, enabling faster app delivery. With its multi-regional configuration, Cloud Spanner allows for globally distributed apps with localized reads and minimal latency. Additionally, Cloud Spanner offers high availability, no planned downtime, and enterprise-grade security.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP LABS
TOPIC: SPEECH RECOGNITION USING MACHINE LEARNING

The Google Cloud Speech API is a powerful tool that allows users to convert audio into text. By utilizing this API, you can easily transcribe speech and determine if your suspicions about certain audio clips are correct. In just 15 minutes, you can learn how to use the Speech API through the Qwiklabs platform.

One of the great things about the Qwiklabs platform is that you don't need a Google Cloud Platform account or project to try it out. An account, project, and all necessary resources are provided to you as part of the Qwiklab experience. This means that you can dive right into learning how to use the Speech API without any barriers.

In the lab, you will create a file for the request to the Speech API. By sending audio to the API, you will receive a text transcription of the speech. The API also provides a confidence value, indicating how accurate the transcription is. This allows you to assess the reliability of the API's transcription.

The Speech API supports both synchronous and asynchronous speech-to-text transcription. In the example provided, a complete audio file was sent for transcription. However, you can also use the sync recognize method to perform streaming speech-to-text transcription while the user is still speaking.

Qwiklabs is an online hands-on lab library that offers a wide range of labs on various cloud topics, including Google Cloud. With over 150 labs available, you can learn new skills in just 30 minutes. Whether you are a beginner or an expert, Qwiklabs has labs suited to your level of expertise.

If you're interested in trying out the lab discussed in this material, you can access it through the provided link. Additionally, if you're ready to sign up for Google Cloud Platform, you can apply a $300 credit to your account using the provided link.

We value your feedback and would love to hear from you. Feel free to leave any questions or comments in the section below. We will address viewer questions on a weekly basis.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP LABS
TOPIC: PROCESSING TEXT WITH CLOUD NATURAL LANGUAGE

Cloud Natural Language is a Google Cloud Machine Learning API that allows us to derive insights from unstructured text. It can be used to extract entities, analyze sentiment, and categorize content. In this didactic material, we will explore the capabilities of Cloud Natural Language and demonstrate a self-paced lab where we extract entities from a snippet of text.

Textual data is abundant in various forms such as online reviews, social media posts, blogs, forums, emails, and call center communication. This data holds valuable information for businesses, providing insights into customer satisfaction, public perception, and product/service reception. However, analyzing such vast volumes of data manually is impractical for humans. This is where Natural Language Processing (NLP) comes in.

Google Cloud Natural Language API uses powerful machine learning models to reveal the structure and meaning of text. It offers an easy-to-use REST API that allows developers to leverage the same technology behind Google search and Google Assistant. With this API, you can perform syntax and sentiment analysis on text, extracting linguistic information and overall feelings expressed.

Entity analysis is another useful feature of Cloud Natural Language. It can identify known entities such as public figures, landmarks, organizations, and products. Additionally, sentiment analysis can be combined with entity analysis to determine positive, negative, or neutral sentiments associated with these entities.

For industries like media or publishing, where content categorization is essential, the Natural Language API can automatically sort documents and content into more than 700 predefined categories.

To demonstrate the capabilities of Cloud Natural Language, we will walk through a self-paced lab. In this lab, you will use the API to extract entities like people, places, and events from a given snippet of text. The lab provides step-by-step instructions on how to set up the API key, build the request, and analyze the entities. It is estimated to take approximately 40 minutes to complete.

By the end of this lab, you will have a better understanding of how to leverage the Cloud Natural Language API to extract valuable insights from unstructured text.

Thank you for joining us in this episode. We hope you enjoyed learning about Cloud Natural Language and its applications. If you have any cool ways to apply the Natural Language API, we would love to hear from you. Don't forget to check out the link provided to apply what you've learned using the $300 free trial credit on Google Cloud Platform.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP LABS
TOPIC: ANALYZING LARGE DATASETS WITH CLOUD DATALAB

Cloud Datalab is an interactive tool provided by Google Cloud that allows users to explore, analyze, and visualize large-scale datasets with ease. It offers a user-friendly interface and requires just a few clicks to perform these tasks efficiently.

One of the main advantages of Cloud Datalab is its integration with other Google Cloud Platform services. It runs on Google Compute Engine, which means users can choose a machine type that suits their data analysis needs in terms of performance and cost characteristics. Additionally, it seamlessly connects to other Google Cloud big data services, enabling users to analyze terabytes or even petabytes of data without any issues.

Cloud Datalab is primarily aimed at data scientists. It is built on Jupyter, an open-source platform that has gained popularity among data scientists for analyzing data. This means that Datalab users can leverage a wide range of open-source Python libraries for data analysis, visualization, and machine learning scenarios. They can easily import notebooks created by their peers, as Datalab runs on Google Cloud's infrastructure.

To demonstrate the capabilities of Cloud Datalab, a self-paced lab is provided. This lab takes approximately 30 minutes to complete and guides users through the process of creating a Datalab instance and a new notebook. It also covers using the web preview and command line tools to manage notebooks effectively.

In the lab, participants create a Cloud Datalab instance called "my-datalab" and access the Datalab home page through the web preview. They then create a new notebook, run code within it, and save the notebook for future use. The lab also shows how to use the command line tool "unget" to manage notebooks, commit changes, and push them to the master branch of the Cloud Datalab VM repository. Participants can view their commits using the command line by SSH-ing into the Datalab VM and opening an interactive shell session within the Cloud Datalab container.

Cloud Datalab is a powerful tool for data scientists to gain valuable insights from raw data. Its integration with other Google Cloud Platform services, scalability, and access to a wide range of Python libraries make it a valuable asset for data analysis, visualization, and machine learning scenarios.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP LABS
TOPIC: PERSONALIZATION OF G SUITE ADMIN

To personalize your G Suite Admin console according to your specific requirements, you can follow the steps outlined in this lab. The lab provides you with an organization called G Suite Labs and a temporary G Suite domain to work in. The objective is to make basic modifications to the G Suite Admin console and customize the company profile.

In the lab, you will start by removing and rearranging some controls on the Admin console dashboard. This allows you to tailor the console to your preferences. Next, you will customize the company profile by adding a support message. This message will be displayed to users when they are unable to sign into their G Suite account.

To ensure that the settings are relevant to your users, you will set the appropriate time zone. Additionally, you will select the "Manual" option for new products. This means that administrators will have to manually add new products for users to access them.

One of the tasks in the lab is to bulk add users using a CSV file. This simplifies the process of creating multiple user accounts. You will also have the opportunity to log in as one of the newly created users and experience the platform from a user's perspective.

If you are not familiar with Qwiklabs, it is an online hands-on lab library that is available 24/7. It offers over 150 labs covering various cloud topics, ranging from introductory to expert levels. Each lab is designed to help you acquire a new skill within approximately 30 minutes using Google Cloud.

When you are ready to take the lab we just completed, you can use the provided link. Furthermore, if you are interested in signing up for Google Cloud Platform (GCP), you can use the second link to apply a $300 credit to your account.

We value your feedback and encourage you to share any questions or thoughts in the comments section below. We will address a viewer's question each week.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP LABS
TOPIC: APACHE SPARK AND HADOOP WITH CLOUD DATAPROC

Cloud Dataproc is a managed Spark and Hadoop cloud service provided by Google Cloud Platform (GCP). It offers a faster, easier, and more cost-effective way to run Apache Spark and Apache Hadoop. In this didactic material, we will explore the features and benefits of Cloud Dataproc, as well as demonstrate a self-paced lab using the GCP console.

When using popular data processing tools like Hadoop and Spark, managing the balance between cost, complexity, scale, and utilization can be challenging. Cloud Dataproc aims to simplify this process by providing a managed service that takes care of the underlying infrastructure, allowing users to focus on the insights provided by their data.

One of the key advantages of Cloud Dataproc is its cost-effectiveness. It ensures that using Spark and Hadoop does not break the bank. In addition to the other GCP resources used, there is only a small incremental fee payable per virtual CPU in the cluster. Dataproc automation also helps save money by automatically turning off clusters when they are not needed. Billing is done in one-second increments, with a minimum of one minute, ensuring that users only pay for what they use.

To showcase the capabilities of Cloud Dataproc, a self-paced lab is provided. The lab utilizes the GCP console to create a Dataproc cluster, run a simple Apache Spark job, and modify the number of worker nodes in the cluster. The lab can be completed in approximately 30 minutes and offers a hands-on experience with Cloud Dataproc.

Furthermore, there is a separate lab available that allows users to complete the same activities using the G Cloud COI2. The lab provides flexibility in choosing the preferred method of interaction with Cloud Dataproc.

In the lab, participants will create a Dataproc cluster in the US central region and navigate to the Dataproc Jobs view. They will then submit a sample Spark job that calculates the value of pi and check the job output to see the calculated value. Additionally, participants will have the opportunity to change the number of worker instances in their cluster, further exploring the scalability of Cloud Dataproc.

Cloud Dataproc has proven to be a valuable tool for running Spark and Hadoop, and the team behind this didactic material encourages users to share their experiences and how Cloud Dataproc has improved their data processing workflows.

For those who have not yet signed up for the $300 free trial credit on GCP, a link is provided to take advantage of this offer. It is a great way to apply the knowledge gained from this material and explore the capabilities of Cloud Dataproc.

Cloud Dataproc is a managed Spark and Hadoop cloud service offered by Google Cloud Platform. It simplifies the process of running data processing tools by providing a cost-effective solution with automated features. The self-paced lab allows users to explore the capabilities of Cloud Dataproc using the GCP console or G Cloud COI2. Sign up for the free trial credit and take advantage of the additional training resources provided.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP LABS
TOPIC: QWIKILABS FOR GOOGLE CLOUD HANDS-ON PRACTICE

Qwiklabs is an online learning environment that provides a set of instructions to guide users through real-world, scenario-based use cases without requiring a Google Cloud Platform (GCP) account. Unlike simulation or demo environments, Qwiklabs offers access to the actual GCP environment. Users can access the lab environment from anywhere using a standard browser.

To take a lab on Qwiklabs, users need to create an account or sign in. Labs can be found by browsing the Qwiklabs Quest or by clicking on the Lab Catalog and then the Lab tab. The Lab tab provides a list of available labs that users can scroll through. By clicking on a lab of interest, users can access more information about it.

Each lab has a timer located next to the Start Lab button, indicating the total time that resources will be available for the lab. The lab will end after the timer finishes counting down. To begin a lab, users need to click the Start Lab button and enter the appropriate credits or token code. The cost of a lab is indicated in credits, with some labs being free and requiring no credits. One credit is equivalent to one dollar, and the number of credits required varies for each lab.

Once a lab is completed, users can click the End Lab button and leave a review. Additionally, Qwiklabs offers quests, which are collections of labs organized by technologies, specific cloud services, and practical use cases. Completing the required labs within a quest earns users badges that recognize their hands-on experience. These badges become a permanent part of their Qwiklabs account and profile, serving as a mark of distinction.

Users are encouraged to share their achievements on social media platforms like LinkedIn by adding a link to their badges. Qwiklabs provides numerous labs for users to choose from, and they can share their experiences and feedback in the comments section.

To sign up for Qwiklabs, viewers are offered a $300 credit for GCP by clicking on the provided link in the video description.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP LABS
TOPIC: CLOUD SDK ESSENTIAL COMMAND-LINE TOOLS

Cloud Computing - Google Cloud Platform - GCP labs - Cloud SDK essential command-line tools

Cloud SDK is a set of tools provided by Google Cloud Platform (GCP) to manage resources and applications on the cloud. It includes command-line tools like GCloud, bq, and GS to help developers and administrators interact with GCP. In this didactic material, we will explore the functionalities of Cloud SDK and how to use it effectively.

Google Cloud Console offers a browser-based UI to manage GCP products and services. However, Cloud SDK provides additional options and flexibility for managing resources through the command line. It allows users to script their actions, log and audit them, and access various GCP products, including App Engine, Compute Engine, Cloud Storage, and BigQuery.

One of the command-line tools available in Cloud SDK is bq, which is used for working with BigQuery. With bq, you can create and manage resources, load and query data, use external data sources, export data, and utilize the BigQuery data transfer service.

To get started with Cloud SDK, you need to install and initialize it on your operating system. The installation process varies depending on the OS, and you can find detailed instructions on the official Google Cloud Platform documentation.

In the lab demonstration, we learn how to install and initialize Cloud SDK on a virtual machine running Red Hat Enterprise Linux 7 or CentOS 7. The lab covers the steps of creating a VM, SSHing into it, updating the Cloud SDK RPM packages, and authenticating the SDK. Once initialized, we can list the accounts, view properties, and access GCloud command help.

Completing the lab will give you hands-on experience in using Cloud SDK and running core GCloud commands from the command line. The lab takes approximately 30 minutes to finish, and you can find the link to start the lab in the provided resources.

We hope you found this overview of Cloud SDK informative and useful. If you have any questions or want to share how you have applied the tools and functionalities of Cloud SDK, please leave a comment below. Don't forget to take advantage of the $300 free trial credit for GCP if you haven't already.

Additional training resources and links to further enhance your knowledge on Google Cloud Platform are provided below. Thank you for watching, and we look forward to seeing you again soon.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP LABS
TOPIC: POSTGRESQL AND MYSQL DATABASES WITH CLOUD SQL

Cloud SQL is a managed service provided by Google Cloud Platform (GCP) that simplifies the management of MySQL and Postgres databases. It allows users to offload time-consuming tasks such as patching, updates, backups, and replication configuration to Google, enabling developers to focus on building applications.

With Cloud SQL, users can easily set up and configure database instances in just a few steps. They can also choose the geographical region that best suits their needs to ensure optimal performance by keeping data close to the services that require it.

In this hands-on lab, we will demonstrate how to create and connect to a Cloud SQL MySQL instance and perform basic SQL operations using the GCP Console and the MySQL Client. The lab is self-paced and should take approximately 30 minutes to complete.

If you prefer using Postgres, don't worry, we have a lab specifically designed for you as well. Both labs offer step-by-step instructions and can be accessed through the provided links.

During the lab, we will enable the Cloud SQL API, create a Cloud SQL instance, and connect to it using the MySQL Client in Cloud Shell. We will then proceed to create a SQL database on the Cloud SQL instance, insert sample data into the guestbook database, and retrieve the data to verify the successful addition of the guests.

We hope you find this lab informative and enjoyable. Feel free to share your experiences or ideas on using Cloud SQL in the comments section below. If you haven't already done so, you can also take advantage of a $300 free trial credit on Google Cloud Platform to apply what you've learned.

For additional training resources and more information, please refer to the links provided below. Thank you for watching, and we look forward to seeing you soon.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP LABS
TOPIC: HELPING TO ORGANIZE WORLD'S GENOMIC INFORMATION WITH GOOGLE GENOMICS

The field of genomics, which focuses on studying the genome and its environments, has become data-rich due to advancements in DNA sequencing. As a result, the amount of genomic data available is growing exponentially. To handle the processing and analysis of such large-scale genomic data, cloud genomics offers a solution.

Google Cloud Platform (GCP) provides tools and services that enable the life science community to organize and securely access genomic information. In this material, we will explore how GCP helps in organizing the world's genomic information using Google Genomics.

One way GCP achieves this is through its implementation of the htsget protocol and SAMtools. These tools allow users to create projects using public genomic data. By enabling the genomics API and running specific commands in Cloud Shell, users can set up an htsget server and connect it to a local Docker container network.

With the htsget server in place, users can utilize sequence alignment map (SAM) tools to view statistics about specific genomic regions. For example, users can analyze a small range on chromosome 11 of a public genome. The power of GCP becomes evident as SAMtools processes over 1,500 reads in just a few seconds, which were streamed from a file stored in Google Cloud Storage. Previously, complex searches like these could have taken minutes, but with GCP, they now take as little as four seconds.

The speed and scalability provided by GCP's cloud genomics capabilities are instrumental in accelerating breakthroughs in understanding the causes and subtypes of various conditions. This knowledge can advance the fields of diagnosis and treatment in unprecedented ways.

To further enhance your understanding of Google Genomics, you can explore the Qwiklabs online lab library. Qwiklabs offers a variety of hands-on labs on different cloud topics, including over 150 labs related to Google Cloud. The Google Genomics Quickstart lab is one such lab that allows you to practice working with the Google Genomics API. By dedicating around 30 minutes to this lab, you can acquire new skills in utilizing Google Cloud for genomics.

If you are interested in getting started with GCP, you can sign up using the provided link, which will also grant you a $300 credit for your account. We value your feedback and encourage you to share any questions or thoughts in the comments section below. We will address viewer questions in future episodes.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP LABS
TOPIC: PROTECTING SENSITIVE DATA WITH CLOUD DATA LOSS PREVENTION

Cloud Data Loss Prevention (DLP) is a crucial aspect of protecting and managing sensitive data in any business. Mismanagement of such information can have severe consequences. The challenge lies in safeguarding sensitive data while still utilizing it for essential business functions like analytics and customer support operations. To address this challenge, Google Cloud Platform offers the Cloud Data Loss Prevention API.

The Cloud DLP API employs various techniques to identify sensitive data, such as credit card numbers, social security numbers, names, and personally identifiable information. It can identify sensitive data within text content as well as standard bitmap images. By applying the API to data streams, it becomes possible to automatically redact or censor sensitive information before it is stored, logged, or used for analysis. This upfront identification of sensitive data enables the selection of the most suitable storage system and appropriate access controls for that data.

The DLP API classifies raw data by utilizing predefined detectors that identify patterns, formats, and checksums. It can even understand contextual clues. Once the location of sensitive data is known, the API provides the option to deidentify that data. Deidentification involves removing identifying information from a dataset, making it harder to associate the remaining data with an individual and reducing the risk of exposure. The deidentified data can then be used for applications, storage, or analysis.

Redaction is another technique offered by the DLP API. It involves removing entire values or entire records from a dataset. Partial masking, on the other hand, hides parts of the data while leaving some data visible. For example, it can mask all but the last seven digits of a US telephone number. Tokenization, or secure hashing, replaces sensitive data with a key. This method is commonly used in credit card processing. Dynamic data masking applies deidentification and masking techniques in real time, allowing certain users to view masked data while others cannot.

The DLP API seamlessly integrates with various other services in the Google Cloud Platform. Cloud Functions can automate the classification of data uploaded to Cloud Storage. Built-in support is available for scanning and classifying sensitive data in Cloud Storage, Cloud Datastore, and BigQuery. Cloud Pub/Sub notifications can be generated in response to completed inspection jobs.

To illustrate the practical application of the DLP API, a self-paced lab is available. In this lab, users can set up the DLP API and use it to inspect a string of data for sensitive information. The lab demonstrates how to authenticate a service account, generate an authorization token, and utilize the curl command to inspect or deidentify JSON files containing sensitive information.

The Cloud Data Loss Prevention API provided by Google Cloud Platform offers powerful tools for protecting sensitive data. It enables the identification, redaction, and deidentification of sensitive information, allowing businesses to handle data securely while still utilizing it for essential operations. The API seamlessly integrates with other GCP services, further enhancing its capabilities.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP LABS
TOPIC: CONTAINER-OPTIMIZED OS

Container-Optimized OS is an optimized operating system image for Compute Engine VMs that is specifically designed for running Docker containers on Google Cloud Platform. It is Google's recommended operating system for container workloads. In this lab, we will learn how to create a container-optimized instance using both the Cloud Console and the command-line interface (CLI).

It is important to note that you do not need a Google Cloud Platform account or project to complete this lab. An account, project, and all necessary resources will be provided to you.

During this lab, you will gain hands-on experience in creating a VM with the container-optimized OS and deploying a Docker container using the CLI. Additionally, you will learn how to create a firewall rule to allow access to the VM and how to access the default Nginx page using the VM's external IP.

By the end of this lab, you will have the knowledge and skills to create a Compute Engine instance with the container-optimized OS and deploy a Docker container of your choice using both the GCP Console and the command-line interface.

Container-Optimized OS comes with all the necessary container-related dependencies pre-installed. This allows your cluster to easily scale up or down in response to changes in traffic or workload, optimizing your spending and improving reliability. Container-Optimized OS is the underlying operating system for various GCP services, including Kubernetes engines and Cloud SQL, making it a reliable solution for container workloads.

Qwiklabs is an online platform that offers hands-on lab exercises covering a wide range of cloud topics, from introductory to expert level. With over 150 labs available, Qwiklabs provides a valuable resource for learning new skills in about 30 minutes using Google Cloud.

To access the lab we just reviewed, please use the following link. If you are interested in signing up for GCP, you can also use this link to apply a $300 credit to your account.

We encourage you to ask any questions or share your thoughts in the comments section below. Each week, we will select a question from our viewers to answer in our upcoming episodes.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP LABS
TOPIC: MASSIVE WORKLOADS WITH CLOUD BIGTABLE DATABASE SERVICE

Cloud Bigtable is a powerful database service offered by Google Cloud Platform (GCP) that is used to handle massive workloads. It is the same database that powers many of Google's core services, such as search, analytics, maps, and Gmail. In this didactic material, we will explore the key features of Cloud Bigtable and demonstrate how to connect to a Cloud Bigtable instance, as well as read and write data in a table using a command line utility.

One of the notable features of Cloud Bigtable is its high performance under high load. This makes it ideal for large applications and workflows, as it enables faster, more reliable, and more efficient operations. Additionally, Cloud Bigtable is capable of storing large amounts of data with very low latency. It can automatically and seamlessly scale to handle billions of rows and thousands of columns, allowing for the storage of petabytes of data. Furthermore, users only pay for the amount of storage they actually utilize.

Cloud Bigtable is a fully managed service, which means that users do not need to worry about configuring and tuning their databases for performance or scalability. The service also provides backups of data to protect against catastrophic events and enable disaster recovery.

Cloud Bigtable offers an HBase compatible interface, which allows applications to seamlessly move between HBase and Cloud Bigtable. This compatibility enables flexibility and ease of migration for users who are already utilizing HBase.

As part of the GCP ecosystem, Cloud Bigtable can interact with other services and third-party clients. It ensures the security of data by encrypting it both during transit and at rest. Access to data in Cloud Bigtable is easily controlled through IAM permissions.

In the hands-on lab, users will utilize the CBT command line utility to connect to a Cloud Bigtable instance and perform read and write operations on a table. The lab provides step-by-step instructions on creating a Cloud Bigtable instance, configuring CBT, creating a table, adding column families, inserting data, and reading the data.

To get started with the lab, users can follow the provided link. The lab is self-paced and takes approximately 30 minutes to complete. It offers a practical and interactive way to apply the concepts learned in this didactic material.

Cloud Bigtable is a powerful and scalable database service offered by Google Cloud Platform. It provides high performance, low latency, and seamless scalability for handling massive workloads. With its HBase compatibility and integration with other GCP services, Cloud Bigtable offers flexibility and security for storing and managing large amounts of data.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP LABS
TOPIC: GOOGLE CLOUD VIDEO INTELLIGENCE

Google Cloud Video Intelligence is a powerful tool that allows you to search and annotate every moment of your video files. By using the easy-to-use REST API, you can extract metadata from your videos stored in Google Cloud Storage, making them searchable and discoverable. This lab will guide you through the process of creating a request file and calling the Video Intelligence API to create an operation for processing your request.

Once the operation is complete, you will be able to see your videos annotated, helping you identify key entities within the video. With a library of 20,000 labels, Cloud Video Intelligence automatically analyzes video content to identify entities and their appearance within the video. The tool does not require any machine learning or computer vision knowledge, making it accessible to users of all levels.

It's important to note that Cloud Video Intelligence improves over time as new concepts are introduced and accuracy is enhanced. This lab is part of a series called QuickStarts, which are designed to provide a glimpse into the various features available within Google Cloud. You can search for QuickStarts in the lab catalog to explore other labs that might interest you.

To get started with this lab, simply follow the link provided. Additionally, if you're interested in signing up for Google Cloud Platform (GCP), you can use the provided link to apply a $300 credit to your account. We value your feedback and encourage you to leave any questions or comments in the section below.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP LABS
TOPIC: RUNNING WORDPRESS ON APP ENGINE FLEXIBLE ENVIRONMENT

In this lab, we will introduce you to running WordPress on the App Engine flexible environment in Google Cloud Platform (GCP). WordPress is a popular open-source content management system, and GCP offers numerous benefits for hosting your website. By utilizing App Engine, your website can scale quickly and automatically to accommodate any number of viewers. Additionally, Google Cloud SQL and Google Cloud Storage provide managed infrastructure for hosting your database and files, eliminating the need for manual resizing or downtime requests.

To successfully complete this lab on QwikLabs, you will need a basic understanding of PHP and Linux text editors. The lab will guide you through enabling the necessary services, setting up App Engine and Cloud SQL to host your website and database separately, and utilizing Google Cloud Storage for your media library. By the end of the lab, you will have the knowledge to fully deploy a WordPress website on GCP.

Please note that you will be using a student or GCP account for these labs. However, if you prefer to use your own account, you can take advantage of a $300 credit to get started.

If you have questions related to GCP IRT core, specifically regarding sending data from an Arduino to GCP using MQTT, we have resources available to assist you. We provide starter open-source code on GitHub that can help you connect your Arduino devices to GCP. You can find the link in the description below for more information. Additionally, we offer a variety of tutorials on Google Cloud IRT Core, including an Internet of Things QuickStart lab. Click on the link in the description to access these resources and begin your learning journey.

We hope you find these materials helpful as you explore the possibilities of Google Cloud Platform.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP SECURITY
TOPIC: SECURING CLOUD ENVIRONMENT

Welcome to the "Cloud Security Basics" series, where we will delve into the intricacies of securing your application on Google Cloud. In this material, we will explore who holds the responsibility for securing various aspects of your cloud environment.

When it comes to securing your cloud environment, enterprise companies prioritize consistent delivery of the right service and data to the correct identity. This level of security must be maintained for every single request. Key considerations include implementing authorization and authentication to ensure only authorized individuals have access to resources and data, proactively preventing threats as bad actors continuously evolve, complying with industry regulatory requirements, and providing flexibility and control to internal teams.

Securing your system involves three main levels of responsibility: platform, infrastructure, and application-level security. As a user, you are responsible for securing your applications by setting up proper authentication, authorization, and identification for users in your system. Google Cloud takes ownership of securing the platform, which includes managing physical machines, data centers, and your application and data use. Infrastructure security is a shared responsibility, with Google Cloud providing tools to assist users in managing their virtual machines, networks, and data access needs.

To ensure the hardening of your cloud security, there are three main security actions you can take: platform and infrastructure actions, preventative actions, and forensic actions.

Platform and infrastructure actions involve securing the underlying hardware or virtual hardware. Platform security is entirely managed by Google and encompasses physical data center security and data replication across regions. Infrastructure security, on the other hand, is managed by the user, with Google Cloud offering tools for assistance. Users can modify settings in load balancers and select more secure VM instance types for their applications.

Preventative actions focus on avoiding breaches and involve locking down access controls. These actions primarily occur at the application level and utilize tools such as Google Cloud Identity and Access Management (IAM) and Google Identity Aware Proxy (IAP) to restrict access within the system. Google provides users with tooling to define access levels.

Forensic actions are taken to identify and stop breaches quickly or even prevent them from occurring. Logging and monitoring play a crucial role in this process. By logging activity and setting up automatic and manual monitoring, suspicious behavior can be detected and addressed promptly. Google Cloud also offers tooling to assist customers in monitoring their environments.

Securing your cloud environment is a shared responsibility model. While Google handles most infrastructure security, you are ultimately responsible for securing your applications and services. It is important to remember that no application or service can be assumed to be 100% secure. However, by leveraging detection-focused tools and planning for recovery, you can enhance the security of your cloud environment.

Securing Cloud Environment

In the field of cloud computing, security is of utmost importance to ensure the safety and integrity of your services and data. In this didactic material, we will discuss the basics of securing a cloud environment, with a focus on access control.

Access control is one of the three distinct areas of cloud security risk. It involves managing and regulating the identities that have access to your data. By implementing proper access control measures, you can prevent unauthorized individuals or entities from accessing your sensitive information.

To ensure the security of your cloud environment, it is crucial to have a well-defined security model in place. This model should include policies, procedures, and technologies that work together to protect your services and data. By carefully designing and implementing your security model, you can minimize the risk of

unauthorized access and potential security breaches.

One important aspect of access control is managing identities. It is essential to authenticate and authorize users, ensuring that only authorized individuals can access your cloud resources. Authentication involves verifying the identity of a user, typically through the use of passwords, biometrics, or multi-factor authentication. Authorization, on the other hand, determines the level of access a user has based on their authenticated identity.

In the event that the wrong identities gain access to your data, it is crucial to have mechanisms in place to detect and respond to such incidents. Intrusion detection systems, log monitoring, and incident response plans are examples of tools and processes that can help you identify and mitigate security breaches.

To further enhance the security of your cloud environment, it is recommended to regularly review and update your security measures. This includes staying informed about the latest security threats and vulnerabilities and applying patches and updates to your systems and applications.

Securing a cloud environment is a complex task that requires careful planning, implementation, and continuous monitoring. By focusing on access control and implementing a robust security model, you can significantly reduce the risk of unauthorized access and protect your services and data.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP SECURITY
TOPIC: TOP 3 RISKS - ACCESS

Access control is a crucial aspect of cloud security, as it ensures that only authorized individuals have access to the right resources within a system. Unauthorized access can lead to various security risks, such as man-in-the-middle attacks, phishing, and denial of service attacks. In this didactic material, we will explore how Google Cloud Platform (GCP) addresses these risks and provides robust access control measures.

To prevent man-in-the-middle attacks and distributed denial of service (DDoS) exploits, Google Cloud encrypts all internet access at the network level by default. This encryption eliminates the concern of passive adversaries listening for sensitive information. Additionally, GCP's load balancers support TLS termination, which provides an entry point to Google's massive serving resources while making successful DDoS attacks challenging.

Google Cloud also offers several tools and services to protect against unauthorized access. One of these tools is the Cloud Identity-Aware Proxy (IAP), which allows users to configure a central policy that requires authentication against a Google Group or G Suite domain. IAP is enforced at the network layer, minimizing the need for application code changes. This means that even legacy applications hosted on-premise behind a firewall can be migrated to a public endpoint on the cloud with little to no changes to the application itself.

To combat phishing, Google Cloud provides universal two-factor authentication (2FA) options. Users can utilize second factors such as one-time passwords or phishing-resistant security keys when signing in to enhance security. Google pioneered the U2F Titan Security Key, a hardware second factor that significantly reduces the effectiveness of phishing attacks. By pairing something the user knows (password) with something they have (security key), phishing becomes extremely difficult.

Endpoint management is another crucial aspect of access control. G Suite Endpoint Management allows organizations to manage the security of devices used by their employees to access company resources. This helps prevent unauthorized access in cases where an admin's laptop or personal device is compromised by malware.

Access control is a fundamental aspect of cloud security, and Google Cloud Platform offers a range of tools and services to address the top risks associated with access. These include encrypted traffic, load balancers, universal two-factor authentication, Cloud Identity-Aware Proxy, and endpoint management. By leveraging these tools, organizations can ensure that only authorized individuals have access to their resources, mitigating the risk of unauthorized access and potential security breaches.

Cloud Computing - Google Cloud Platform (GCP) Security - Top 3 Risks - Access

In the realm of cloud computing, ensuring the security of your company's data is of utmost importance. This becomes even more crucial when employees are allowed to use their personal devices for work purposes. Google Cloud Platform (GCP) offers a range of security measures to protect access to your data, allowing you to strike a balance between convenience and safety.

One such feature provided by Google Cloud is Identity-Aware Proxy. This tool helps secure access and authentication by building up context associated with the access, ensuring that it adheres to the established rules. For example, access should ideally come from a Chromebook, the user should possess proper credentials to log in, they should have the required access levels, and a hardware second factor is necessary to access specific APIs. These individual protections layer on top of each other, creating a robust security framework for your cloud environment.

Endpoint verification is another valuable tool offered by Google Cloud. It simplifies the process of setting up policies, such as separating work apps from personal apps on Android devices. Additionally, Chrome OS, designed with security in mind, prevents the installation of unauthorized software. Admin access should only be granted if the admin is using a Chromebook, further bolstering security measures.

Universal two-factor authentication (2FA) is yet another layer of protection provided by Google Cloud. By requiring a second factor, such as a hardware token or a mobile app, in addition to the traditional username and

password, the risk of unauthorized access is significantly reduced.

By employing these security features, you can enhance the safety and security of your cloud environment, ensuring that only authorized individuals can access your company's data. Google Cloud's comprehensive approach to access security helps mitigate the risks associated with unauthorized access.

Protecting access to your data is essential in cloud computing. Google Cloud Platform offers a range of security measures, including Identity-Aware Proxy, endpoint verification, and universal two-factor authentication, which collectively work to strengthen access security. By implementing these measures, you can safeguard your cloud environment and mitigate the risks associated with unauthorized access.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP SECURITY
TOPIC: TOP 3 RISKS - DATA

Welcome to this educational material on the topic of Cloud Computing - Google Cloud Platform (GCP) security. In this material, we will focus on the top 3 risks related to data security in the context of cloud computing.

Data security is a critical aspect of cloud computing, and it involves protecting the data that is stored and processed in the cloud. The risks associated with data security can have severe consequences, including data breaches, unauthorized access, and loss of sensitive information.

One of the significant risks related to data security is improper disclosure of information by employees. This can happen when employees accidentally send emails or other communications containing sensitive data such as credit card or social security numbers. To mitigate this risk, it is essential to have proper training and awareness programs for employees to ensure they understand the importance of data protection.

Another risk is the storage of data without proper security protocols in place. If data is stored in a location that lacks adequate security measures, it becomes vulnerable to unauthorized access or data breaches. It is crucial to ensure that data is stored in secure environments that comply with industry standards and best practices.

The third risk is related to the transfer and storage of data. It is essential to have mechanisms in place to track and monitor the movement of data within the cloud environment. Losing data or being unable to locate it can have significant consequences for businesses and their customers. Implementing tools and services that provide visibility and control over data transfer and storage can help mitigate this risk.

Google Cloud Platform offers a range of tools and services to help protect data and mitigate these risks. Some of these tools include:

1. Identity and Access Management (IAM): IAM allows granular access control to specific resources, preventing unauthorized access to sensitive data. It follows the principle of least privilege, where only necessary permissions are granted to access specific resources.

2. Encryption: Google Cloud Platform provides encryption capabilities to ensure that stored and transferred data cannot be read, even if it is stolen. Encryption adds an extra layer of protection to sensitive data.

3. Logging and Monitoring: Google Cloud Platform offers tools such as Google Cloud Logging and Google Cloud Monitoring, which enable the collection and analysis of request logs. These tools help track who is accessing data and provide alerts in case of any suspicious activity.

4. Data De-identification (DeID): DeID ensures that personally identifiable information (PII) is stripped before it is stored in the system. This helps protect sensitive user information and reduces the risk of data breaches.

5. Organizational Policy: Setting up organizational policies provides a centralized configuration of restrictions on how resources can be used. These policies define guardrails for development teams and help ensure compliance with data security regulations.

By leveraging these tools and services provided by Google Cloud Platform, businesses can enhance their data security and mitigate the risks associated with storing and processing data in the cloud.

Data security is a crucial aspect of cloud computing, and businesses must be aware of the risks and take appropriate measures to protect their data. Google Cloud Platform offers a range of tools and services to help businesses ensure the security of their data in the cloud.

Google Cloud Platform (GCP) provides various tools to ensure the security of data stored in the cloud. These tools include IAM (Identity and Access Management), encryption, logging and monitoring, and organizational policy. By utilizing these features, customers can protect their data, control access to it, and easily locate stored data.

IAM allows customers to manage user access to resources within their GCP projects. This ensures that only authorized individuals can access and manipulate data. Encryption is another important security measure provided by GCP. It ensures that data is encrypted both at rest and in transit, making it difficult for unauthorized parties to access sensitive information.

Logging and monitoring tools enable customers to track and analyze activities within their cloud environment. This helps in identifying any suspicious or unauthorized access attempts and provides insights into potential security breaches. Organizational policies allow customers to define and enforce security rules across their entire organization, ensuring consistent security practices.

Securing data in the cloud is crucial, as it protects sensitive information from unauthorized access and potential data breaches. With GCP's robust security features, customers can have peace of mind knowing that their data is well-protected.

In the next episode of Cloud Security Basics, the focus will be on securing the platform, which is the last of the three distinct areas of cloud security risk. Stay tuned for more insights on how to secure your virtual and physical hardware in the cloud.

If you want to explore the topic of securing data in the cloud further, you can check out the article linked in the description below. Remember, when it comes to security, it is essential to stay vigilant and not let bad actors compromise your data.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP SECURITY
TOPIC: TOP 3 RISKS - PLATFORM

The platform is one of the three distinct areas of cloud security risk. It represents the physical systems that house and deliver information. It's important to understand that the abstract notion of a user accessing information is realized in physical systems, and the sum total of those systems is the platform. Google is responsible for ensuring the physical and virtual hardware they provide is operating as it should.

Google Cloud Services are designed to deliver better security than many traditional on-premise solutions. Google uses a multilayered defense in depth approach to security, with strict controls for access and privilege at each layer. This approach includes physical data center components, hardware provenance, secure boot, secure interservice communication, secure data, and protected access to services from the internet. Each of these layers is continually evolving and improving.

To protect the physical and virtual hardware, Google Cloud has a strong focus on security. They have over 850 security engineers, invest $200 billion annually in security, maintain a 24/7 active watch, and have published over 160 academic research papers on security. They also have a bug bounty program and a penetration testing program. Google's security teams triage, investigate, and respond to incidents around the clock, and they conduct regular exercises to measure and improve security detection and response.

On the application side, Google Cloud provides libraries and frameworks that prevent developers from introducing certain classes of security bugs. Automated tools like fuzzers, static analysis tools, and web security scanners are used to detect security bugs. On the access side, Google makes heavy investments in protecting their employees' devices and credentials from compromise. They have technologies and strict policies for physical computer data and network security, access management, security login, and more.

Google Cloud's platform is secure because security is foundational to everything they do. They design their data centers, software, and processes with security in mind. Since Google Cloud runs on the same infrastructure they provide to customers, all of these protections can be used by organizations to protect their business.

In the next episode, we will focus on how Google Cloud protects the data that resides on their servers.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP SECURITY
TOPIC: SECURING CUSTOMER DATA

Cloud Computing - Google Cloud Platform - GCP Security - Securing Customer Data

In this didactic material, we will explore how Google Cloud Platform (GCP) ensures the security of customer data. Securing data at rest and in transit is crucial for maintaining the integrity and confidentiality of sensitive information.

To protect customer data, Google Cloud logically isolates each customer's data from that of others, even when stored on the same physical server. This ensures that data remains private and secure, and it also prevents employees from abusing their access privileges. Only a small group of authorized Google employees have access to customer data, and Google does not scan or sell customer data to third parties. Furthermore, if customers choose to delete their data, Google commits to removing it from their systems within 180 days. Additionally, Google provides tools that enable customers to easily transfer their data if they decide to stop using Google services.

When it comes to data in transit, Google Cloud employs various measures to protect against interception. Data stored in Google Cloud is encrypted before it is written to physical storage. Encryption is performed at the application layer using keys from a central key management service. This approach isolates the infrastructure from potential threats at lower levels of storage. Hardware encryption and other protective layers are also utilized.

As data leaves Google's secure servers and travels across the public internet, it must traverse multiple devices, known as hops, which introduces potential vulnerabilities. However, Google's global network, connected to most ISPs worldwide, reduces the number of hops and improves the security of data in transit. All traffic is routed through custom Google front end servers, which detect and prevent malicious requests and distributed denial of service attacks. These servers are restricted to communication with a controlled list of internal servers, enhancing security.

Google employs cryptography to ensure the privacy and integrity of data during transit. Cryptographic features are encapsulated within Google Cloud RPC mechanisms, making them available to other application layer protocols. This provides application layer isolation and reduces dependence on the security of the network path. Encrypted interservice communication remains secure even if the network is tapped or compromised.

In addition to encryption and cryptographic measures, Google implements industry-standard firewalls and access control lists (ACLs) to ensure network segregation. This adds an extra layer of protection to sensitive networks.

Google's infrastructure also incorporates DDoS and man-in-the-middle protections, which further safeguard customer data. These protections apply to data stored in Google's data centers, ensuring the safety of data throughout its lifecycle.

Google Cloud Platform employs a comprehensive set of security measures to protect customer data. This includes logical isolation, limited employee access, encryption at rest and in transit, custom networking hardware, interservice encryption, ACLs, and industry-standard firewalls. These measures ensure the confidentiality, integrity, and availability of customer data within the Google Cloud Platform.

Cloud security is a critical aspect of ensuring the safety and confidentiality of customer data in the Google Cloud Platform (GCP). In this episode, we will discuss the importance of physical security in protecting customer data.

Physical security refers to the measures put in place to safeguard the physical infrastructure that houses data centers and hardware. Google Cloud recognizes the significance of physical security and has implemented various measures to ensure the protection of customer data.

One of the key components of physical security is access control. Google employs multiple layers of access control mechanisms to restrict unauthorized entry into its data centers. These measures include biometric

authentication, security badges, and strict access policies. Only authorized personnel are granted access to the data centers, and their activities are closely monitored.

Another important aspect of physical security is surveillance. Google Cloud data centers are equipped with advanced surveillance systems, including CCTV cameras and motion sensors, to detect and deter any unauthorized activities. These systems are monitored 24/7 by security personnel to ensure the safety of the infrastructure.

Furthermore, Google Cloud places a strong emphasis on environmental controls. Data centers are designed to withstand various environmental threats, such as fire, floods, and earthquakes. Fire suppression systems, redundant power supplies, and backup generators are in place to minimize the risk of service interruptions and data loss.

In addition to these measures, Google Cloud also implements rigorous security protocols for the transportation and disposal of hardware. This ensures that customer data remains protected even when hardware is being moved or retired.

It is essential for organizations to have confidence in the security of their data when utilizing cloud services. Google Cloud's commitment to physical security demonstrates its dedication to safeguarding customer data. By implementing robust access controls, advanced surveillance systems, and environmental controls, Google Cloud ensures the highest level of protection for customer data.

Physical security plays a vital role in securing customer data in the Google Cloud Platform. By employing comprehensive access control mechanisms, advanced surveillance systems, and environmental controls, Google Cloud ensures the safety and confidentiality of customer data.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP SECURITY
TOPIC: SECURING HARDWARE

Physical security is a crucial aspect of securing data centers and hardware in cloud computing. Google Cloud takes extensive measures to ensure the physical defense of their data centers and protect the physical devices that run software.

Google limits access to their data centers to a small number of specially qualified employees. Less than 1% of Google employees have access to the data center floor, and access is only possible through a security corridor that implements multi-factor access controls using security badges and biometrics.

In terms of securing the virtualization and hardware components, Google has developed custom tooling to protect its users. Google data centers have thousands of server machines connected to a local network, providing an initial layer of security. The server boards and networking equipment are custom designed to adhere to Google's strict security requirements.

Google also uses the Titan hardware security chip, which can be deployed on servers and peripherals. This chip allows Google to identify and authenticate devices at the hardware level, establishing a strong identity for each machine. The Titan chip offers integrity verification of firmware and software components, ensuring a true audit trail of any changes made to the system. It also provides tamper-evident logging capabilities to identify actions performed by insiders with root access.

Additionally, Google utilizes the kernel-based virtual machine (KVM) as the foundation for Google Compute Engine and Google Kubernetes Engine. KVM is an open source virtualization technology built into Linux, allowing one host machine to run multiple isolated virtual machines. Google invests in additional security hardening and protections for KVM, including thorough code reviews and proprietary fuzzing tools to test its security.

By implementing these measures, Google Cloud ensures the physical security of their data centers and hardware, providing a secure environment for running software and protecting user data.

Google Cloud Platform (GCP) takes security seriously and employs various measures to protect its hardware and ensure the security of its services and customers. One of these measures is the use of KVM (Kernel-based Virtual Machine) for virtualization, which provides simplicity, better testing, and significant security advantages. By leveraging hardware, virtualization, and physical security, Google is able to safeguard its tooling effectively.

The security of the platform starts with the data center, which is the physical hardware that runs the software. Google's commitment to security-first design ensures that the platform remains secure. They manage security throughout the data lifecycle, from the data center to the device, by employing a range of technologies and approaches.

While it is important to have the basics covered, it is crucial to be aware of potential threats. Forensic and preventative actions play a significant role in maintaining a secure environment. Google Cloud continues to address these aspects and will provide further insights in upcoming episodes.

To learn more about how Google secures its data centers and physical hardware, you can refer to the linked article in the description. Stay tuned for future episodes where we will delve into preventative and forensic actions. Remember, in cloud security, it is essential to stay vigilant and not let bad actors compromise your data.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP SECURITY
TOPIC: CLOUD ARMOR

Cloud Armor is a web application firewall and distributed denial of service (DDoS) mitigation service provided by Google Cloud. It offers layer 7 protection and filtering for workloads deployed on Google Cloud Platform (GCP), on-premises, or with other infrastructure providers.

Cloud Armor is deeply integrated with the global load balancing infrastructure and is able to inspect and filter incoming requests after SSL termination has occurred. It allows customers to protect their HTTP-fronted applications from DDoS attacks and filter incoming requests based on various parameters such as geography, request headers, or cookies.

As a web application firewall, Cloud Armor comes with pre-configured rules to prevent common attacks and vulnerability exploit attempts. It also provides real-time telemetry in the form of logs sent to Cloud Logging, which contains Cloud Armor's decisions on a per-request basis. Additionally, there is a monitoring dashboard that provides granular views of allowed, denied, or previewed traffic, as well as correlated web application firewall (WAF) security findings sent to the Cloud Security Command Center.

In March of this year, a rich set of WAF capabilities for Cloud Armor was made generally available. Preconfigured rules can help mitigate the OWASP top 10 risks, and the mod security core rule set has been ported over, introducing rules to detect and block SQL injection and cross-site scripting attempts.

Cloud Armor works in conjunction with other key network security controls provided by Google Cloud. These controls include Cloud Load Balancing, Identity Aware Proxy, Firewalls, VPC Service Controls, packet mirroring, Cloud NAT, and various interconnected VPN options. Together, these controls enable customers to follow a defense-in-depth approach and deploy security controls at various levels of their stack and infrastructure to enforce access controls and ensure the privacy and security of their data and mission-critical workloads.

Cloud Armor is a powerful web application firewall and DDoS mitigation service offered by Google Cloud. It provides layer 7 protection and filtering, allowing customers to protect their applications from DDoS attacks and filter incoming requests based on various parameters. With pre-configured rules and real-time telemetry, Cloud Armor helps mitigate common attacks and provides insights into the security of the application. It works in conjunction with other network security controls provided by Google Cloud to offer a comprehensive security solution.

Cloud Armor is a security feature provided by Google Cloud Platform (GCP) that offers advanced protection against various types of attacks, including DDoS attacks. It allows customers to enforce access controls based on the source geography of each request, using IP-to-geo mappings sourced from Google's own geo team, ensuring accuracy.

One of the key features of Cloud Armor is its extensible rules language, which enables users to configure custom layer 7 filtering policies across request headers, request parameters, and cookies. This allows for fine-grained control over the filtering of incoming requests, enhancing the security of applications and websites.

Cloud Armor is integrated with the Security Command Center (SCC), providing customers with visibility into potential attacks against their protected applications and websites. Cloud Armor findings and assets are sent to the SCC dashboard to alert defenders and facilitate prompt action against potential threats.

Recent updates to Cloud Armor have significantly increased its flexibility and coverage. It now supports the protection of an expanded set of customer infrastructure on GCP, as well as hybrid use cases located on-premises or in other cloud providers. Cloud Armor can protect cloud Content Delivery Network (CDN) origin servers by enforcing security policies on dynamic requests and cache misses destined for the CDN origin server.

Cloud Armor also helps mitigate computationally expensive cache busting attacks and protects dynamic portions of websites and applications from the OWASP Top 10 risks. Enterprises can enforce a consistent set of security controls for their applications, regardless of whether they are deployed on GCP or in permanently hybrid configurations.

Another notable feature is the support for internet network and point groups, which allows customers to leverage Google's Edge infrastructure, including cloud load balancers, Cloud CDN, and Cloud Armor, to protect their websites or applications hosted anywhere.

For users of Google Kubernetes Engine (GKE), Cloud Armor provides GKE ingress support, allowing containerized workloads to be protected by placing them behind cloud load balancers and configuring the Cloud Armor security policy for layer 7 filtering and Web Application Firewall (WAF) use cases.

Cloud Armor also introduces the capability to allow or deny traffic through a security policy based on a pre-configured, named IP list. This feature is particularly useful for customers who receive traffic into their GCP projects from upstream service providers, such as other CDNs. By referencing the named IP lists, customers can configure a security policy to deny all traffic from the internet by default and allow only traffic from desired IP ranges.

To further enhance the protection offered by Cloud Armor, Google has launched Cloud Armor Managed Protection, a set of DDoS mitigation and WAF services offered at two service tiers: standard and plus. This service bundles together all the features and capabilities of Cloud Armor with additional value-added services, providing enterprise-friendly and predictable monthly subscriptions. With Cloud Armor Managed Protection, customers can leverage Google's Edge capacity and DDoS defense expertise to protect their applications and other publicly exposed workloads.

Cloud Armor Managed Protection includes rules, policies, and requests in the subscription plan, ensuring a relatively fixed monthly price even in the face of high-volume layer 7 DDoS attacks that require mitigation by Cloud Armor. Google plans to expand the services offered in the plus tier, starting with Google-curated rule sets like the named IP lists.

Cloud Armor is a powerful security feature provided by Google Cloud Platform. It offers advanced protection against various types of attacks, including DDoS attacks, and provides customers with fine-grained control over filtering policies. Cloud Armor is integrated with the Security Command Center, supports a wide range of deployment scenarios, and offers additional features through Cloud Armor Managed Protection.

DDoS protection is a critical aspect of cloud security, and Google Cloud Platform (GCP) offers robust measures to ensure the availability of its services. The DDoS protection provided to GCP customers is the same that Google has developed and refined over the past two decades to protect its own services. Google's global network allows for the absorption, dissipation, and mitigation of layer 3 and layer 4 network or volumetric attacks across various components in its global load balancing infrastructure.

Automatic mitigation is a key feature of Google's DDoS protection. All three types of global load balancers in GCP only proxy requests back to the customer backend service after the request has completed a three-way TCP handshake. For volumetric and protocol-based DDoS attacks, such as UDP amplification or reflection, as well as TCP floods, the TCP handshake is never established, allowing Google to drop this unwelcome traffic far upstream of the customer's infrastructure.

In addition to automatic mitigation, GCP provides Cloud Armor security policies that can be configured and attached to load balanced backend services to further enhance layer 7 application layer protection and access controls. These security policies can limit access based on source IP or geographical location, utilize pre-configured Web Application Firewall (WAF) rule sets, and employ customizable rules to craft custom layer 7 filtering policies.

Cloud Armor security policies offer granular control over access to protected resources. They can simultaneously invoke pre-configured WAF rules and user-defined rules to inspect request headers, parameters, and cookies. These policies are stored, evaluated, and enforced at the edge of Google's network, far upstream of the customer's infrastructure.

To illustrate the use of Cloud Armor security policies, consider an example policy that denies access to external clients attempting to access the admin portal of an application. The policy can also invoke pre-configured WAF rules to detect and block known signatures for SQL injection and cross-site scripting attacks. If the request does not target the admin portal or contain any SQL injection or cross-site scripting signatures, the traffic is allowed

as per the default rule.

GCP provides customers with various use cases for protecting their applications. To safeguard against volumetric and protocol-based DDoS attacks, deploying a global load balancer in front of the HTTP or TCP workload is sufficient. Cloud Armor, in conjunction with the load balancer, automatically mitigates DDoS attacks such as DNS amplification attacks, SYN floods, and other common layer 3 and layer 4 DDoS attacks. Only well-formed layer 7 requests that have completed the three-way handshake are proxied back to the applications.

For layer 7 protection, customers can configure a Cloud Armor security policy and attach it to the backend service hosting the application or workload to be protected. These policies allow for customization of access to protected resources. Each policy consists of a prioritized list of rules and a default rule. As an incoming request reaches the customer backend service, Cloud Armor evaluates each rule in priority order. The first matching rule determines the action to take, whether to allow or deny the traffic. If the traffic does not match any rules, the action configured with the default rule is applied.

Visibility and telemetry are crucial for a comprehensive application protection solution. Cloud Armor provides near real-time per-request logs that capture all decisions made by the security policies regarding layer 7 requests, including which rules fired and why. Real-time telemetry for request volumes is available through Cloud Monitoring, allowing users to visualize and create learning policies based on changes in traffic patterns. Furthermore, correlated security findings about unexpected traffic spikes are sent to the Security Command Center to trigger investigation and incident response workflows.

In complex use cases, Cloud Armor security policies can be dynamically updated using GCP's feature-rich REST API or CLI. This flexibility allows for sophisticated application protection strategies, ensuring that security policies remain up to date.

Google Cloud Platform offers robust DDoS protection through automatic mitigation and Cloud Armor security policies. These measures safeguard against volumetric and protocol-based DDoS attacks, provide layer 7 application layer protection, and offer granular access controls. Visibility and telemetry features enable users to monitor and respond to security events effectively.

Telemetry data is an essential component of monitoring and analytics workflows in cloud computing. This data can be collected through various means, including customer-built solutions, commercial off-the-shelf tools, or by utilizing the native data analytics tools provided by Google Cloud Platform (GCP), such as BigQuery.

In addition to telemetry data, other sources of information like application logs, data from Cloud Armor, and network devices are incorporated into the system. This diverse range of data allows for a comprehensive view of the environment and enables effective threat detection and fraud prevention.

Threat detection algorithms play a crucial role in analyzing the collected telemetry data. These algorithms correlate the different data sources to identify patterns and generate signatures that indicate malicious traffic. These signatures are then used to create new rules in the Cloud Armor security policy. By doing so, newly detected malicious behavior can be swiftly blocked at the edge of Google's network.

Telemetry data, along with threat detection algorithms and Cloud Armor, form a robust security framework within the Google Cloud Platform. This framework enables the detection and prevention of malicious activities, ensuring the safety and integrity of cloud-based systems.

For more information, please refer to our recently published blogs or visit our product page. If you have any specific questions, feel free to reach out to us. We are here to provide detailed answers and assist you further.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP SECURITY
TOPIC: DATA CENTER SECURITY LAYERS

A Google data center is equipped with six layers of security to protect customer data. The first layer is the property boundaries, which include signage and fencing. The second layer, known as the secure perimeter, features smart fencing, overlapping cameras, 24/7 guard patrols, and more. Behind the scenes, there are guards in vehicles and on foot, as well as a vehicle crash barrier to prevent unauthorized access.

The third layer is building access, where visitors must go through a secure lobby. Here, authentication is done using an ID card and iris scan to ensure the person is who they claim to be. Only one person is allowed to badge through a door at a time in secure areas.

Layer four is the security operations center (SOC), which monitors the data center round the clock. The SOC is responsible for overseeing the doors, cameras, badge readers, and iris scan. Any suspicious activity is immediately detected and addressed.

The data center floor, layer five, is strictly limited to authorized technicians and engineers who maintain, upgrade, or repair the equipment. Data at rest is encrypted, and customers can manage their own encryption keys to ensure the privacy and security of their data.

The final layer, layer six, is the most restricted area. It is where disks are erased and destroyed. Only a select few technicians have access to this area. Disks that need to be retired are placed in a secure two-way locker system, and only authorized technicians can retrieve them for erasure or destruction. The destruction process involves using a crusher to ensure the disk is completely destroyed.

In addition to these six layers of security, Google Cloud has two security testing programs. One program hires external companies to attempt to break into data center sites from the outside, while the other program tasks internal employees with trying to break security protocols from the inside. This comprehensive approach ensures the highest level of security for customer data.

Google Cloud Platform (GCP) is known for its robust security measures, particularly when it comes to data center security. In order to ensure the highest level of protection, GCP implements multiple layers of security protocols.

One of the key security measures in GCP data centers is the requirement for individuals to pass through full metal detection every time they leave the data center floor. This strict access control ensures that only authorized personnel can enter or exit the data center. By implementing this measure, GCP aims to prevent unauthorized access to sensitive information and infrastructure.

In addition to physical security measures, GCP also places a strong emphasis on compliance with global standards, regulations, and certifications. With over 40 compliance certifications, GCP demonstrates its commitment to meeting the highest security standards in the industry. By adhering to these standards, GCP ensures that customer data is protected and that the platform is in line with industry best practices.

Furthermore, GCP continuously tests, optimizes, and improves its systems to stay ahead of emerging security threats. This commitment to ongoing improvement makes GCP a leader in data center security. By regularly updating security protocols and implementing the latest technologies, GCP ensures that its infrastructure remains secure and reliable.

GCP's data center security is built upon multiple layers of protection, including strict access control, compliance with global standards, and continuous system optimization. These measures work together to safeguard customer data and maintain the platform's reputation as a leader in cloud security.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP SUPPORT
TOPIC: GETTING SUPPORT WITH GOOGLE CLOUD CUSTOMER CARE

Cloud Computing - Google Cloud Platform - GCP Support - Getting Support with Google Cloud Customer Care

Cloud support is crucial for startups as it provides assistance across various aspects of their journey, including product usage, technical help, feature requests, unexpected product behavior, and billing and administration questions. Google Cloud aims to offer an incredible experience and comprehensive support to startups building on their platform.

To meet customer needs and deliver a better experience, Google Cloud has re-envisioned their customer care portfolio. The vision places customers at the center of the model, providing a flexible service that allows startups to choose the right support offering for their business needs. The goal is to establish an ongoing partnership, ensuring that all questions are answered.

The Google Cloud customer care portfolio offers a scalable set of offerings tailored to startup needs. The core offerings include basic, standard, enhanced, and premium support. Additionally, there are value-added services available for enhanced and premium support.

Basic support is provided at no cost when signing up for Google Cloud and is available for billing-related issues. Standard support is a paid offering recommended for small and medium enterprises. It enables startups to easily build, troubleshoot, and test their workloads on Google Cloud, with a four-hour response time for high-impact cases and an overall availability of 8/5.

Enhanced support delivers rapid response times and additional services to boost productivity and ensure efficient Cloud operations. It offers both case and phone support for technical issues, with a one-hour response time, 24/5 availability, and 24/7 support for critical issues. Startups can also access add-ons like technical account advisor service and event management.

The highest value support offering is premium support, which provides incredibly fast response times. It includes features such as a named technical account manager, new product previews, operational health reviews, training resources, and an event management service. Premium support offers case and phone support for technical issues, third-party technology support, a 15-minute response time for critical impact cases, and 24/7 support for critical impact issues.

To purchase support within the Google Cloud Console, startups can go to their dashboard, access the Navigation menu, hover over Support, and click on Overview. From there, they can view the support offerings and select the desired support tier. They will need to choose the appropriate resources and billing account before agreeing to the terms of service.

Google Cloud provides a range of support options to meet the needs of startups. From basic support for billing-related issues to premium support with fast response times and additional services, startups can choose the level of support that aligns with their requirements.

To get support with Google Cloud Customer Care, you can follow these steps:

1. To purchase a support package, go to your console and navigate to the Support page as a support user.
2. Depending on your support package, you can create cases through various channels.
3. In the Cases tab, you can see a list of previously created cases.
4. To create a new case, click on Create Case and complete the required fields.
- Give your case a title.
- Select a priority ranging from P1 to P4, with P1 being the most critical.
- Choose a category and a component.
- Fill out the description. A description template is provided based on the selected category and component.
5. On the right-hand side of the screen, you will find a help assistant providing useful articles relevant to your case.
6. Finally, click Submit to submit the case.

7. After submitting the case, you can make edits to the attributes, add comments, and upload attachments on the case page.

This concludes the overview of support packages. For more information about customer care and support at Google Cloud, please refer to the links in the description box below.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP SUPPORT
TOPIC: GCP SUPPORT CASE BEST PRACTICES

In Google Cloud Engineering Support, our main objective is to collaborate closely with you, the engineer, to resolve any issues you may encounter while utilizing Google Cloud Platform (GCP) products. This didactic material aims to provide you with best practices for filing an issue report with our support engineers in the Google Cloud Support Center. These practices are also applicable when seeking technical assistance from any GCP engineer, including support cases, bug reports, and issue trackers, as well as posts to user groups and forums such as Stack Overflow.

The key principle when reporting an issue is clarity. It is crucial to specify the right level of technical detail and explicitly communicate your expectations. By doing so, we can better assist you in resolving the issue. The Support Center's help page and previous materials provide guidance on how to file cases, offering insights into what information to include and why it is important. In this didactic material, we will explain the specific details we require and why they are significant.

There are four critical details that should be included with every case:

1. Specific times when you experienced the issue: Providing the onset time and duration allows us to focus our time series monitoring on the relevant period. Please be explicit about whether the issue is ongoing or if it was only observable in the past. If the issue is not ongoing, kindly state that and provide the end time if known. It is recommended to use the ISO 8601 format, as it is unambiguous and easy to sort. Additionally, always include the time zone. Our internal systems typically operate in Google time, which is US-specific, but our agents follow a "follow the sun" model and may be located in different time zones.

2. GCP products being used: Clearly specify the GCP products involved in the issue. This information helps us locate the components or logs necessary for diagnosing the problem. Be as specific as possible, referencing the specific APIs or console.cloud.google.com, and consider including screenshots or linking to the relevant documentation page.

3. Location: Include the location information, particularly the region and zone. Rollouts of changes often occur on a region or zone basis, and these details help us identify if a rollout is underway or map it to an internal release ID for bug reporting purposes. For example, mentioning "I tried regions us-east1 or us-central1" provides valuable context.

4. Specific identifiers for relevant resources: It is essential to include specific identifiers for resources related to your case. The project ID is a required field when filing a case and serves as an input for most troubleshooting tools. Please provide the numeric project ID, not just the project name. If the error is observed in multiple projects or in one project but not another, include that information in the description. Additionally, include IDs of other objects such as instance IDs, BigQuery job IDs, table names, or IP addresses. IP addresses act as unambiguous identifiers, so when specifying a cloud platform IP, provide the context of how it is used (e.g., connected to a GCE instance, a load balancer, a custom route, or an API endpoint).

Following these best practices when filing an issue report will significantly improve the efficiency and effectiveness of our support process. By providing the necessary details, you enable our support engineers to better understand and diagnose the problem, ultimately leading to a quicker resolution.

When troubleshooting a connection issue in Google Cloud Platform (GCP), it is important to provide specific information about the IP addresses involved. This includes details about whether it is your home internet, a VPN endpoint, or an external monitoring system. General statements like "one of our instances" or "we can't connect from the internet" are not sufficient for support to begin troubleshooting. To avoid delays, be explicit and provide all relevant details. Screenshots can be helpful to visually demonstrate the issue, and for web-based interfaces, a .HAR file or HTTP archive can be provided. GCP documentation offers instructions on how to obtain a .HAR file from major browsers.

When troubleshooting networking issues, it is recommended to include TCP dump output if available. Additionally, attaching log snippets and example stack traces that are relevant to the issue can be helpful.

When filing a support case, the priority field is used for initial routing, especially for issues that may require immediate attention. The case creation form itself provides information on case priorities. For production emergencies, select P1 as the priority. Consider the impact of the issue on your business when determining the priority. It can be beneficial to include a sentence describing the impact in your own words. For example, even if no end users are directly affected, a problem with the development version may be considered a P1 if it blocks a critical security fix. Providing explicit explanations for the selected priority helps avoid incorrect assumptions.

Support cases have built-in response timers that aim to set reasonable expectations. If you have specific time constraints or deadlines, it is important to communicate them to the support team. For example, if you need a response by 5:00 PM because that is when your shift ends, inform the team accordingly. If you find that you are affected by an issue that has already been reported on the status.cloud.google.com, you can track the progress through the dashboard or use the "Me Too" link in the cloud support portal to receive automatic updates for known issues.

Customers with 24/7 support can request that their case follows the sun, meaning it will be reassigned multiple times per day to ensure an active support engineer is always available. Lower priority cases, however, will not be managed around the clock, as the assigned engineer will go off shift at some point. This can cause delays if the assigned engineer is not in your time zone. To address this, you can ask for the case to be managed in your time zone, such as "please manage this case in my time zone, EST." This can facilitate easier communication with the support engineer during a lengthy discussion, but it may not be as effective if the development team is located in a different time zone.

If you need to continue the conversation via Google Hangouts or phone after filing a case, provide a link or phone number for the support team to reach you. Support will attempt to call when it does not interfere with issue resolution. For video conferences, Google Hangouts is the preferred option, but other solutions that work within a Chrome browser without requiring extensions can be used. When requesting a callback, provide two or three available time slots to initiate the scheduling process.

Thank you for reading. If you have any questions, please leave them in the comments section below, and we will address them in future materials.

EITC/CL/GCP GOOGLE CLOUD PLATFORM DIDACTIC MATERIALS
LESSON: GCP SUPPORT
TOPIC: HOW TO USE THE CLOUD SUPPORT API FEATURE IN GOOGLE CLOUD PREMIUM SUPPORT

In today's fast-paced world, time is of the essence, especially when it comes to resolving issues. Switching between different systems to keep track of problems can be a hassle. This is where the Cloud Support API feature in Google Cloud Premium Support comes into play. With this powerful tool, Premium Support customers gain access to Google's Cloud Support API, which seamlessly integrates Google support tracking systems with your own.

The Cloud Support API allows you to automatically sync case information between your internal ticketing system and Google's case management system. This means that you can manage and track support cases as they progress, all in one place. No more bouncing between multiple systems or wasting time searching for information. By integrating the Cloud Support API, you can rely on a single source of information, streamlining your workflows and saving valuable time.

One of the key benefits of using the Cloud Support API is enhanced visibility. By allowing your data to cascade across platforms, you have a comprehensive view of your support cases and their status. This eliminates the need for manual updates and ensures that you are always up to date and informed. With the Cloud Support API, you can maintain a system of record that pulls data directly from Google systems, fully integrated and hassle-free.

To access the Cloud Support API, simply navigate to the Cloud Console and select APIs and Services from the menu. Enable APIs and Services, then search for Cloud Support in the API library. Once you find the Google Cloud Support API, enable it, and you'll be ready to go in no time. It's that easy!

The Cloud Support API feature in Google Cloud Premium Support is a valuable tool for managing and tracking support cases efficiently. By integrating this API, you can save time, keep track of issues in a single system, and set yourself up for success with Premium Support.