1. what is gcloud?

Google Cloud (also known as Google Cloud Platform or GCP) is a provider of computing resources for developing, deploying, and operating applications on the Web. Although its [cloud infrastructure](https://www.zdnet.com/article/what-is-cloud-computing-everything-you-need-to-know-about-the-cloud/) does serve as the host for applications such as [Google Workplace](https://workspace.google.com/) (formerly G Suite, and before that Google Apps), GCP is mainly a service for building and maintaining original applications, which may then be published via the Web from its hyperscale data centre facilities.

2. How to use the gcloud shell?

**Step 1:** Click the **Activate Cloud Shell** button at the top of the google cloud console window to start a cloud shell.

**Step 2:** Your Google Cloud Shell session is now ready to use.  
**Step 3:** When the Google Cloud Shell is started, the active project in Console is propagated to your gcloud configuration inside Cloud Shell.  
**Step 4:** When you start the Google Cloud Shell, a small Google Compute Engine virtual machine running a Debian-based Linux operating system is already provisioned for you. Instance persists while the Cloud Shell session is active and terminates after an hour of inactivity.  
**Step 5:** You can use “gcloud version” command to display a list of installed Cloud SDK components and their versions:

1. What is Helm? How to configure it, what is the purpose of using it?

Ans. In simple terms, Helm is a package manager for Kubernetes. Helm is the K8s equivalent of yum or apt. Helm deploys charts, which you can think of as a packaged application. It is a collection of all your versioned, pre-configured application resources which can be deployed as one unit.

* helm, the package manager for Kubernetes, is a useful command line tool for: installing, upgrading and managing applications on a Kubernetes cluster. Helm packages are called charts. We will be installing and managing JupyterHub on our Kubernetes cluster using a Helm chart.
* Charts are abstractions describing how to install packages onto a Kubernetes cluster. When a chart is deployed, it works as a templating engine to populate multiple yaml files for package dependencies with the required variables, and then runs kubectl apply to apply the configuration to the resource and install the package.
* Purpose: Because it’s a huge shift in the way the server-side applications are defined, stored and managed. Adoption of Helm might well be the key to mass adoption of micro services, as using this package manager simplifies their management greatly.

1. What is a service mesh? How can istio be configured on the GCP?

Ans.

* In software architecture, a service mesh is a dedicated infrastructure layer for facilitating service-to-service communications between services or micro services, using a proxy.
* To configure istio : Go to the Kubernetes clusters page in the Cloud Console and select the cluster you want to update
* Select Edit.
* Under Istio (beta) select Enabled to display Istio mTLS (beta).
* Select the mTLS security mode you want to use for your cluster from the drop-down.
* Click Save to update your cluster.

1. What is DNS, VPC?

Ans.

* DNS: The Domain Name System is a hierarchical and decentralized naming system for computers, services, or other resources connected to the Internet or a private network. It associates various information with domain names assigned to each of the participating entities.
* VPC: A virtual private cloud (VPC) is a secure, isolated private cloud hosted within a public cloud. VPC customers can run code, store data, host websites, and do anything else they could do in an ordinary private cloud, but the private cloud is hosted remotely by a public cloud provider.

1. What does cloud storage do?

Ans.

* Cloud storage allows you to save data and files in an off-site location that you access either through the public internet or a dedicated private network connection.
* Data that you transfer off-site for storage becomes the responsibility of a third-party cloud provider.

1. What is API gateway, how to configure the APIGEE?

Ans.

* An API gateway is programming that sits in front of an API (Application Programming Interface) and is the single-entry point for defined back-end APIs and micro services (which can be both internal and external).
* To build and deploy your first API proxy:

Create an Apigee account. You have to do this before you can go any further. Or is it farther?

Create an API proxy using the Edge management UI. You'll connect your proxy to a mock endpoint so you can see how it works.

Test your new proxy to make sure you're on track.

Change your target endpoint so your policy has more interesting data to play with.

Add a policy to convert the response from XML to JSON. Policies are at the heart of your proxy's request-response flow.

1. What is PUB/SUB? Purpose of it.

Ans.

* In software architecture, publish–subscribe is a messaging pattern where senders of messages, called publishers, do not program the messages to be sent directly to specific receivers, called subscribers, but instead categorize published messages into classes without knowledge of which subscribers, if any, there may be.
* Similarly, subscribers express interest in one or more classes and only receive messages that are of interest, without knowledge of which publishers, if any, there are.
* Publish–subscribe is a sibling of the message queue paradigm, and is typically one part of a larger message-oriented middleware system. Most messaging systems support both the pub/sub and message queue models in their API; e.g., Java Message Service (JMS).
* This pattern provides greater network scalability and a more dynamic network topology, with a resulting decreased flexibility to modify the publisher and the structure of the published data.
* Purpose: Pub/Sub enables you to create systems of event producers and consumers, called publishers and subscribers. Publishers communicate with subscribers asynchronously by broadcasting events, rather than by synchronous remote procedure calls (RPCs).

1. What is a topic?

Ans. A named resource to which messages are sent by publishers. In a pub/sub model, any message published to a topic is immediately received by all of the subscribers to the topic. Pub/sub messaging can be used to enable event-driven architectures, or to decouple applications in order to increase performance, reliability and scalability