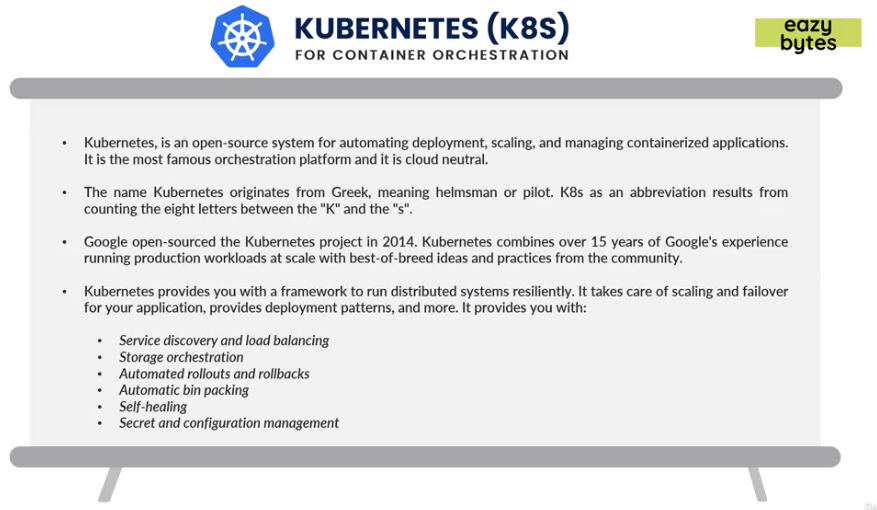
**Kubernetes**

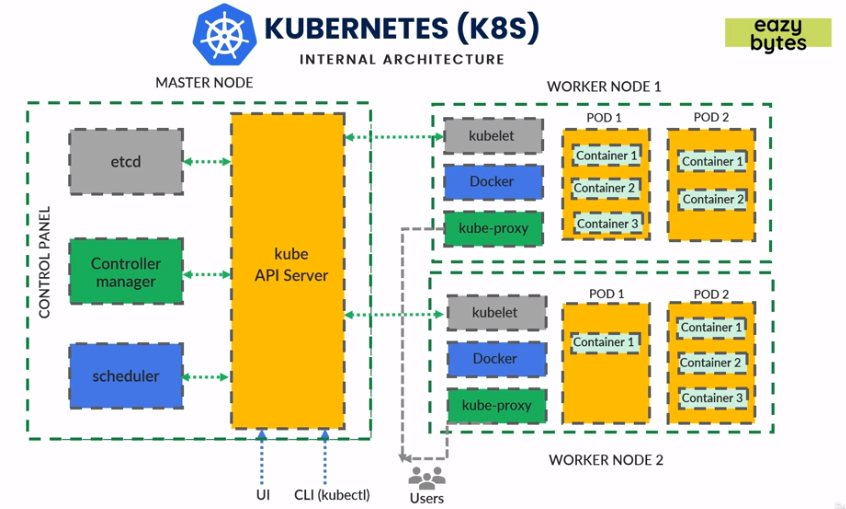
****

*Container Orchestration* solves challenges with micro services like automatic deployments, rollouts and rollbacks, self-healing of containers in case of any failures in containers, scaling up/down services based on required in case of tons of micro services.

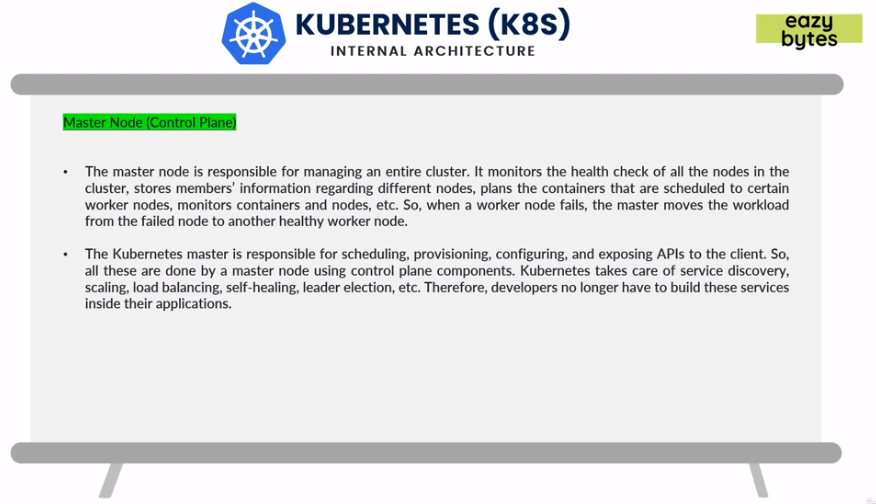
* **Kubernetes:** *Orchestration* means managing containerized applications and taking care of automatic deployment, scaling and managing irrespective of Geographical location. *Kubernetes* is an open-source system and is open sourced by *Google* in *2014*. It is the most famous *orchestration* platform and it is cloud neutral. *Kubernetes (K8s)* has following advantages and provides features like:
  + Service Discovery and Load balancing.
  + Storage orchestration i.e., managing storage systems.
  + Automated rollouts and rollbacks of micro services.
  + Self-healing of failing containers.
  + Configurations and Secret management.

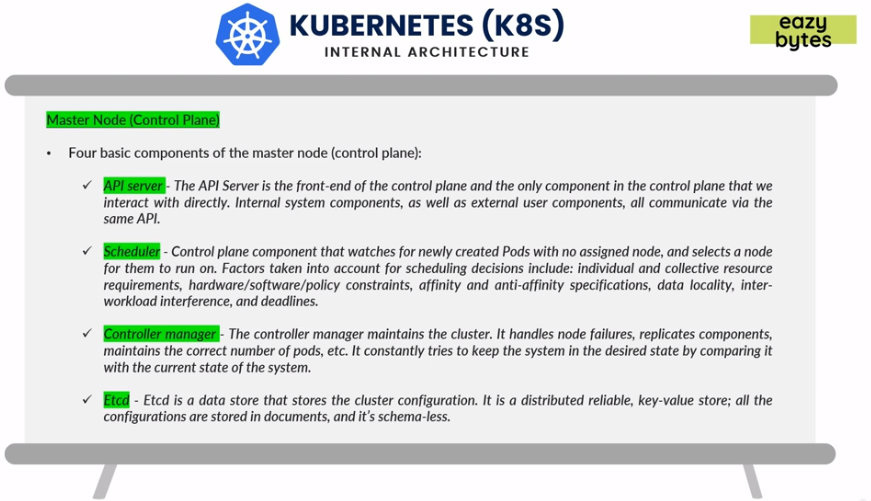


* **Internal Architecture of Kubernetes:**

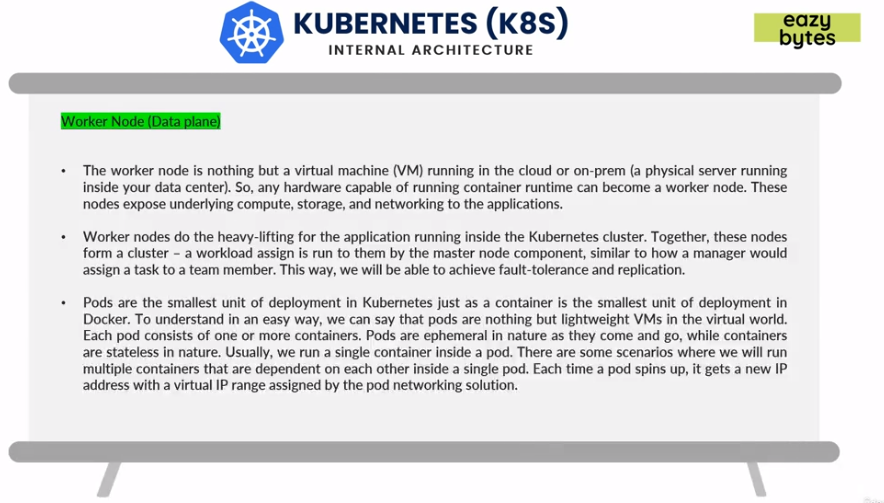


A *cluster* is a collection of servers. *Worker Node* consists of containers, where all micro services will be deployed and code will be executing. All these *Worker Nodes* are managed by *Master Node* like how many instances, information of all containers and managing each worker nodes. Using UI and *KubeCtl (Kube Controller)* CLI we can connect with *Kube API server* which is like a gateway to connect with cluster. We can only connect with the *Master node* and pass on commands (like command to deploy new container, scale etc), rest is all managed via *Master Node* itself. *Kube API server* passes all the user command to *Scheduler. Scheduler* is a component in *K8s* which will make calculations like to deploy in which *worker node* and is responsible to work with *worker node*. *Scheduler* will schedule a deployment, scaling etc based on the command given in *worker node.* *Controller Manager* to always keep checking *worker node* health stats like looking for any failures and also try to maintain the number of replicas desired given during deployment instruction. *Controller Manager* will compare current state and desired state to make sure number of instances or pods are maintained. *Etcd* is like the brain of *k8s* cluster or *master node* which will store all the information of all pods, worker nodes and their information. *Etcd* is like database for all cluster configuration of pods in forms of key and value pair. *Scheduler* updates all information in *Etcd* and interact with it to get status of *worker nodes.*

**

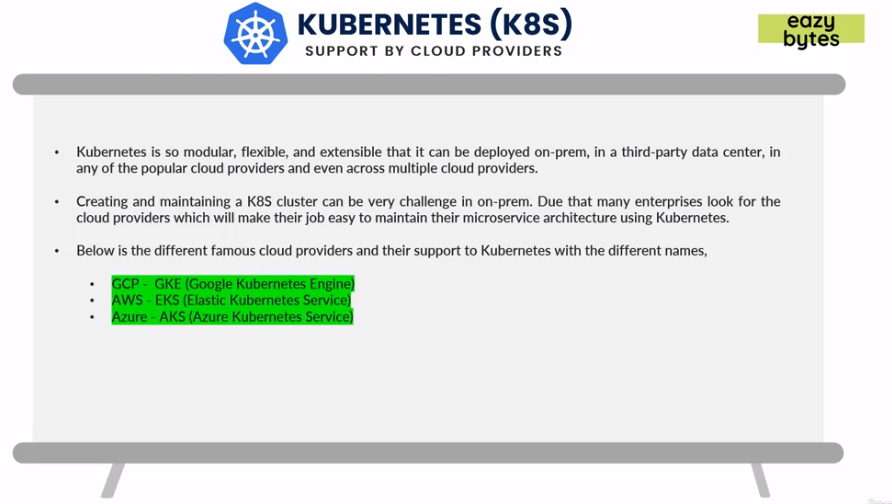
**

*Worker Nodes* are VMs running in cloud and on-prem. *Kubelet* is a component running in each *K8s* and using *Kubelet* only *Scheduler* can interact with *worker nodes* once it identifies which *worker node* to interact with. *Kube-proxy* is endpoints exposed via *worker nodes* so that the end users can interact. *Pods* are smallest deployable units in *worker nodes,* which will have containers inside it and it provides memories to these containers.

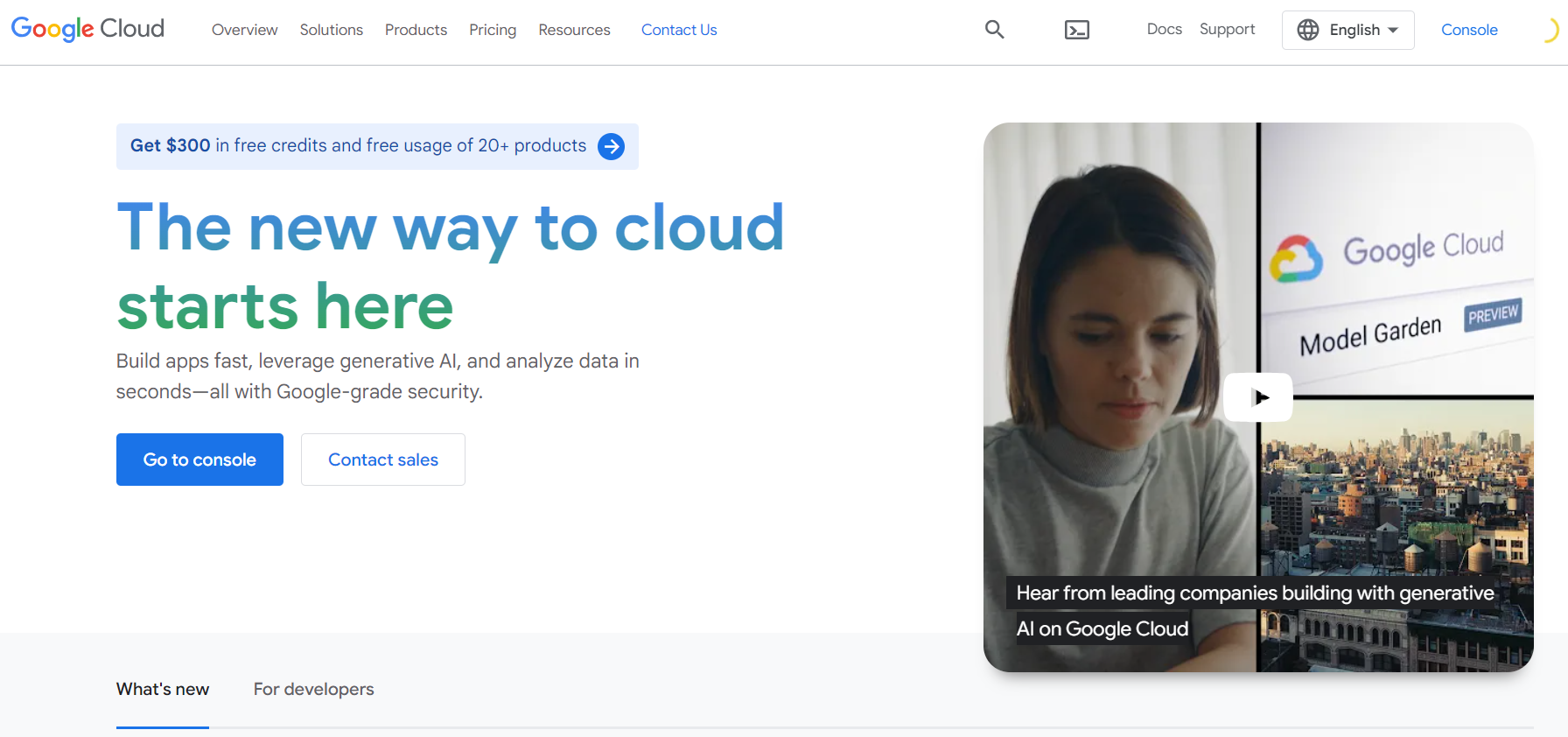


*K8s* is flexing enough and can be deployed in on-prem servers, any cloud platform using different cloud providers and support or in any data centers. Creating and maintaining *K8s* cluster on on-prem servers can be little difficult hence most preferable way is to go for cloud providers. Three main cloud providers are:

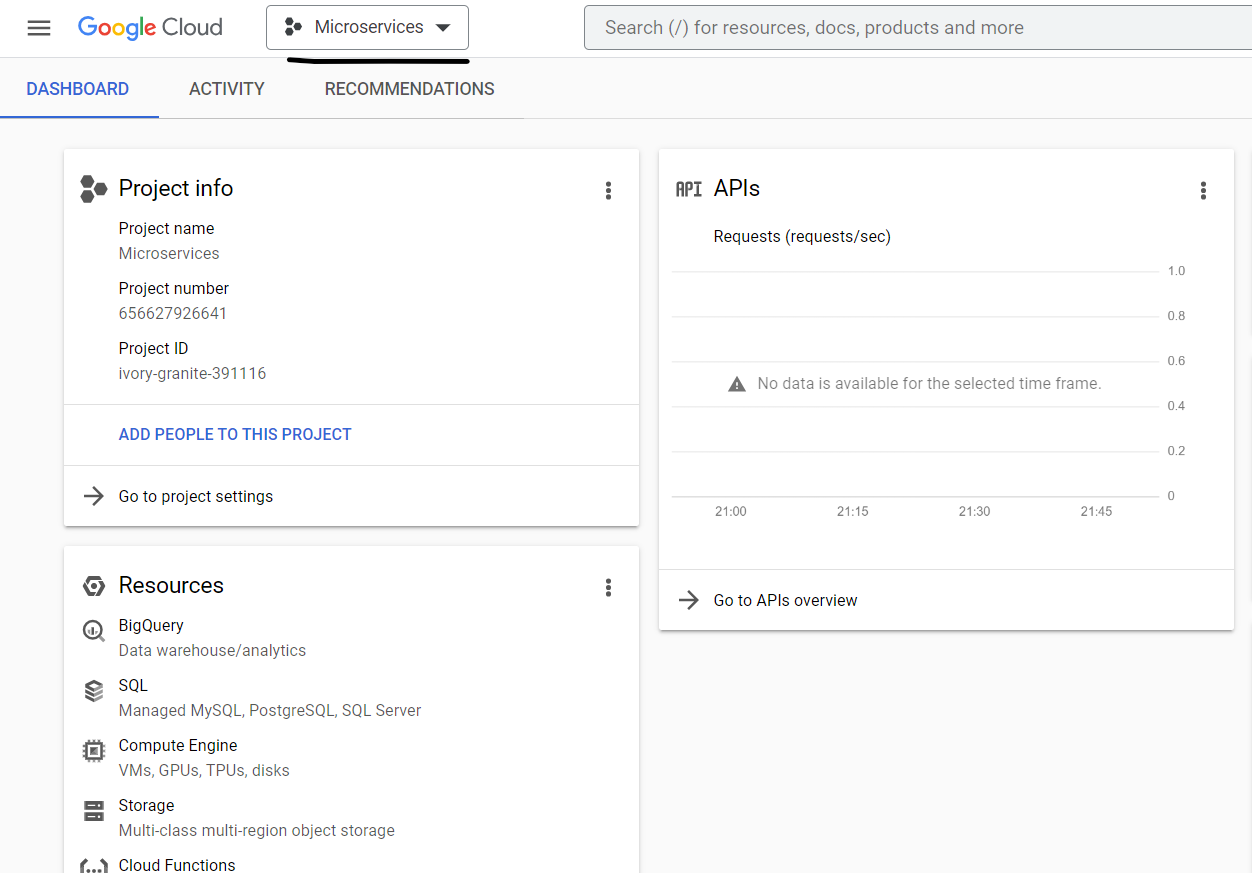
1. GCP - GKE (Google Kubernetes Engine)
2. AWS – EKS (Elastic Kubernetes Service)
3. Azure – AKS (Azure Kubernetes Service)



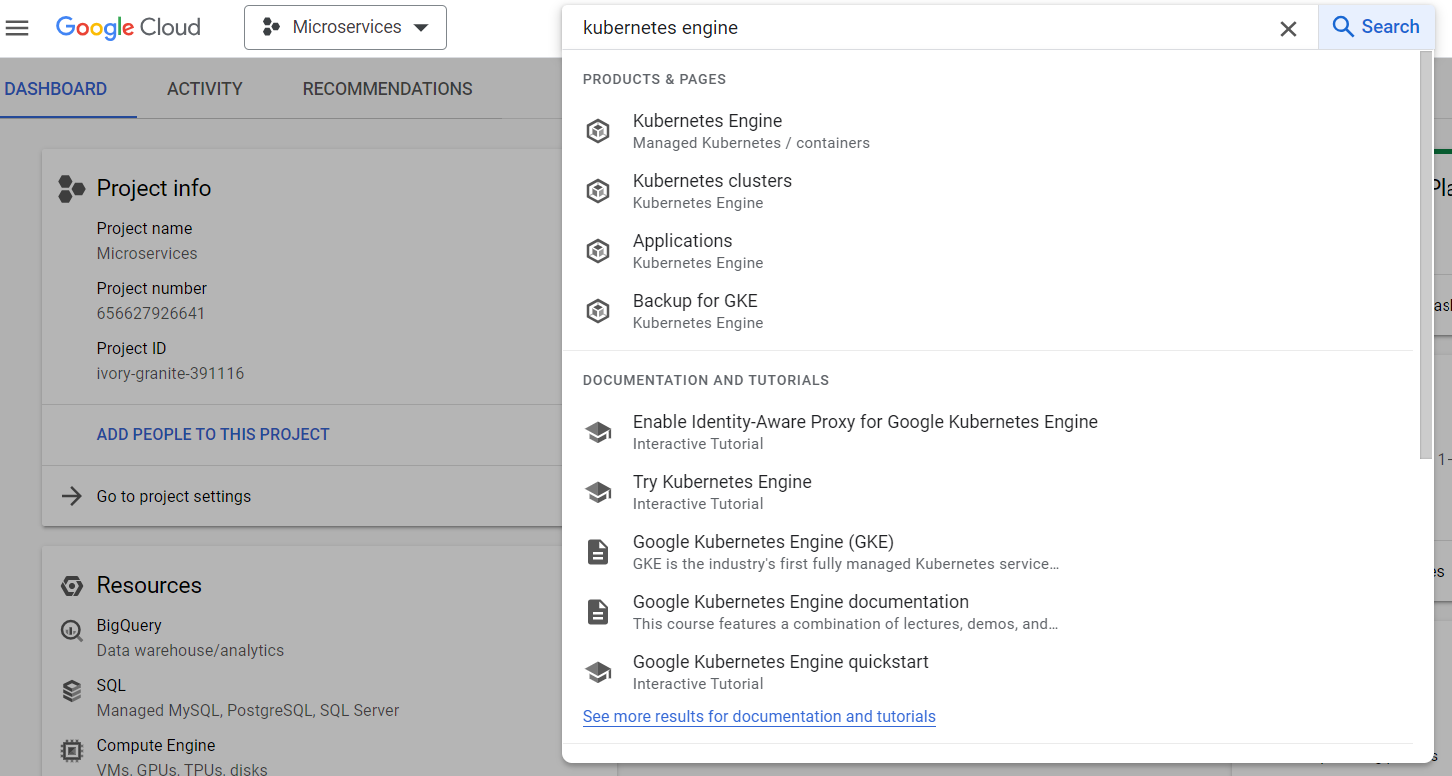
* **Creating an account in GCP and setting up Google SDK:** 
  + Go to [*http://cloud.google.com*](http://cloud.google.com)and sign in with Google credentials.
  + Once sign-in done, click on Go to Console or Console on top right corner.

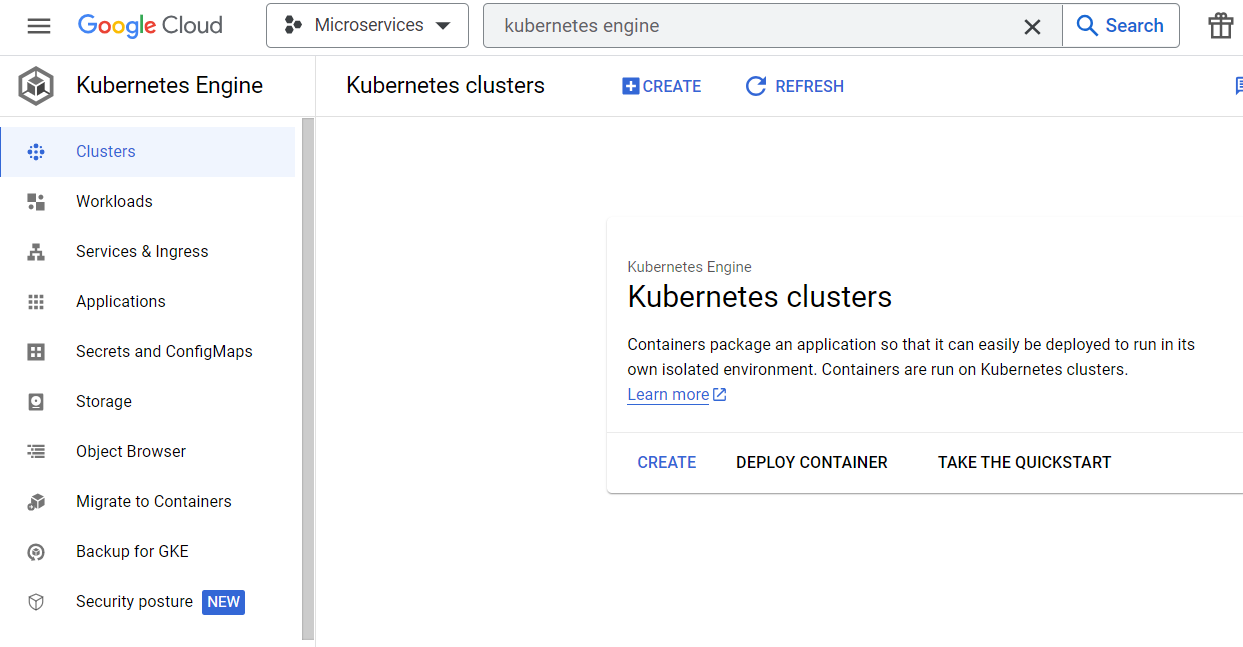


* + Create a new project and go to Dashboard.

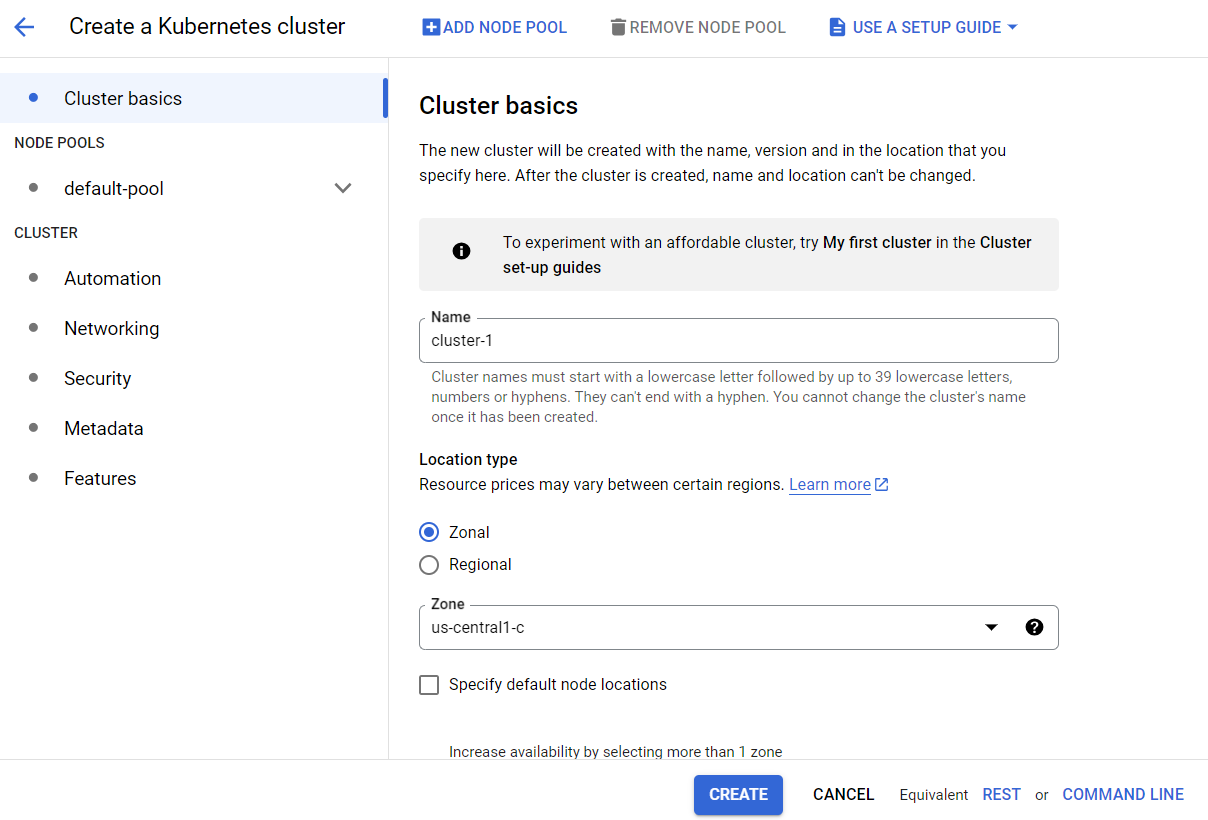


* + Search in Dashboard *Kubernetes Engine* and create a basic standard Cluster.

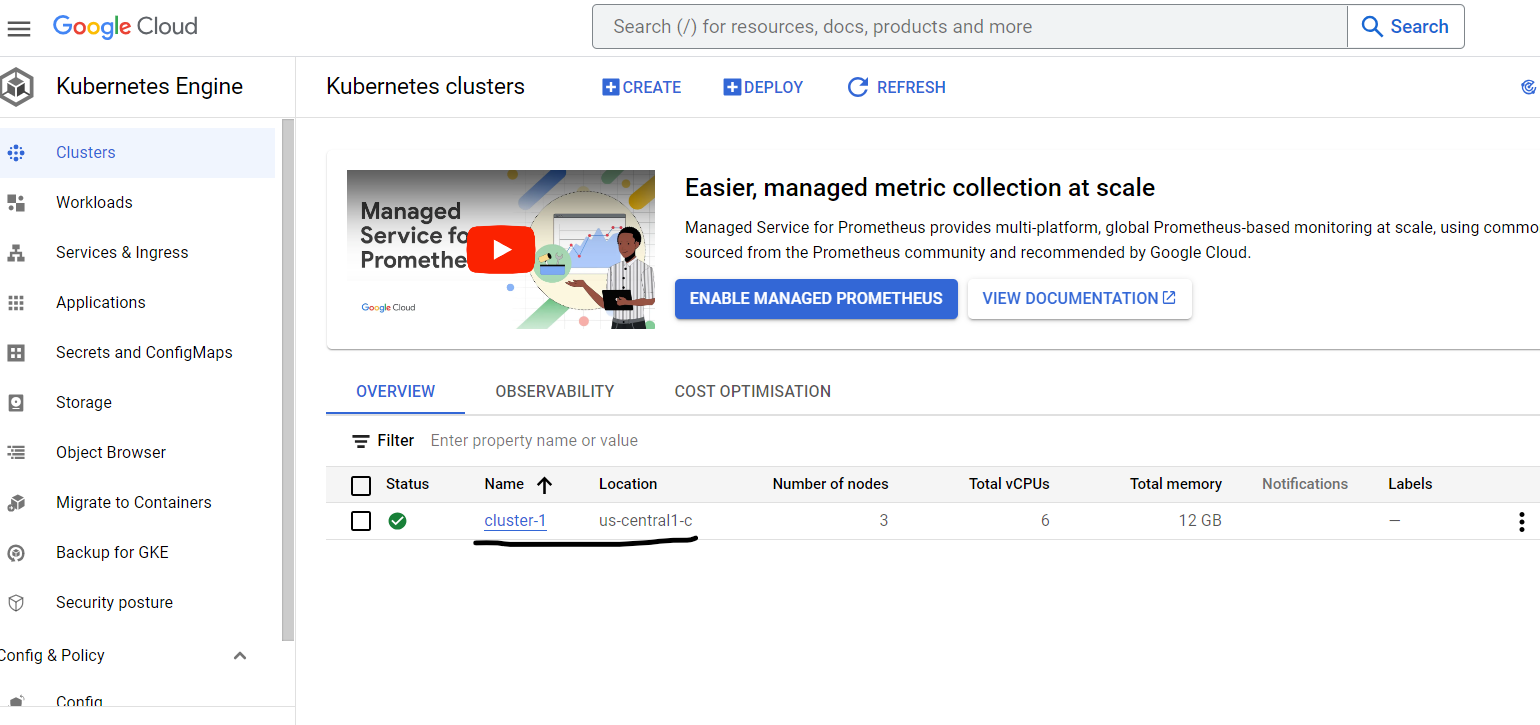




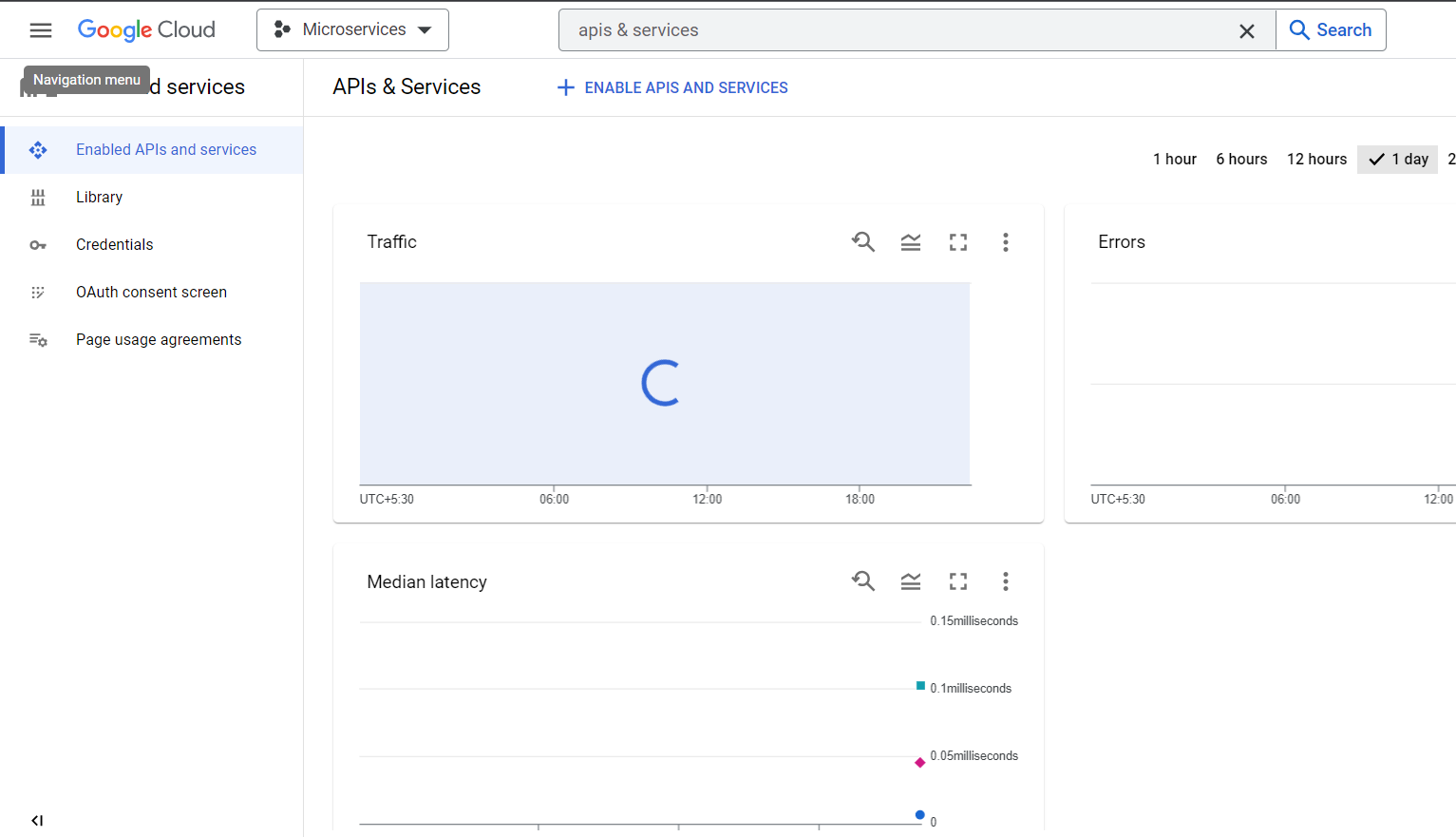
* + Select Standard Cluster and create a cluster.

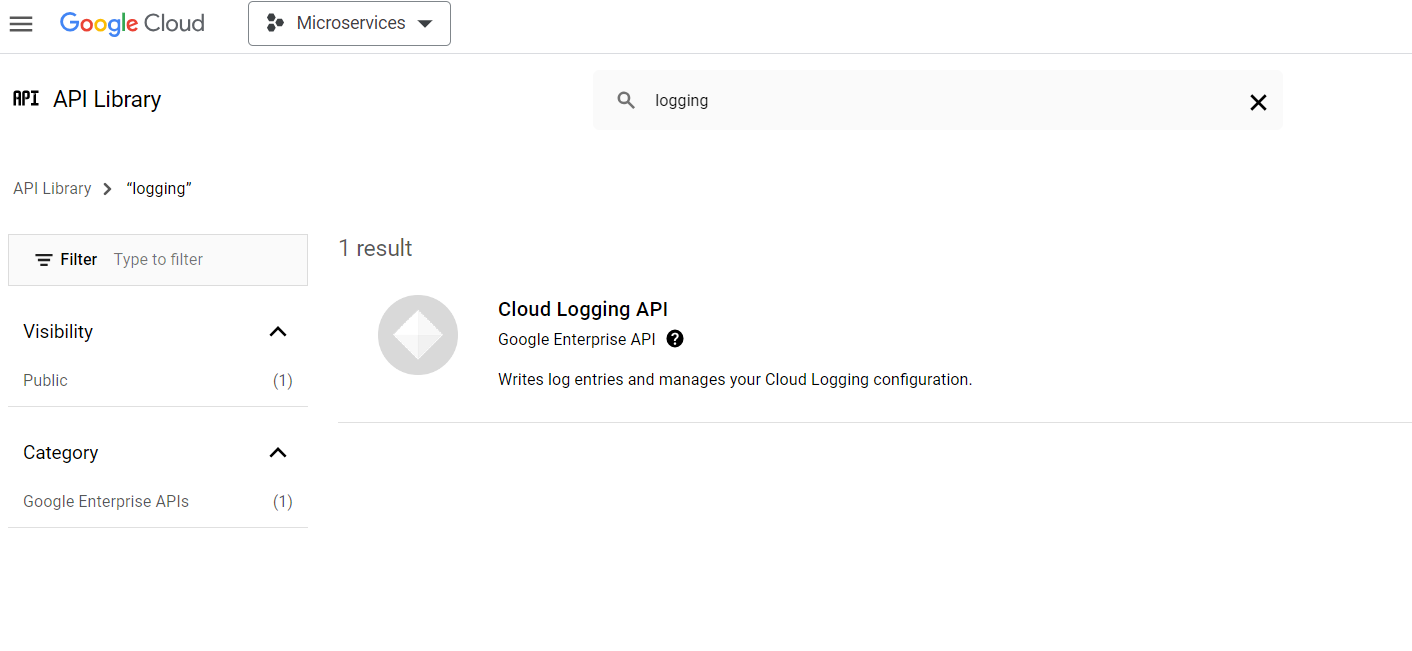


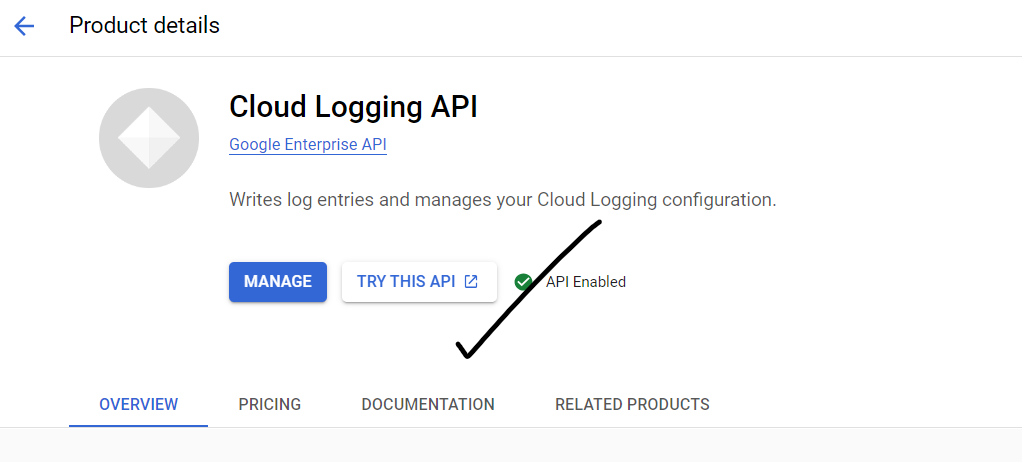
* + Once a cluster is created (will take 5-10 mins to create), we can see a green tick in cluster.



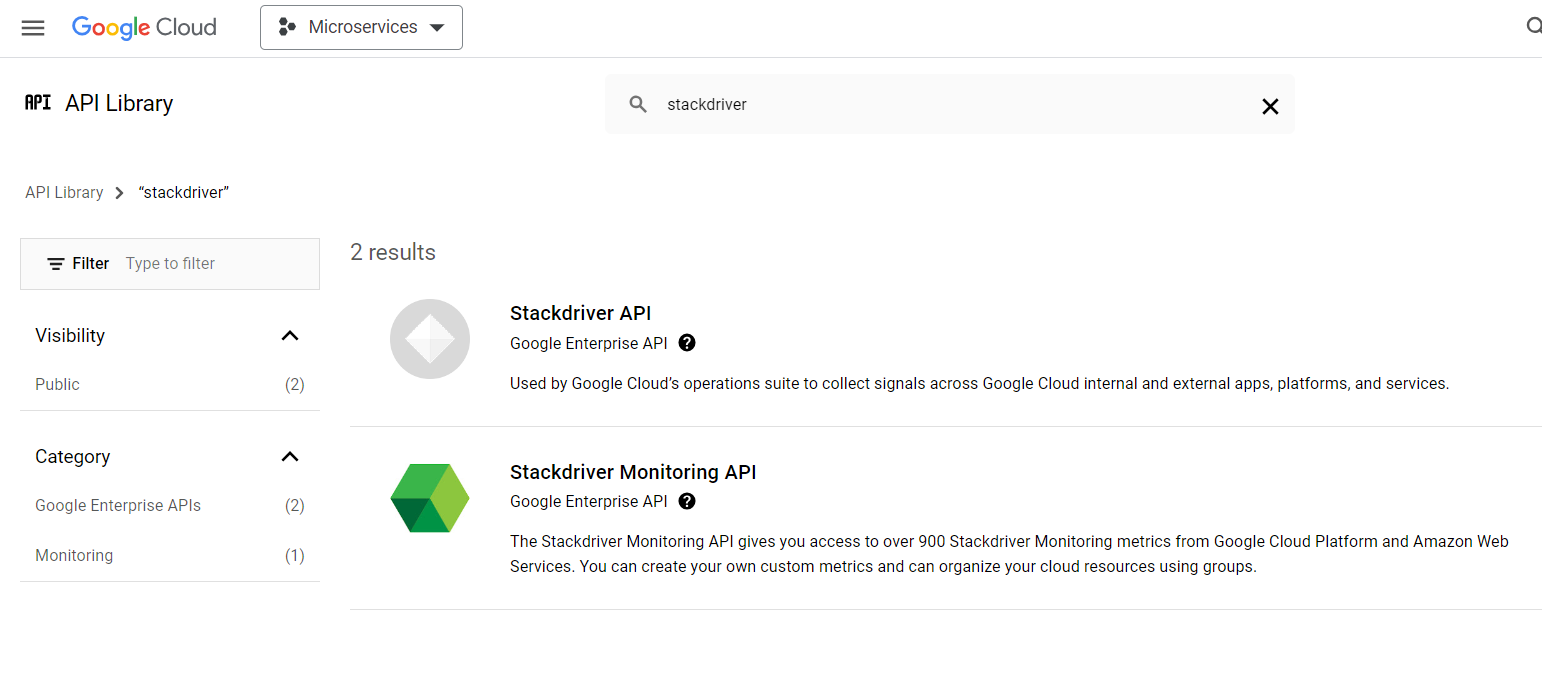
* + We need to enable logging API so for that in Search type *APIs and Services*, Click on Enable APIS and Services and in that search for Cloud Logging API.



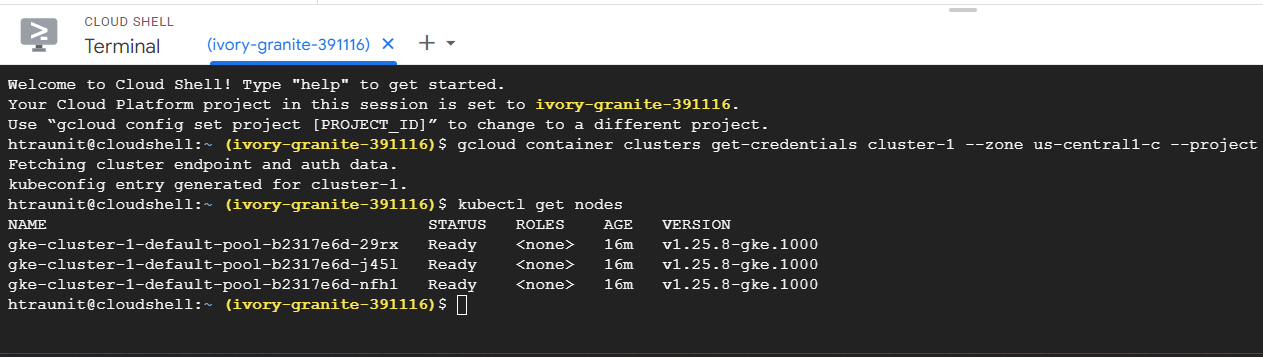




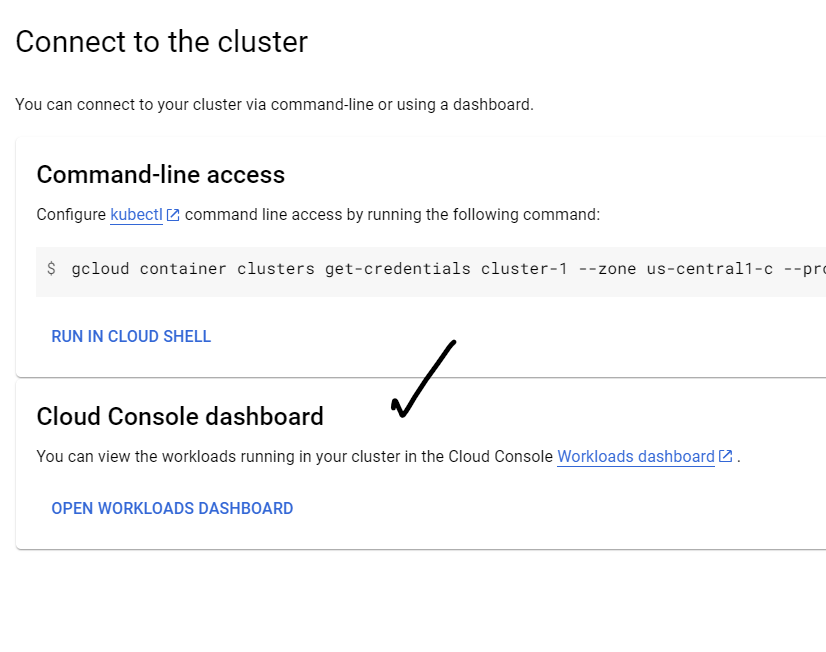
* + For enabling Monitoring we can use *StackDriver* API. Search for *StackDriver* and Enable them.



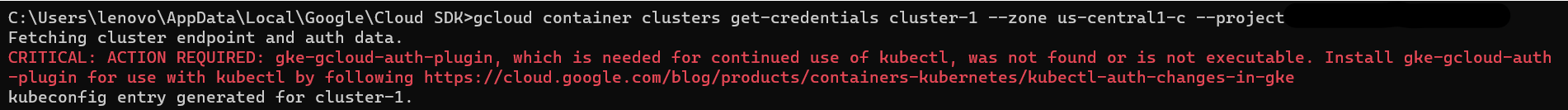
* + *Google SDK* is a CLI used to interact with *GCP* or using Google Shell cloud [*https://cloud.google.com/sdk/docs/install*](https://cloud.google.com/sdk/docs/install)but it’s better to use SDK.



* + Once installed we can connect to cluster using SDK. Go to Cluster in dashboard, right click and get the connect string.



* + If you get error like below use following commands in SDK. Once done issues should be resolved.



* + - gcloud components install gke-gcloud-auth-plugin
  + Some important commands: