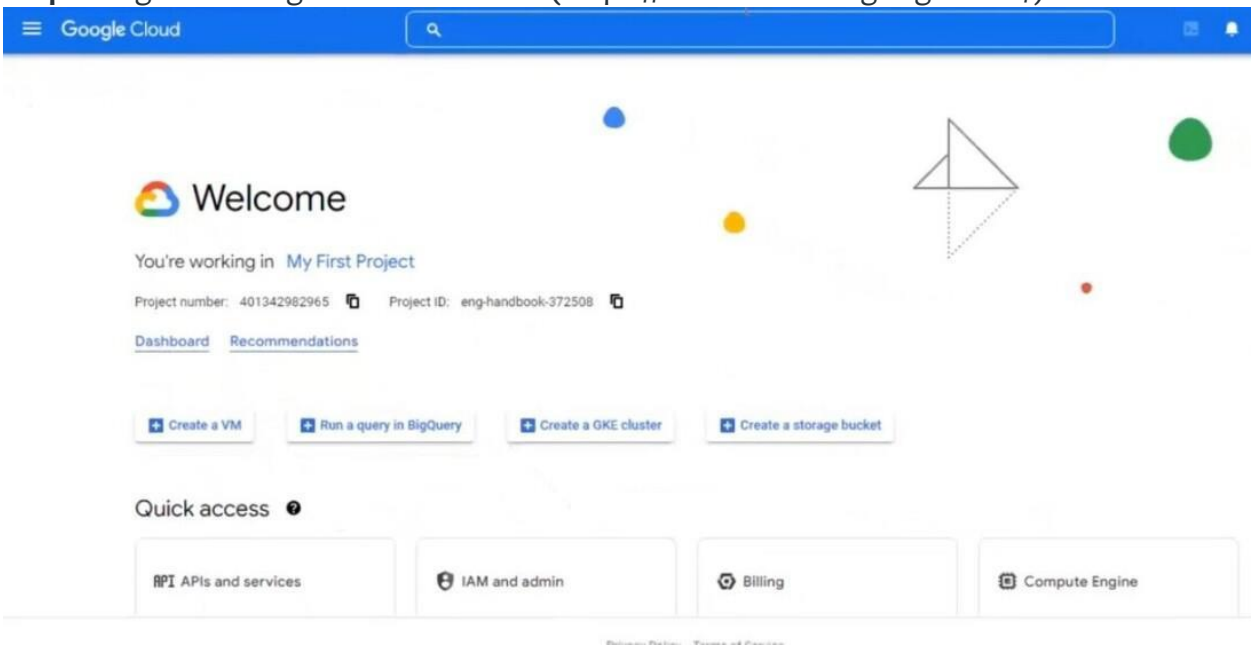
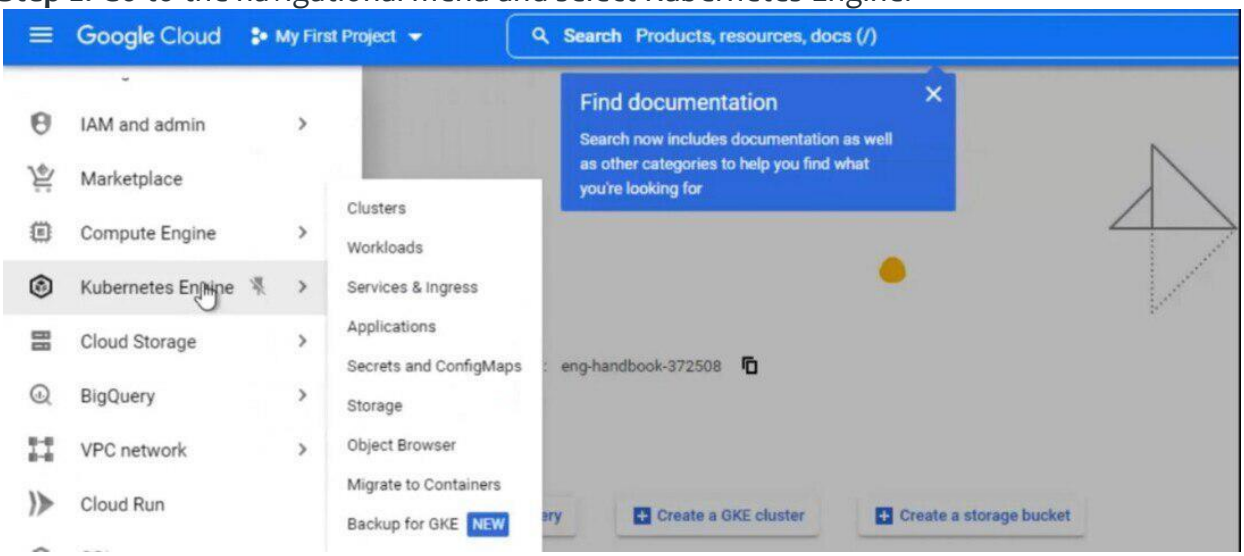
Step 1: Log in to Google Cloud Platform (https://console.cloud.google.com/)



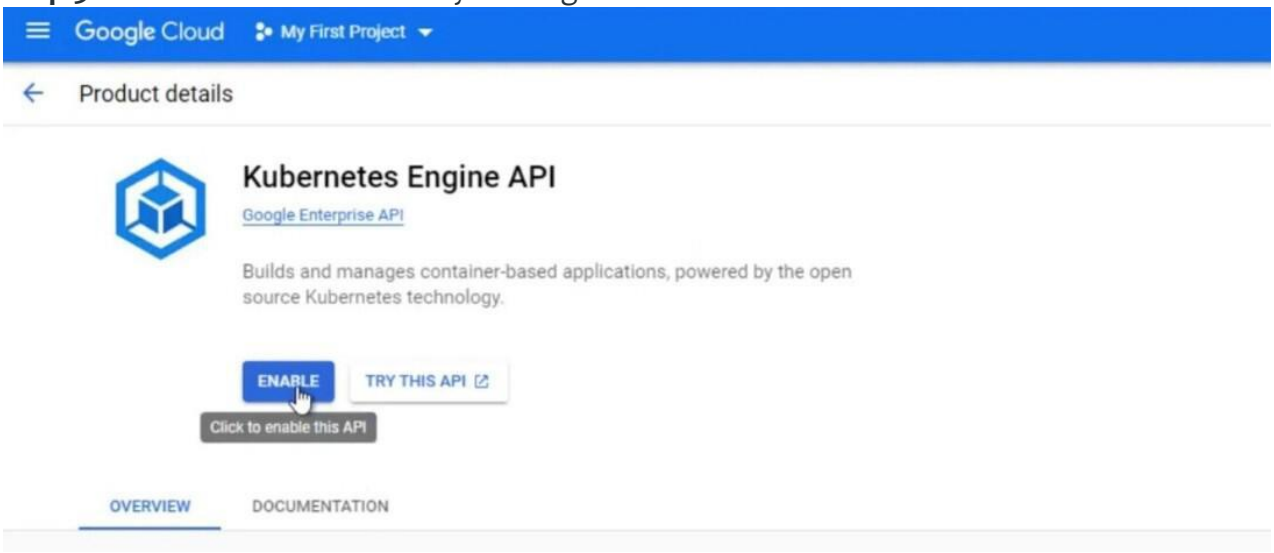
Log in to Google Cloud Platform

Step 2: Go to the navigational menu and select Kubernetes Engine.



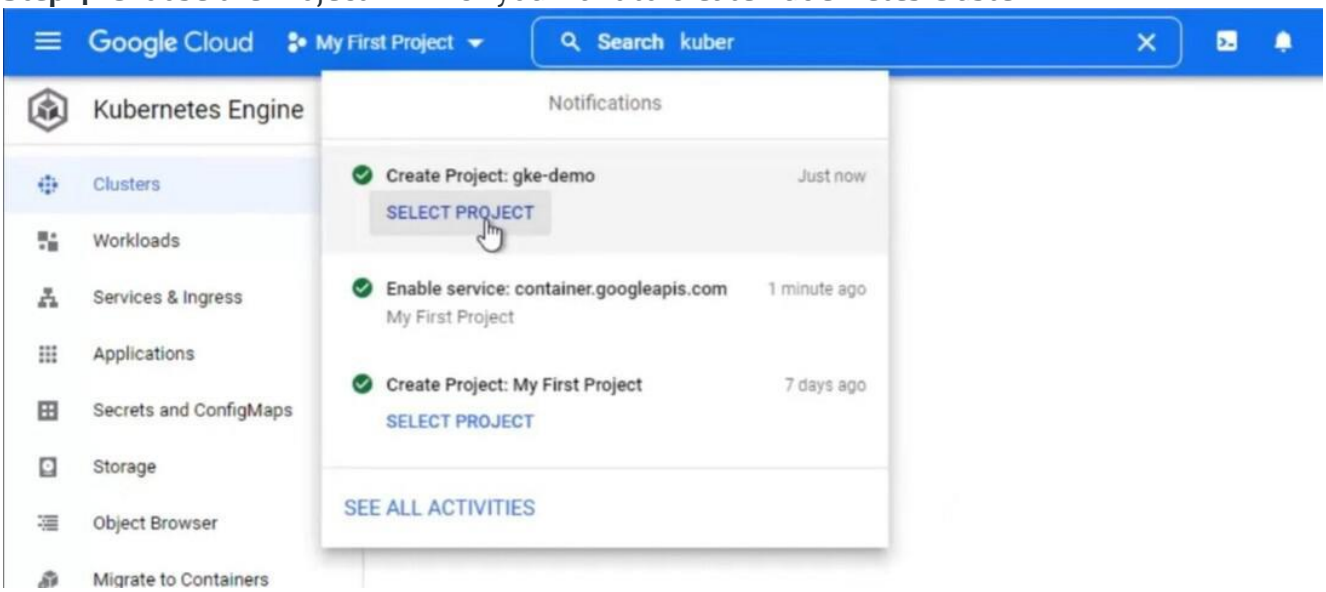
Select Kubernetes Engine

Step 3 : Enable Kubernetes API by clicking on Enable button.



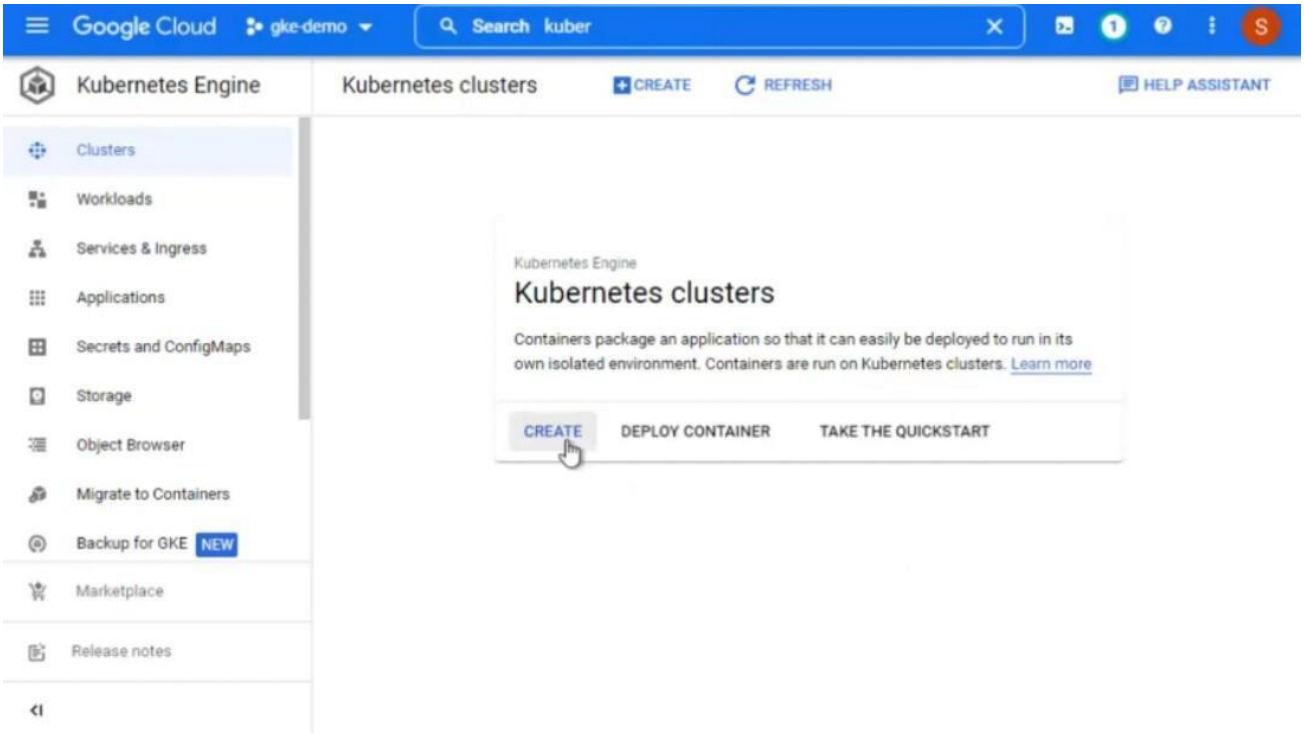
Enable API

Step 4: Choose the Project in which you want to create Kubernetes Cluster.



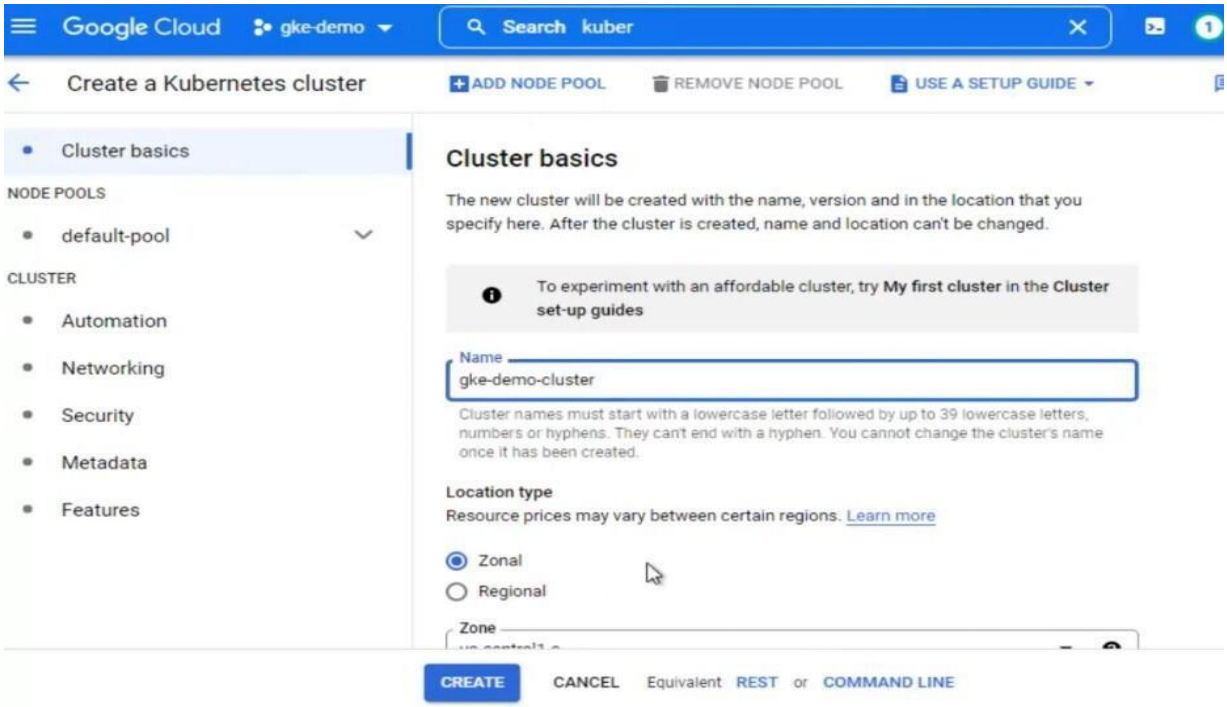
Select Project

Step 5: Now, after enabling API and choosing project its time to Create Kubernetes Cluster, Click on Create button.

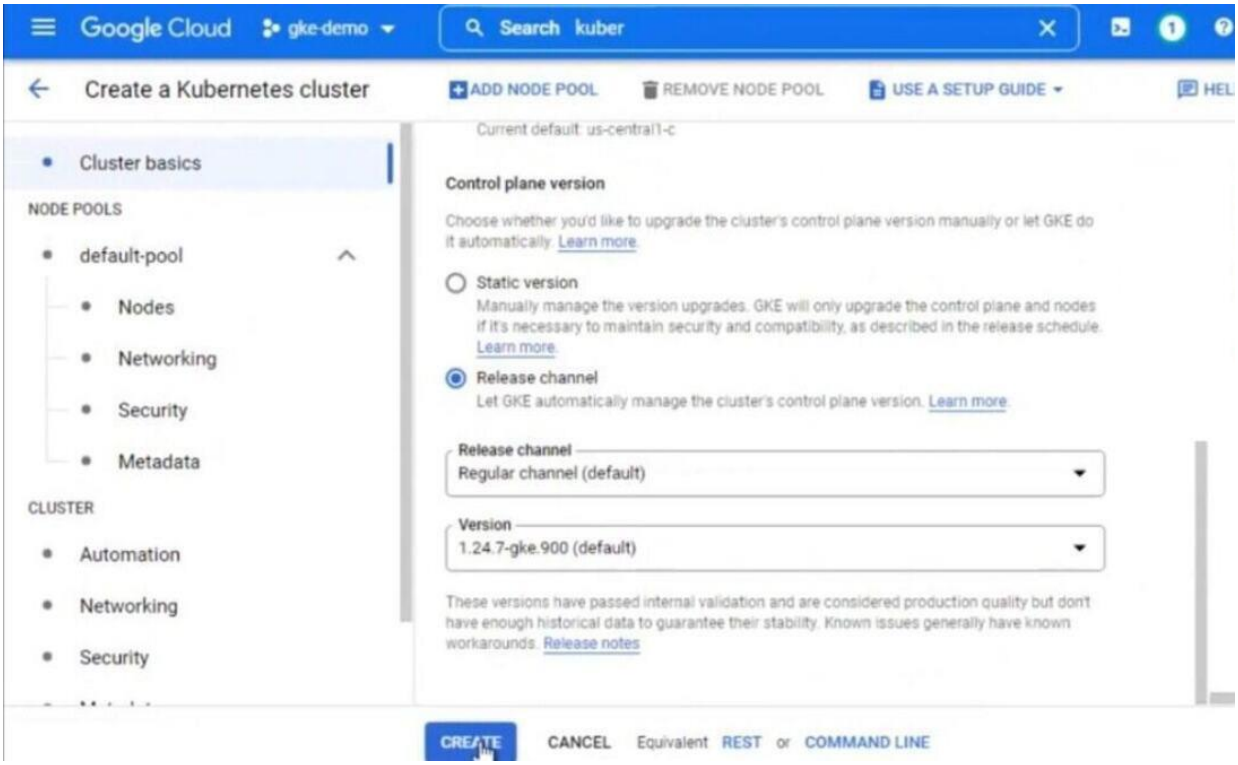


Click on Create Kubernetes Cluster

Step 6: Give all the basic details like Cluster Name, Location, Zone etc and then click on the create button.



Give all Basic Info



Click on Create Button

Kubernetes is Successfully deploy in Google Cloud Platform (GCP).

Troubleshooting Common Issues

- **Networking Issues:** Check network rules and firewall regulations. Ensure right communication between pods and services within the cluster.
- **Resource Constraints:** Monitor useful resource utilization (CPU, reminiscence) the usage of tools like Stackdriver. Adjust aid requests and boundaries in your container to prevent useful resource exhaustion.
- **Pod Scheduling Failures:** Investigate node resource availability. Nodes would possibly lack sufficient resources for the requested pods. Consider scaling your cluster or adjusting resource requests.
- **Image Pull Failures:** Verify image availability and authentication. Ensure correct image paths and proper access permissions.
- **Cluster Autoscaler Issues:** Check cluster autoscaler logs and configuration. Ensure the cluster autoscaler is efficiently configured to add/remove nodes primarily based on useful resource needs.

Best Practices for Deploying Kubernetes on GCP

- **Use Managed Services:** Leverage controlled services like GKE to simplify cluster management and consciousness on software development.
- **Resource Optimization:** Optimize resource requests and limits for container. Monitor aid usage and regulate configuration.
- **Implement CI/CD Pipelines**: Implement strong CI/CD pipelines to automate the deployment method, ensuring speedy and dependable application updates.