Day - 14

12 June 2025 18:46

GKE → Google Kubernetes engine.

① gcloud. → GUI Part

② CLI → Google command line Interface

③ Terraform → IaC.

Cluster → Standard. (you manage the Node)

→ Autopilot (fully managed by google)

GKE

↳ Kubernete master (control Plane)

① Node health, Upgrades, Scaling & Security Patches

② Integration with GCP service like Networking, logging, Monitoring & IAM.

Cluster → Collection of Masters & Worker node where your app runs.

Node → A VM (collection) where containers are deployed.

Pod → Smallest deployable unit in Kubernetes

Control Plane → Manage the cluster - schedule, Monitor, scaling etc.

Node Pool → A group of Identical Nodes (VMs)

Why GKE? → Kubernetes Control Plane.

① Fully Managed.

② Cloud Native.

IAM, cloud Monitoring

③ Autoscaling.

↓

Scale node as per workload.

④ Security. → Regular patches, IAM Integration & n/w Policies.

⑤ Devops friendly. → CI/CD Integration with cloud Build Artifact Registry

⑥ Logging & Monitoring.

↓

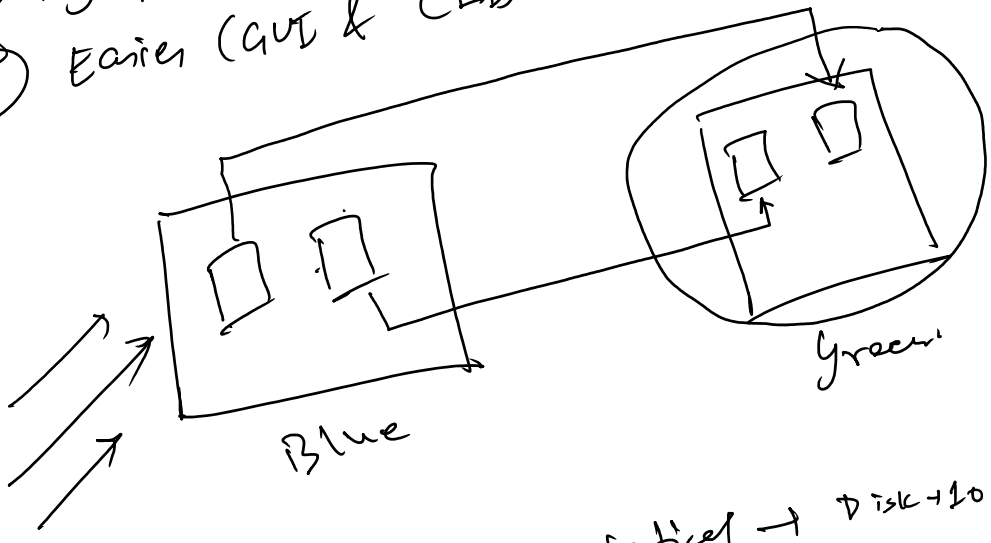
Native stackdriver Integration with logs/Metrics.

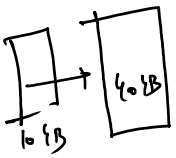
Use Cases :-

1. Hosting Microservice App. like Online store / chat apps
2. Running CI/CD pipelines in container
3. Machine learning model serving.
4. High-availability & Global scale production workloads.

GKE vs Self Managed Kubernetes :-

1. Handled by Google — You manage everything.
2. Built in Auto scaling — Manual setup.
3. upgraded / Patches Automated by google — Manual
4. Pay for Nodes + Minimal Control fees — Infra Ops.
5. Easier (GUI & CLI) — Complex Setup.

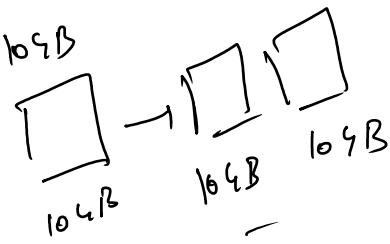




Auto Scaling → Vertical
→ Horizontal.

AutoScaling → Horizontal.

↳ Disk = 10GB



Kubectl get nodes

↳ Basic Commands Concise overview.

Kubectl get nodes -o wide.

↳ Additional Info. Container runtime -.

- Internal IP
- External IP
- OS-
- Kernel version