Day 8

Topics

1. Docker [Contd.]
2. Source Control
3. Google Kubernetes
    a. Standard
    b. Autopilot
4. Cloud Build
5. Cloud Deploy
6. Detailed GCP CI/CD Demo

Source Control

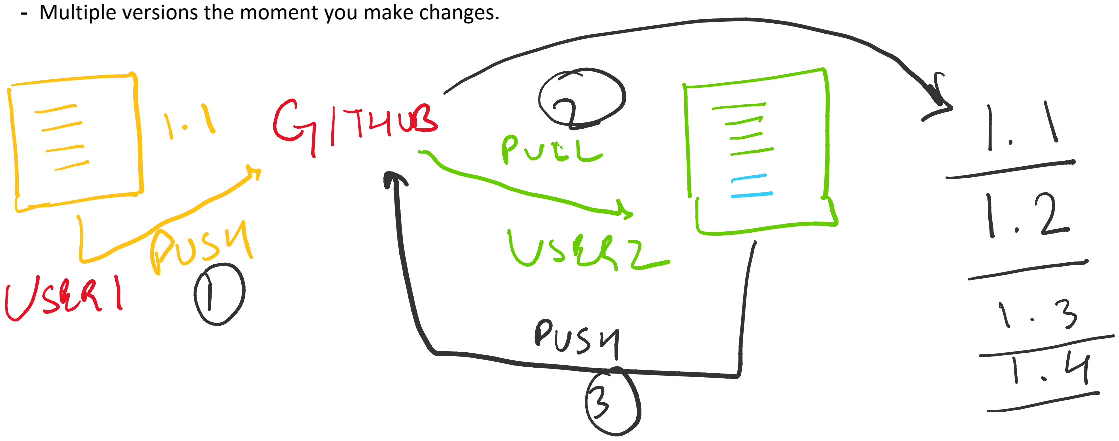
- Source Control helps you to track and manage changes to your code.
- You/your organization will have access to a repo and you can push your code to that repo and keep track of it.

1. Backup
2. Tracking changes
3. Collaboration
4. Maintain a single source of truth [version history]

Eg - GitHub
BitBucket
GitLab
Stash
P4,SVN

Git

- Version control system, commonly used to track changes to computer files.
- Multiple versions the moment you make changes.



Local Repository

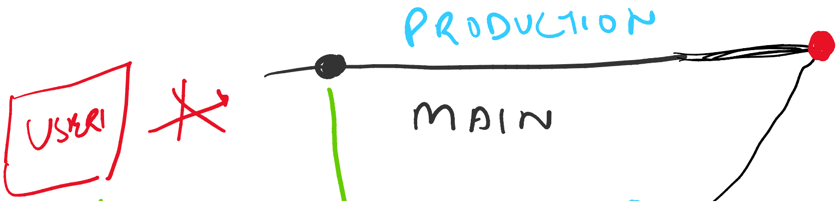
- Present on your laptop/computer.
- Used to make changes locally/offline.

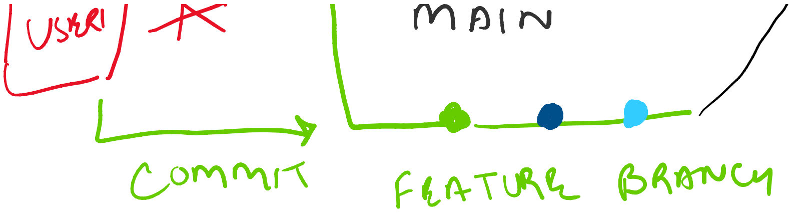
Remote Repository

- Server repository that can be present anywhere.
- Github.
- Which will be used by all your team members to track changes.

Branching

-

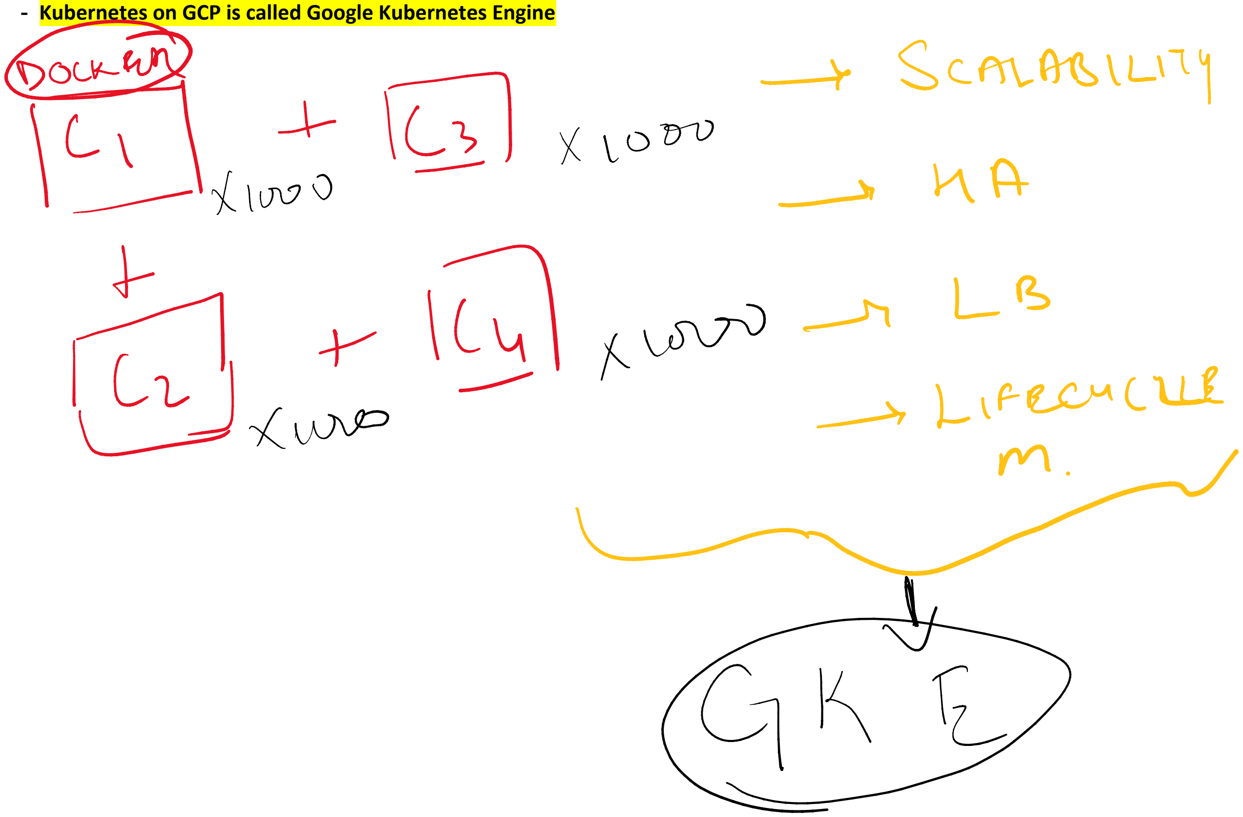




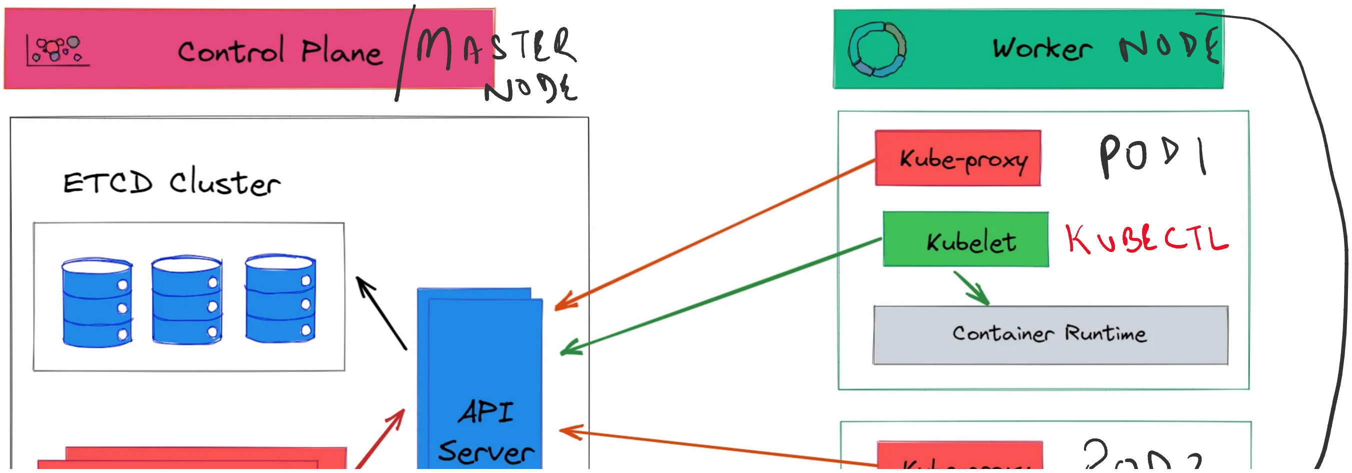
Google Kubernetes Engine

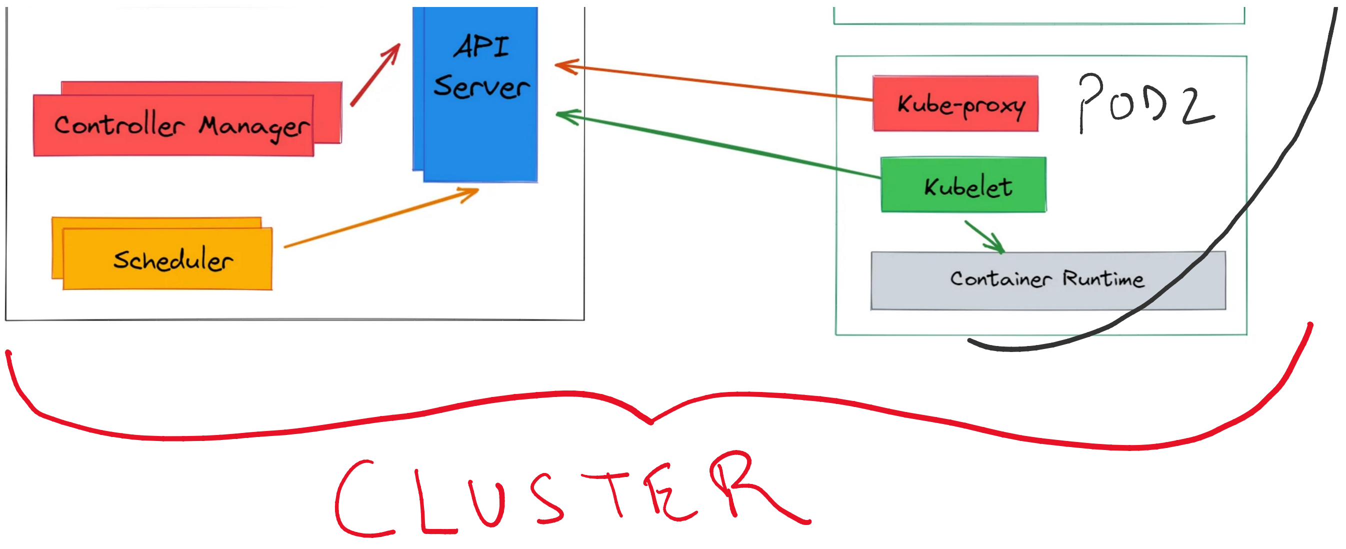
- Docker is the tool that helps you to create Containers.
- Orchestrate the containers you need Kubernetes.
- Scale/redeploy/Start
- From configuring, deploying, managing and monitoring . Kubernetes can help you manage container's lifecycle.
- High availability
- Load balancing

- **Kubernetes on GCP is called Google Kubernetes Engine**



KUBERNETES ARCHITECTURE

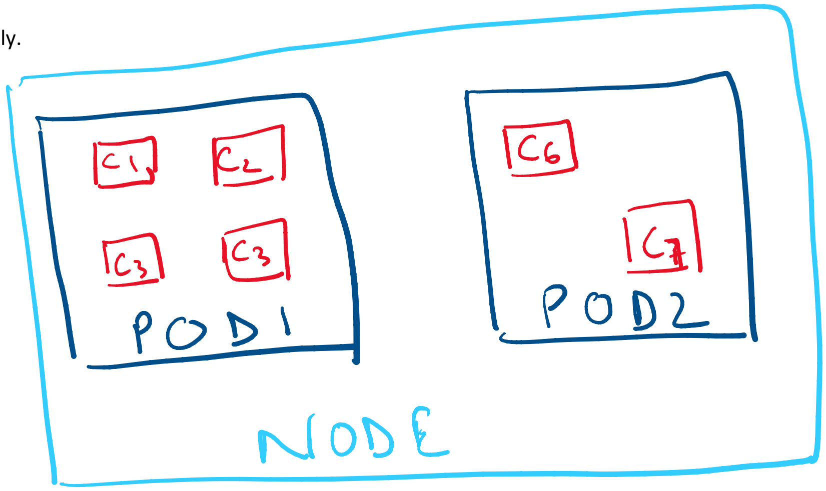




- K8 supports **declarative configuration**
    - Desired state = 8 containers
- 2 Important elements inside a file [YAML]
    - **Object Spec [Desired State]**
    - **Object Status [Current State]**

Control Plane

1. **Kube-APIserver**
    a. This is the only single component that you will interact with directly.
    b. Accept commands that view or change th state of the cluster.
    c. Launch Pods.
2. **Etcd cluster**
    a. The cluster's Database**.**
    b. Cluster's configuration data.
3. **Kube-scheduler**
    a. Schedule Pods onto the Nodes
4. **Controller manager**
    a. Monitoring unit of K8.
    b. Manages state of the cluster.



Worker Node

1. **Kube-proxy**
    a. Networking component of the Node.
2. **Kubelet/Kubectl**
    a. Communicate states of the Nodes to the API server
3. Container runtime on GKE by default is containerd.

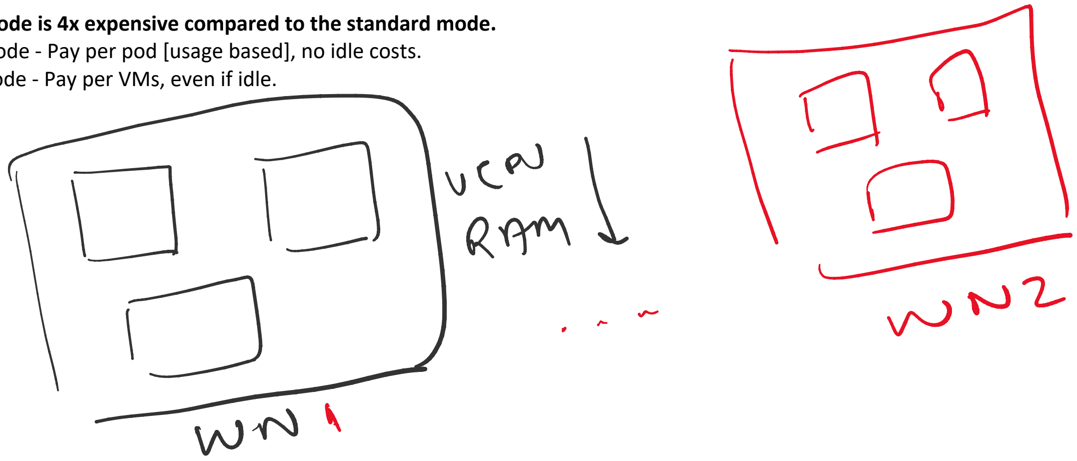
GKE Autopilot vs GKE Standard

1. **Autopilot mode** - Google will manage the underlying infrastructure.
    a. Node configuration
    b. Autoscaling
    c. Upgrades
    d. Security
    e. Networking

1. **Standard Mode**
    a. You want to manage the underlying infrastructure
        a. Node configuration
        b. Autoscaling
        c. Upgrades

d. Security
e. Networking
   You cannot control the master node.

Autopilot mode is 4x expensive compared to the standard mode.
Autopilot Mode - Pay per pod [usage based], no idle costs.
Standard Mode - Pay per VMs, even if idle.



Cloud Build

Cloud Build is the tool - that will help you out to build the latest version of your code and push it to the repository.
Automate the build process

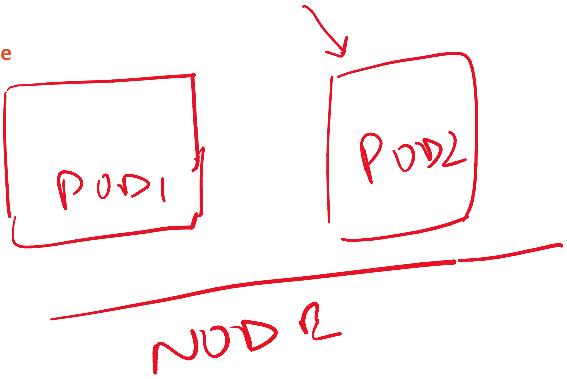
- Building
- Testing

Speed in the build process.
CI - Continuous Integration

Cloud Deploy

Cloud Deploy **Continuous Delivery** service designed on GCP.
This will help us to automate the deployment of your code on GKE

Max Surge



Demo GCP CI/CD

Part 1 - Create the app

1. Created a emarket application in local machine on visual studio.
2. Tested the application on local host.

Part 2 - Containerize the app locally

3. We created the docker file for the application code.
4. Containerized the application

Part 3 - Push your Code to GitHub

5. Created a new repository on GitHub.
6. Add the remote repo.
7. Pushed my code to the main branch.

Part 4 - Create infrastructure of GKE