Kubernetes Research (developed by Google managed by CNCF)

geeksforgeeks.org/introduction-to-kubernetes-k8s/

Kubernetes is open-source Container Management tool that automates container deployment, container scaling, descaling, and container load balancing.

Benefits of using Kubernetes:

Automated deployment and management

Scalability-offers horizontal pod scaling – pods automatically scaled depending on load

High availability- reducing latency issues

Cost-effectiveness

Improved developer productivity

Deploying and Managing Containerized Applications with Kubernetes

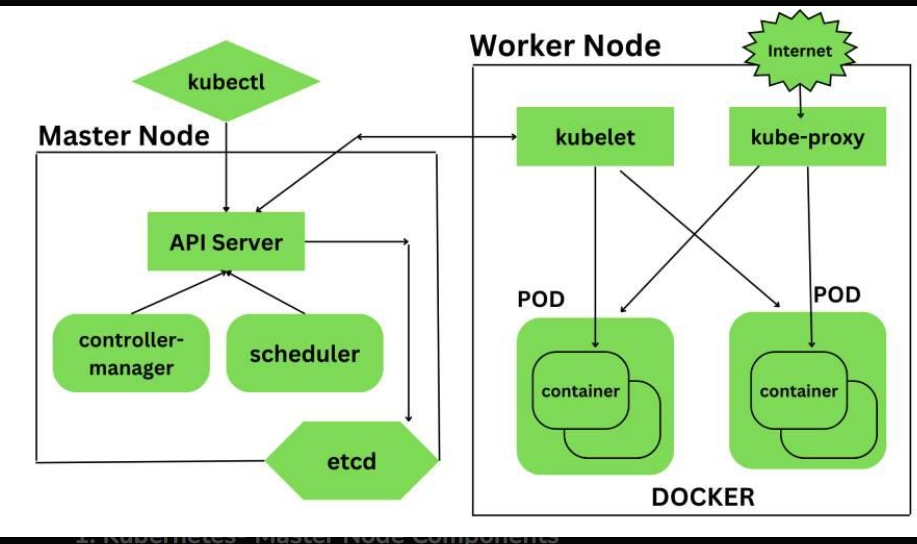
Install Kubernetes ad setup Kubernetes cluster – one master and 2 worker nodes in any of the clouds

Know create deployment manifestfile – specify exact number of pods required and the container image and what types of resources are required after completion of writing the manifestfile apply the file using kubectl command.

After creating pods- expose the service to the outside of the – write one more manifestfile which contains service type (LoadBalancer or ClusterIP), ports,and selectors.

Networking – use different types of plugins to increase flexibility.

Kubernetes follows the client-server architecture – master installed on one machine and the node on separate Linux machines. Follows master-slave model—master manages Docker containers across multiple Kubernetes nodes. A master and its controlled nodes(worker nodes) make up a “Kubernetes cluster”.



Kubernetes master is responsible for managing the entire cluster, coordinates all activities inside the cluster, and communicates with worker nodes to keep the Kubernetes and application running.

When Kubernetes is installed, we have 4 primary components of Kubernetes Master that are also installed. Components of the Kubernetes Master: API Server-entry point for all the REST commands used to control cluster (API tool kubectl), Scheduler-service that distributes workload and allocates the pod to the new node, Controller Manager- known as controllers—it is a daemon that runs in a non-terminating loop and is responsible for collecting and sending information to the API server.

Kubernetes Worker Node Components: Kubelet-primary node agent that communicates with master node and executes on each worker node inside the cluster, Kube-Proxy-core networking component inside the Kubernetes cluster-responsible for maintaining entire network configuration, Pods- a group of containers that are deployed together on the same host, Docker- the containerization platform used to package your application and all its dependencies together in the form of containers to make sure your application works in any environment.

Kubernetes is used with hybrid and multi-cloud deployments as well as edge computing –other applications listed these are the ones that stood out for me relating to our project

5 BASIC Steps of Building a Private Cloud:

1 Standardization – software and hardware components used for deploying private cloud

2. Consolidation- trimming down equipment to what you need and decide what is able to be scaled up in the future

3. Virtualization- depends on intended delivery model and design of private cloud (Hyper-V, Xen, or VMware)

4. Automation – modify IT processes to your liking

5. Orchestration – final stage – “self-service” ability to private cloud architecture- two main components – a self-service portal and a service catalog

Kubernetes in the cloud and on bare metal differ

<https://kubernetes.io/blog/2024/04/05/diy-create-your-own-cloud-with-kubernetes-part-1/>

<https://kubernetes.io/blog/2024/04/05/diy-create-your-own-cloud-with-kubernetes-part-2/>

<https://kubernetes.io/blog/2024/04/05/diy-create-your-own-cloud-with-kubernetes-part-3/>

<https://deploy.equinix.com/blog/guide-to-running-kubernetes-on-bare-metal/>

[Link-](https://kubernetes.io/docs/tasks/tools/install-kubectl-linux/) to install and set up kubernetes.