Components of Kubernetes:•

Master Node Components:

1. API Server 2. Controller Manager 3. ETCD 4. Scheduler

1) API Server:It is the front-end for the Kubernetes control plane.

2) Controller Manager:This is a component on the master that runs controllers.

•Node Controller:Responsible for noticing and responding when nodes go down.

•Replication Controller: Responsible for maintaining the correct number of pods for every replicationcontroller object in the system.

•Endpoints Controller: Populates the Endpoints object (that is, it joins Services and Pods).

•Servervice Account and Token Controllers:Create default accounts and API access tokens for new namespaces.

3)ETCD: Consistent and highly-available key value store used as Kubernetes' backing store for all cluster data.

4)kube-scheduler -Control plane component that watches for newly created Pods with no assigned node, andselects a node for them to run on

Node ComponentsNode components run on every node, maintaining running pods and providing the Kubernetesruntime environment

1)Kubelet- An agent that runs on each node in the cluster. It makes sure that containers arerunning in a Pod.The kubelet takes a set of PodSpecs that are provided through various mechanisms and ensuresthat the containers described in those PodSpecs are running and healthy. The kubelet doesn'tmanage containers which were not created by Kubernetes.

2)kube-proxy- kube-proxy is a network proxy that runs on each node in your cluster,implementing part of the Kubernetes Service concept. Kube-proxy maintains network rules onnodes. These network rules allow network communication to your Pods from network sessionsinside or outside of your cluster.

3)Container runtime- The container runtime is the software that is responsible for runningcontainers.

Manifest File Components:apiversionKindMetadataSpecFile name : nginx-deployment.ya

Service components:Kubernetes ServiceTypes allow you to specify what kind of Serviceyou want. The default is ClusterIP.Type values and their behaviors are:

ClusterIP: Exposes the Service on a cluster-internal IP. Choosing thisvalue makes the Service only reachable from within the cluster. Thisis the default ServiceType.(To talk to other nodes in the cluster)

NodePort: Exposes the Service on each Node's IP at a static port(the NodePort). A ClusterIP Service, to which the NodePort Serviceroutes, is automatically created. You'll be able to contactthe NodePort Service, from outside the cluster, byrequesting <NodeIP>:<NodePort>. (The entrpoint for nod

•LoadBalancer: Exposes the Serviceexternally using a cloud provider'sloadbalancer. NodePort and ClusterIP Services, to which the external loadbalancer routes, are automaticallycreated.

•Externalname: Maps the Serviceto the contents ofthe externalName field(e.g. foo.bar.example.com), byreturning a CNAME record with itsvalue. No proxying of any kind isset up.

Kubernetes Commands

1)Kubectl create deploy my-nginx –image=nginx –dry-run=client –oyaml > mynginx.yaml

2)Kubectl get pods

3)Kubectl get namespace

4)Kubectl create ns mydev

5)Kubectl create –f pod.yaml

6)Kubectl describe svc nginx

7)Kubectl exec -it pod1 bash