**Kubeadm**

sudo su

sudo apt update -y

sudo apt install docker.io -y

sudo systemctl start docker

sudo systemctl enable docker

sudo curl -fsSLo /usr/share/keyrings/kubernetes-archive-keyring.gpg <https://packages.cloud.google.com/apt/doc/apt-key.gpg>

echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee /etc/apt/sources.list.d/kubernetes.list

sudo apt update -y

sudo apt install kubeadm=1.20.0-00 kubectl=1.20.0-00 kubelet=1.20.0-00 -y

**Master Node**

sudo su

hostnamectl set-hostname master

exec bash

kubeadm init (this will generate the join token command which needs to be executed on the worker node)

  mkdir -p $HOME/.kube

  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config

  sudo chown $(id -u):$(id -g) $HOME/.kube/config

**Deploy CNI**

kubectl apply -f <https://github.com/weaveworks/weave/releases/download/v2.8.1/weave-daemonset-k8s.yaml>

kubectl get nodes -w

**Worker Node**

sudo su

hostnamectl set-hostname worker

exec bash

kubeadm join

**Rollback Cluster / Remove K8**

kubeadm reset

rm -rf /etc/cni

**K8 Deployment using MiniKube**

<https://minikube.sigs.k8s.io/docs/start/>

**Visual Studio**

<https://visualstudio.microsoft.com/downloads/>

Extensions -

1. YAML (Redhat)
2. Kubernetes (Microsoft)

**To Deploy Pod on K8:**

touch pod.yaml

vim pod.yaml

apiVersion: v1

kind: Pod

metadata:

 name: mypod

spec:

 containers:

   - name: myweb

     image: nginx

     ports:

     - containerPort: 80

**To create pod from manifest file:**

kubectl create -f pod.yaml

**To list Pods:**

Kubectl get pods

**To list Pods running on worker node:**

Kubectl get pods -o wide

**To get details of the Pod:**

Kubectl describe pod pod\_name

**To delete pod using manifest file:**

Kubectl delete -f pod.yaml

**To create Replicas of Pod:**

touch replica.yaml

vim replica.yaml

apiVersion: apps/v1

kind: ReplicaSet

metadata:

 name: myreplicaset

spec:

 replicas: 2

 selector:

   matchLabels:

     app: frontend

 template:

   metadata:

    name: myPod

    labels:

      app: frontend

   spec:

    containers:

    - name: myweb

      image: nginx

      ports:

      - containerPort: 80

**To create replicaset from manifest file:**

kubectl create -f replica.yaml

**To list replicaset:**

kubectl get rs

kubectl get pods

**To get info of replicat set:**

kubectl describe rs replicaset\_name

apiVersion: apps/v1

kind: ReplicaSet

metadata:

 name: myreplicaset

spec:

 Replicas: 6

 selector:

   matchLabels:

     app: frontend

 template:

   metadata:

    name: myPod

    labels:

      app: frontend

   spec:

    containers:

    - name: myweb

      image: nginx

      ports:

      - containerPort: 80

**To update an existing replica set:**

Kubectl apply -f replica.yaml

**To delete a replica set:**

Kubectl delete -f replica.yaml

**Open two terminals:**

One terminal connected with master node

Another terminal also connect with the master node and running this command “watch -n 1 "kubectl get pods"”

kubectl get pods

kubectl delete pod pod\_name

**Deployment**

**==========**

mkdir project

cd project

echo “<h1>Hello World v1</h1>” > index.html

vim Dockerfile

FROM nginx

COPY index.html /usr/share/nginx/html

EXPOSE 80

docker build -t myimage:v1 .

echo “<h1>Hello World v2</h1>” > index.html

docker build -t myimage:v2 .

**touch deploy.yaml**

**apiVersion: apps/v1**

**kind: Deployment**

**metadata:**

**name: mydeployment**

**spec:**

**strategy:**

**type: RollingUpdate / Recreate**

**replicas: 10**

**selector:**

**matchLabels:**

**app: frontend**

**template:**

**metadata:**

**name: myPod**

**labels:**

**app: frontend**

**spec:**

**containers:**

**- name: myweb**

**image: gauravdemo06/myimage:v1**

**ports:**

**- containerPort: 80**

**To create the deployment**

kubectl create -f deploy.yaml

**To list the deployment**

kubectl get deploy

**To update the deployment**

**apiVersion: apps/v1**

**kind: Deployment**

**metadata:**

**name: mydeployment**

**spec:**

**strategy:**

**type: RollingUpdate / Recreate**

**replicas: 10**

**selector:**

**matchLabels:**

**app: frontend**

**template:**

**metadata:**

**name: myPod**

**labels:**

**app: frontend**

**spec:**

**containers:**

**- name: myweb**

**image: gauravdemo06/myimage:v2**

**ports:**

**- containerPort: 80**

**kubectl apply -f deploy.yaml**

**Day - 4**

**Wordpress Deployment on K8 - SAAS**

**touch deploy-frontend.yaml**

apiVersion: apps/v1

kind: Deployment

metadata:

  name: myfrontend

spec:

  strategy:

    type: RollingUpdate

  replicas: 2

  selector:

    matchLabels:

      app: frontend

  template:

    metadata:

      name: mypod

      labels:

        app: frontend

    spec:

     containers:

       - name: mywebapp

         image: wordpress

         ports:

         - containerPort: 80

touch deploy-database.yaml

apiVersion: apps/v1

kind: Deployment

metadata:

  name: mydatabase

spec:

  strategy:

    type: RollingUpdate

  replicas: 1

  selector:

    matchLabels:

      app: database

  template:

    metadata:

      name: mydb

      labels:

        app: database

    spec:

     containers:

       - name: mydb

         image: mysql

         env:

            - name: MYSQL\_ROOT\_PASSWORD

              value: redhat

            - name: MYSQL\_DATABASE

              value: wordpress

         ports:

         - containerPort: 3306

touch service-backend.yaml

apiVersion: v1

kind: Service

metadata:

 name: mydbservice

spec:

  type: ClusterIP

  ports:

    - port: 3306

      targetPort: 3306

  selector:

    app: database

touch service-frontend.yaml

apiVersion: v1

kind: Service

metadata:

  name: myfrontend-service

spec:

  type: NodePort

  ports:

    - port: 80

      targetPort: 80

      nodePort: 30000

  selector:

    app: frontend

kubectl create -f **deploy-frontend.yaml**

kubectl create -f **deploy-database.yaml**

kubectl create -f **service-frontend.yaml**

kubectl create -f **service-database.yaml**

**To List services:**

kubectl get services

**To List Endpoints:**

kubectl get endpoints

**To describe service:**

kubectl describe service serivce\_name

touch service-frontend-lb.yaml

apiVersion: v1

kind: Service

metadata:

  name: myfrontend-service-lb

spec:

  type: LoadBalancer

  ports:

    - port: 80

      targetPort: 80

      nodePort: 30001

  selector:

    app: frontend

**Ingress**

**======**

<https://kubernetes.github.io/ingress-nginx/deploy/baremetal/>

Installation - kubectl apply -f <https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v1.1.1/deploy/static/provider/baremetal/deploy.yaml>

kubectl get services --all-namespaces

kubectl get services --all-namespaces | grep ingress | grep NodePort

Deployment

=========

apiVersion: apps/v1

kind: Deployment

metadata:

 name: mydeployment01

spec:

 strategy:

   type: RollingUpdate

 replicas: 4

 selector:

   matchLabels:

     app: frontend01

 template:

   metadata:

    name: myPod

    labels:

      app: frontend01

   spec:

    containers:

    - name: myweb

      image: gauravdemo06/myimage:v1

      ports:

      - containerPort: 80

---

apiVersion: apps/v1

kind: Deployment

metadata:

 name: mydeployment02

spec:

 strategy:

   type: RollingUpdate

 replicas: 4

 selector:

   matchLabels:

     app: frontend02

 template:

   metadata:

    name: myPod

    labels:

      app: frontend02

   spec:

    containers:

    - name: myweb

      image: gauravdemo06/myimage:v2

      ports:

      - containerPort: 80

Services

=======

apiVersion: v1

kind: Service

metadata:

 name: myappservice-01

spec:

  type: ClusterIP

  ports:

    - port: 80

      targetPort: 80

  selector:

    app: frontend01

---

apiVersion: v1

kind: Service

metadata:

 name: myappservice-02

spec:

  type: ClusterIP

  ports:

    - port: 80

      targetPort: 80

  selector:

    app: frontend02

**Ingress Manifest file**

apiVersion: networking.k8s.io/v1

kind: Ingress

metadata:

  name: myingress

  annotations:

    kubernetes.io/ingress.class: nginx

    ingress.kubernetes.io/rewrite-target: /

spec:

  rules:

    - host: "example.com"

      http:

        paths:

          - path: "/"

            pathType: Prefix

            backend:

              service:

                name: myappservice-01

                port:

                  number: 80

    - host: "test.com"

      http:

        paths:

          - path: "/"

            pathType: Prefix

            backend:

              service:

                name: myappservice-02

                port:

                  number: 80

**Local Name Resolution in Linux/Mac**

vim /etc/hosts

Host\_ip domain\_name

18.216.52.33 example.com

**Local Name Resolution in Windows**

1. Go to Start > run Notepad.
2. Right click on the Notepad icon and select Run as a administrator.
3. Select Open from the File menu option.
4. Select All Files (\*.\*) from the file type drop-down.
5. Browse to c:\Windows\System32\drivers\etc
6. Open the hosts file.
7. Add the host name and IP address to the bottom of the host file. The comments in this file show how to format this entry.  
   For example:  
   8.8.8.8    fqdn    
   (Commonly used, when is required to access and external site fqdn, that is different to the internal site, and where DNS is is not available).

Host\_ip domain name

8.8.8.8 example.com

1. Save the host file.

K8 Dashboard

============

kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v2.6.1/aio/deploy/recommended.yaml

kubectl edit service kubernetes-dashboard -n kubernetes-dashboard

Type: ClusterIP

Type: NodePort

**touch cluster.yaml**

apiVersion: v1

kind: ServiceAccount

metadata:

  name: admin-user

  namespace: kubernetes-dashboard

- - -

apiVersion: rbac.authorization.k8s.io/v1

kind: ClusterRoleBinding

metadata:

  name: admin-user

roleRef:

  apiGroup: rbac.authorization.k8s.io

  kind: ClusterRole

  name: cluster-admin

subjects:

- kind: ServiceAccount

  name: admin-user

  namespace: kubernetes-dashboard

kubectl apply -f cluster.yaml

Kubectl get secrets —all-namespace | grep admin-user

kubectl describe secret admin-user-token-r8m8b -n kubernetes-dashboard

**EKS Workshop - https://www.eksworkshop.com/**