# EKS- Day#2 Task

EKS is a fully managed service which is managed by AWS.

It is highly integrated with AWS services such as EC2, EBS, ELB, EFS, and VPC

**Spot Instances**: Below yaml shows how to set up an eks cluster with a mix of spot instances & on demand instances.

Spot instances (slaves nodes) are launched only when bid price is above the spot price. We can set the bid price for spot instances while launching it.

The instances will be launched only when bid price is above the spot price. If the spot price is above bid price Spot instances will be terminated by AWS. There is a pricing associated with Spot instances.

Using eksctl we can create a single node group with mixed instances such as t2.small, t2.micro etc.

```yaml
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig

metadata:
  name: mncluster
  region: ap-south-1

nodeGroups:
  - name: ng1
    desiredCapacity: 2
    instanceType: t2.micro
    ssh:
      publicKeyName: kubeks
  - name: ng2
    desiredCapacity: 1
    instanceType: t2.small
    ssh:
      publicKeyName: kubeks
  - name: ng-mixed
    minSize: 2
    maxSize: 5
    instancesDistribution:
      maxPrice: 0.017
      instanceTypes: ["t3.small", "t3.medium"] # At least one instance type should be specified
      onDemandBaseCapacity: 0
      onDemandPercentageAboveBaseCapacity: 50
      spotInstancePools: 2
    ssh:
      publicKeyName: kubeks
```

EKS Cluster creation through command line

```
Command Prompt - eksctl  create cluster -f D:\EKS\D2\cluster.yml

C:\Program Files\Kubernetes\Minikube>eksctl create cluster -f D:\EKS\D2\cluster.yml
[ℹ]  eksctl version 0.21.0
[ℹ]  using region ap-south-1
[ℹ]  setting availability zones to [ap-south-1a ap-south-1c ap-south-1b]
[ℹ]  subnets for ap-south-1a - public:192.168.0.0/19 private:192.168.96.0/19
[ℹ]  subnets for ap-south-1c - public:192.168.32.0/19 private:192.168.128.0/19
[ℹ]  subnets for ap-south-1b - public:192.168.64.0/19 private:192.168.160.0/19
[ℹ]  nodegroup "ng1" will use "ami-073969767527f7306" [AmazonLinux2/1.16]
[ℹ]  using EC2 key pair "kubeks"
[ℹ]  nodegroup "ng2" will use "ami-073969767527f7306" [AmazonLinux2/1.16]
[ℹ]  using EC2 key pair "kubeks"
[ℹ]  nodegroup "ng-mixed" will use "ami-073969767527f7306" [AmazonLinux2/1.16]
[ℹ]  using EC2 key pair "kubeks"
[ℹ]  using Kubernetes version 1.16
[ℹ]  creating EKS cluster "mncluster" in "ap-south-1" region with un-managed nodes
[ℹ]  3 nodegroups (ng-mixed, ng1, ng2) were included (based on the include/exclude rules)
[ℹ]  will create a CloudFormation stack for cluster itself and 3 nodegroup stack(s)
[ℹ]  will create a CloudFormation stack for cluster itself and 0 managed nodegroup stack(s)
[ℹ]  if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=ap-south-1
--cluster=mncluster'
[ℹ]  CloudWatch logging will not be enabled for cluster "mncluster" in "ap-south-1"
[ℹ]  you can enable it with 'eksctl utils update-cluster-logging --region=ap-south-1 --cluster=mncluster'
[ℹ]  Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "mncluster"
 in "ap-south-1"
[ℹ]  2 sequential tasks: { create cluster control plane "mncluster", 2 sequential sub-tasks: { no tasks, 3 parallel sub-
tasks: { create nodegroup "ng1", create nodegroup "ng2", create nodegroup "ng-mixed" } } }
[ℹ]  building cluster stack "eksctl-mncluster-cluster"
[ℹ]  deploying stack "eksctl-mncluster-cluster"
```

```
[ℹ]  building nodegroup stack "eksctl-mncluster-nodegroup-ng1"
[[0m]  building nodegroup stack "eksctl-mncluster-nodegroup-ng1"
[[0m]  building nodegroup stack "eksctl-mncluster-nodegroup-ng1"

[ℹ]  --nodes-min=2 was set automatically for nodegroup ng2
[ℹ]  --nodes-max=2 was set automatically for nodegroup ng1
[ℹ]  --nodes-max=1 was set automatically for nodegroup ng2
[ℹ]  --nodes-max=1 was set automatically for nodegroup ng2
[ℹ]  deploying stack "eksctl-mncluster-nodegroup-ng-mixed"
[ℹ]  deploying stack "eksctl-mncluster-nodegroup-ng1"
[ℹ]  deploying stack "eksctl-mncluster-nodegroup-ng2"
[!]  retryable error (RequestError: send request failed
caused by: Post "https://cloudformation.ap-south-1.amazonaws.com/": EOF) from cloudformation/DescribeStacks - will retry after delay of 45.028752ms
[ℹ]  waiting for the control plane availability...
[!]  unable to write kubeconfig , please retry with 'eksctl utils write-kubeconfig -n mncluster': unable to read existing kubeconfig file "C:\\Users\\arvind.ramugade/.k
ube/config": error loading config file "C:\Users\arvind.ramugade/.kube/config": read C:\Users\arvind.ramugade/.kube/config: The process cannot access the file because a
nother process has locked a portion of the file.
[ℹ]  no tasks
[ℹ]  all EKS cluster resources for "mncluster" have been created
[ℹ]  adding identity "arn:aws:iam::819612740240:role/eksctl-mncluster-nodegroup-ng1-NodeInstanceRole-V4N82JTSQ86B" to auth ConfigMap
[ℹ]  nodegroup "ng1" has 0 node(s)
[ℹ]  waiting for at least 2 node(s) to become ready in "ng1"
[ℹ]  nodegroup "ng1" has 2 node(s)
[ℹ]  node "ip-192-168-25-50.ap-south-1.compute.internal" is ready
[ℹ]  node "ip-192-168-85-212.ap-south-1.compute.internal" is ready
[ℹ]  adding identity "arn:aws:iam::819612740240:role/eksctl-mncluster-nodegroup-ng2-NodeInstanceRole-ZFTVREN1C1MB" to auth ConfigMap
[ℹ]  nodegroup "ng2" has 0 node(s)
[ℹ]  waiting for at least 1 node(s) to become ready in "ng2"
[ℹ]  nodegroup "ng2" has 1 node(s)
[ℹ]  node "ip-192-168-74-110.ap-south-1.compute.internal" is ready
[ℹ]  adding identity "arn:aws:iam::819612740240:role/eksctl-mncluster-nodegroup-ng-mix-NodeInstanceRole-1CE7ZWIUHOEP2" to auth ConfigMap
[ℹ]  nodegroup "ng-mixed" has 0 node(s)
[ℹ]  waiting for at least 2 node(s) to become ready in "ng-mixed"
[ℹ]  nodegroup "ng-mixed" has 2 node(s)
[ℹ]  node "ip-192-168-14-197.ap-south-1.compute.internal" is ready
[ℹ]  node "ip-192-168-95-56.ap-south-1.compute.internal" is ready
[ℹ]  EKS cluster "mncluster" in "ap-south-1" region is ready
```

EKS Cluster verification through command line

```
C:\Program Files\Kubernetes\Minikube>eksctl get cluster
NAME            REGION
mncluster       ap-south-1
```

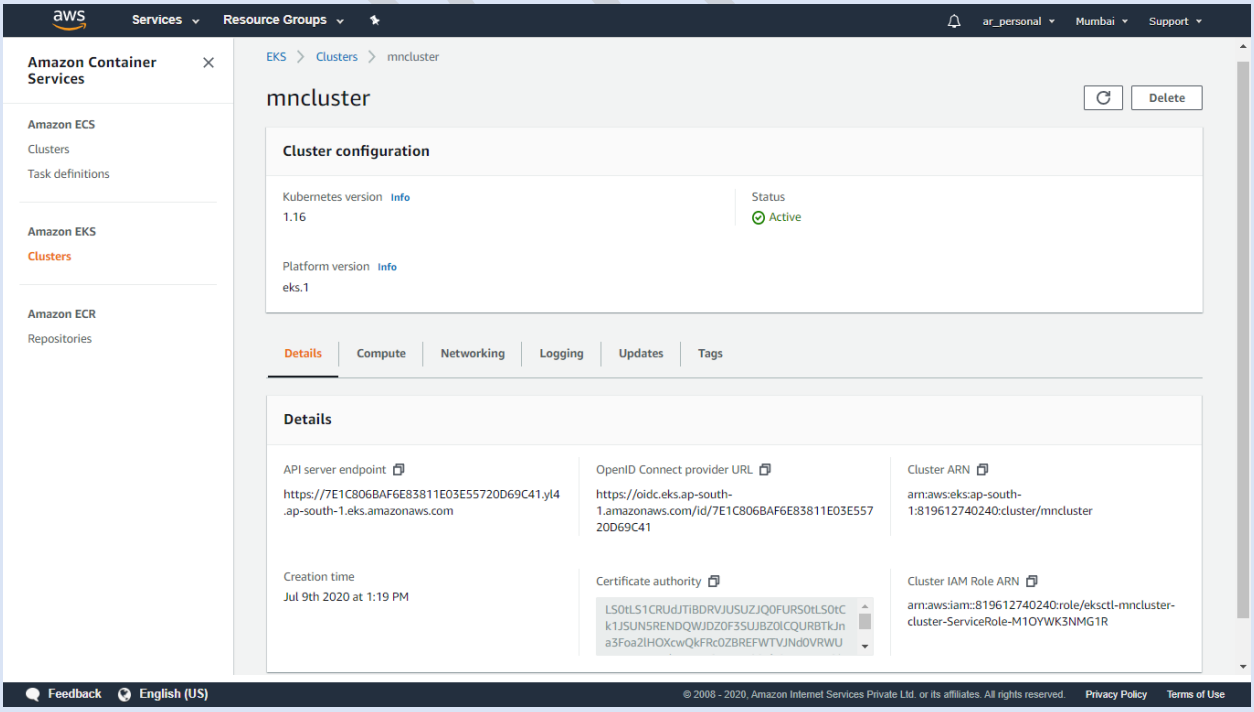
```
C:\Program Files\Kubernetes\Minikube>eksctl get nodegroup --cluster mncluster
CLUSTER         NODEGROUP       CREATED                 MIN SIZE        MAX SIZE        DESIRED CAPACITY        INSTANCE TYPE   IMAGE ID
mncluster       ng-mixed        2020-07-09T08:00:28Z    2               5               0                       t3.small        ami-0739697675 27f7306
mncluster       ng1             2020-07-09T08:00:28Z    2               2               2                       t2.micro        ami-0739697675 27f7306
mncluster       ng2             2020-07-09T08:00:29Z    1               1               1                       t2.small        ami-0739697675 27f7306
```

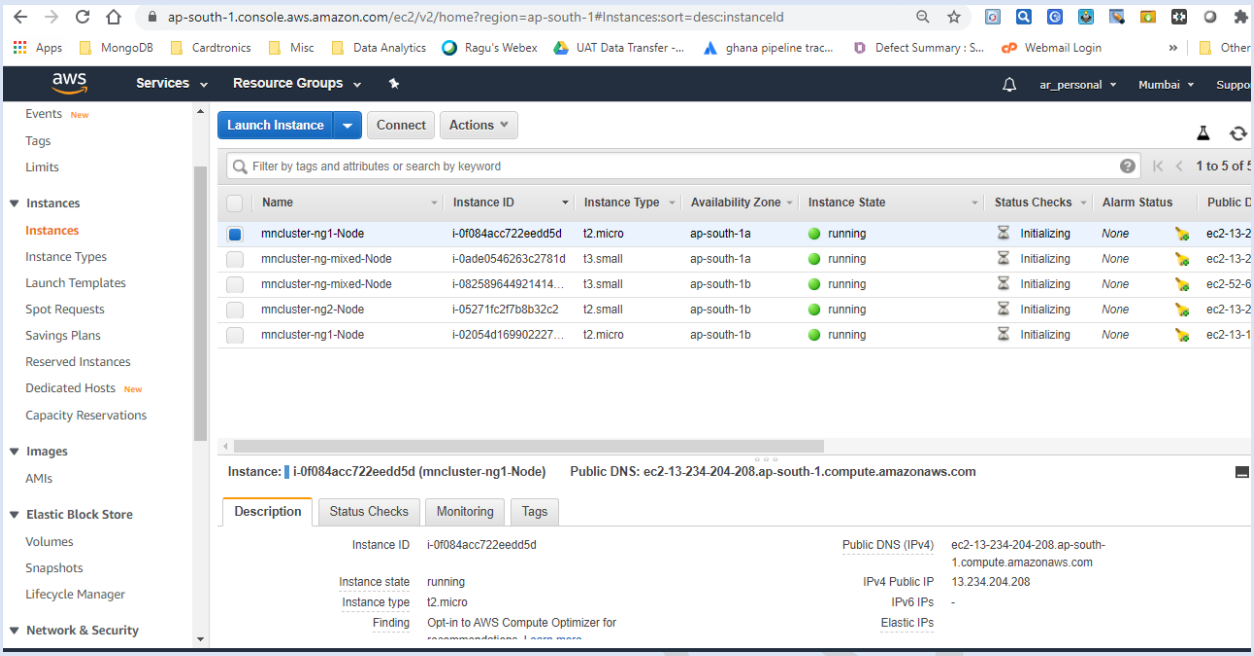
```
C:\Program Files\Kubernetes\Minikube>kubectl cluster-info
Kubernetes master is running at https://7E1C806BAF6E83811E03E55720D69C41.yl4.ap-south-1.eks.amazonaws.com
CoreDNS is running at https://7E1C806BAF6E83811E03E55720D69C41.yl4.ap-south-1.eks.amazonaws.com/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
```

```
C:\Program Files\Kubernetes\Minikube>kubectl config view
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: DATA+OMITTED
    server: https://B281A5F495624F50E70A2F5BB0BBBBF5.yl4.ap-south-1.eks.amazonaws.com
  name: arn:aws:eks:ap-south-1:819612740240:cluster/lwcluster
- cluster:
    certificate-authority-data: DATA+OMITTED
    server: https://7E1C806BAF6E83811E03E55720D69C41.yl4.ap-south-1.eks.amazonaws.com
  name: arn:aws:eks:ap-south-1:819612740240:cluster/mncluster
- cluster:
    certificate-authority-data: DATA+OMITTED
    server: https://F1D498ED792166C8A8DC0775B26535A2.sk1.ap-southeast-1.eks.amazonaws.com
  name: arn:aws:eks:ap-southeast-1:819612740240:cluster/far-lwcluster
- cluster:
    certificate-authority: C:\Users\arvind.ramugade\.minikube\ca.crt
    server: https://192.168.99.100:8443
  name: minikube
contexts:
- context:
    cluster: arn:aws:eks:ap-south-1:819612740240:cluster/lwcluster
    user: arn:aws:eks:ap-south-1:819612740240:cluster/lwcluster
  name: arn:aws:eks:ap-south-1:819612740240:cluster/lwcluster
- context:
    cluster: arn:aws:eks:ap-south-1:819612740240:cluster/mncluster
    namespace: mnns
    user: arn:aws:eks:ap-south-1:819612740240:cluster/mncluster
  name: arn:aws:eks:ap-south-1:819612740240:cluster/mncluster
- context:
    cluster: arn:aws:eks:ap-southeast-1:819612740240:cluster/far-lwcluster
    user: arn:aws:eks:ap-southeast-1:819612740240:cluster/far-lwcluster
  name: arn:aws:eks:ap-southeast-1:819612740240:cluster/far-lwcluster
- context:
    cluster: minikube
    user: minikube
  name: minikube
current-context: arn:aws:eks:ap-south-1:819612740240:cluster/mncluster
kind: Config
preferences: {}
users:
```

```
- name: arn:aws:eks:ap-south-1:819612740240:cluster/lwcluster
  user:
    exec:
      apiVersion: client.authentication.k8s.io/v1alpha1
      args:
      - --region
      - ap-south-1
      - eks
      - get-token
      - --cluster-name
      - lwcluster
      command: aws
      env: null
- name: arn:aws:eks:ap-south-1:819612740240:cluster/mncluster
  user:
    exec:
      apiVersion: client.authentication.k8s.io/v1alpha1
      args:
      - --region
      - ap-south-1
      - eks
      - get-token
      - --cluster-name
      - mncluster
      command: aws
      env: null
- name: arn:aws:eks:ap-southeast-1:819612740240:cluster/far-lwcluster
  user:
    exec:
      apiVersion: client.authentication.k8s.io/v1alpha1
      args:
      - --region
      - ap-southeast-1
      - eks
      - get-token
      - --cluster-name
      - far-lwcluster
      command: aws
      env: null
- name: minikube
  user:
    client-certificate: C:\Users\arvind.ramugade\.minikube\profiles\minikube\client.crt
    client-key: C:\Users\arvind.ramugade\.minikube\profiles\minikube\client.key
```

We can verify the cluster creation through AWS Console as well as shown below





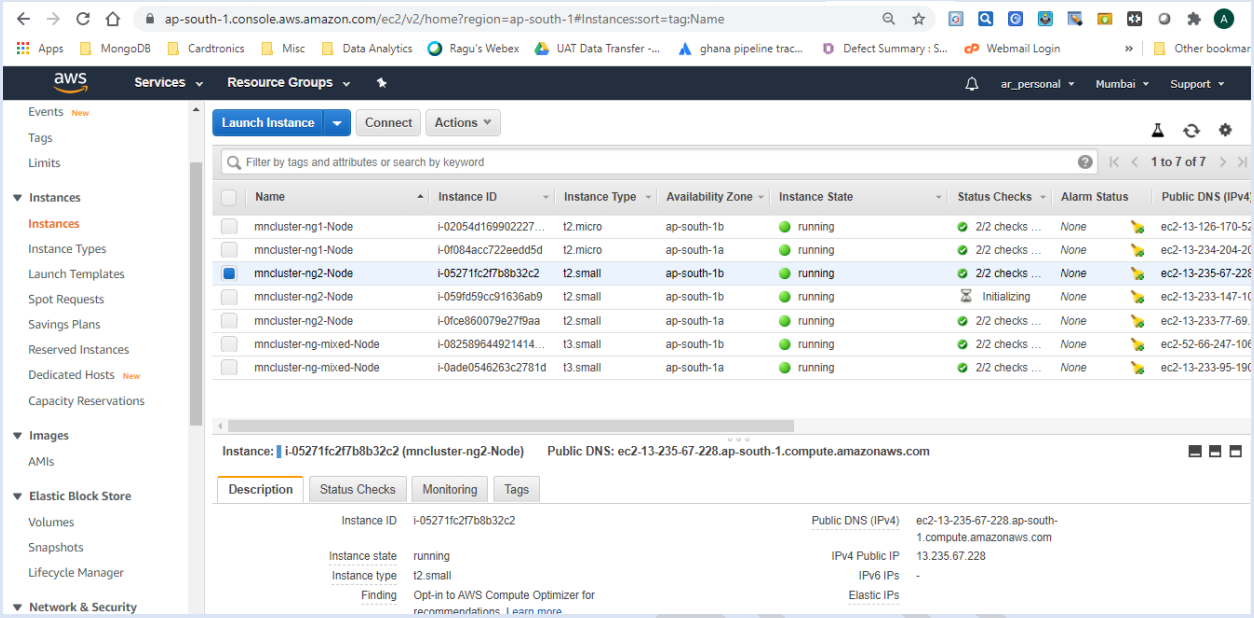

Using below command we can update kube-config file so that we can use kubectl command in EKS cluster

aws eks update-kubeconfig --name=mncluster

kubectl get nodes

We can scale our cluster for a particular node group as under:

```
C:\Program Files\Kubernetes\Minikube>eksctl scale nodegroup --cluster mncluster --name ng2 --nodes=3 --nodes-max=5
[ℹ]  scaling nodegroup stack "eksctl-mncluster-nodegroup-ng2" in cluster eksctl-mncluster-cluster
[ℹ]  scaling nodegroup, desired capacity from 1 to 3, max size from 1 to 5
```




```
C:\Program Files\Kubernetes\Minikube>eksctl get nodegroup --cluster mncluster
CLUSTER         NODEGROUP       CREATED                 MIN SIZE        MAX SIZE        DESIRED CAPACITY        INSTANCE TYPE   IMAGE ID
mncluster       ng-mixed        2020-07-09T08:00:28Z    2               5               0                       t3.small        ami-073969767527f7306
mncluster       ng1             2020-07-09T08:00:28Z    2               2               2                       t2.micro        ami-073969767527f7306
mncluster       ng2             2020-07-09T08:00:29Z    1               5               3                       t2.small        ami-073969767527f7306

C:\Program Files\Kubernetes\Minikube>
```

```
C:\Program Files\Kubernetes\Minikube>kubectl create namespace mnns
namespace/mnns created

C:\Program Files\Kubernetes\Minikube>kubectl config set-context --current --namespace=mnns
Context "arn:aws:eks:ap-south-1:819612740240:cluster/mncluster" modified.

C:\Program Files\Kubernetes\Minikube>
```

By default in a container there is no connectivity for pods running on multiple nodes. If there are multiple pods in a single slave node they can communicate with each other, however they can't connect with pods in another slave.

Using CNI (also known as flannel)we can achieve this. CNI, VPC, Subnet created by Eksctl automatically once we set up them multi node cluster (1master& 2 slaves)

k8s_coredns manages outside network connectivity.

Limit on no. of pods in a node:
Following command shows maximum no. of podswhich we can launch
ps aux | grep kubectl

The limit on no. of pods which can be launched in a node varies based on instance type. e.g. for t2.micro instance it has capability of 4 NIC of which two are used for instance IP address only two NIC are available and we can run only two pods in the t2.micro.

Network interface attached by AWS CNI for pods help in inter connection with other pods on different nodes.

```
root@ip-192-168-85-212:~
[root@ip-192-168-85-212 ~]# ps aux | grep kubelet
root      4576  1.5  8.8 864464 89064 ?        Ssl  08:04   0:03 /usr/bin/kubelet --node-ip=192.168.85.212 --node-labels=alpha.ek
sctl.io/cluster-name=mncluster,alpha.eksctl.io/nodegroup-name=ng1,alpha.eksctl.io/instance-id=i-02054d16990222776 --max-pods=4 --
register-node=true --register-with-taints= --cloud-provider=aws --container-runtime=docker --network-plugin=cni --cni-bin-dir=/op
t/cni/bin --cni-conf-dir=/etc/cni/net.d --pod-infra-container-image=602401143452.dkr.ecr.ap-south-1.amazonaws.com/eks/pause-amd64
:3.1 --kubeconfig=/etc/eksctl/kubeconfig.yaml --config=/etc/eksctl/kubelet.yaml
root     11112  0.0  0.0 119420   944 pts/0    S+   08:07   0:00 grep --color=auto kubelet
[root@ip-192-168-85-212 ~]#
```

```
root@ip-192-168-85-212:~
[root@ip-192-168-85-212 ~]# ps aux | grep kubelet
root      4576  1.5  8.8 864464 89064 ?        Ssl  08:04   0:03 /usr/bin/kubelet --node-ip=192.168.85.212 --node-labels=alpha.ek
sctl.io/cluster-name=mncluster,alpha.eksctl.io/nodegroup-name=ng1,alpha.eksctl.io/instance-id=i-02054d16990222776 --max-pods=4 --
register-node=true --register-with-taints= --cloud-provider=aws --container-runtime=docker --network-plugin=cni --cni-bin-dir=/op
t/cni/bin --cni-conf-dir=/etc/cni/net.d --pod-infra-container-image=602401143452.dkr.ecr.ap-south-1.amazonaws.com/eks/pause-amd64
:3.1 --kubeconfig=/etc/eksctl/kubeconfig.yaml --config=/etc/eksctl/kubelet.yaml
root     11112  0.0  0.0 119420   944 pts/0    S+   08:07   0:00 grep --color=auto kubelet
[root@ip-192-168-85-212 ~]#
```

```
root@ip-192-168-85-212:~
[root@ip-192-168-85-212 ~]# ifconfig -a
eni36209f359b9: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 9001
        inet6 fe80::6821:3eff:fe49:8bb6  prefixlen 64  scopeid 0x20<link>
        ether 6a:21:3e:49:8b:b6  txqueuelen 0  (Ethernet)
        RX packets 428  bytes 36451 (35.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 427  bytes 144354 (140.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enibc3cd0f2e6b: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 9001
        inet6 fe80::d032:aff:fe73:9483  prefixlen 64  scopeid 0x20<link>
        ether d2:32:0a:73:94:83  txqueuelen 0  (Ethernet)
        RX packets 475  bytes 40346 (39.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 477  bytes 158569 (154.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 9001
        inet 192.168.85.212  netmask 255.255.224.0  broadcast 192.168.95.255
        inet6 fe80::8bd:d3ff:fe05:1130  prefixlen 64  scopeid 0x20<link>
        ether 0a:bd:d3:05:11:30  txqueuelen 1000  (Ethernet)
        RX packets 153354  bytes 219122950 (208.9 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 9653  bytes 996491 (973.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 9001
        inet 192.168.66.4  netmask 255.255.224.0  broadcast 192.168.95.255
        inet6 fe80::8ff:39ff:fe4c:b8b4  prefixlen 64  scopeid 0x20<link>
        ether 0a:ff:39:4c:b8:b4  txqueuelen 1000  (Ethernet)
        RX packets 280  bytes 126361 (123.3 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 275  bytes 21708 (21.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 1514  bytes 115587 (112.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1514  bytes 115587 (112.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

We can confirm that Docker is already installed in the instance due to EKS cluster.

```
[root@ip-192-168-85-212 ~]# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)
   Active: active (running) since Thu 2020-07-09 08:04:27 UTC; 6min ago
     Docs: https://docs.docker.com
 Main PID: 3964 (dockerd)
    Tasks: 13
   Memory: 367.6M
   CGroup: /system.slice/docker.service
           └─3964 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Jul 09 08:05:25 ip-192-168-85-212.ap-south-1.compute.internal dockerd[3964]: time="2020-07-09T08:05:25.370736969Z" level=info msg="ignoring event" module=libc...Delete
Jul 09 08:05:26 ip-192-168-85-212.ap-south-1.compute.internal dockerd[3964]: time="2020-07-09T08:05:26.374255808Z" level=info msg="ignoring event" module=libc...Delete
Jul 09 08:05:27 ip-192-168-85-212.ap-south-1.compute.internal dockerd[3964]: time="2020-07-09T08:05:27.411124168Z" level=info msg="ignoring event" module=libc...Delete
Jul 09 08:05:28 ip-192-168-85-212.ap-south-1.compute.internal dockerd[3964]: time="2020-07-09T08:05:28.414021207Z" level=info msg="ignoring event" module=libc...Delete
Jul 09 08:05:29 ip-192-168-85-212.ap-south-1.compute.internal dockerd[3964]: time="2020-07-09T08:05:29.470419680Z" level=info msg="ignoring event" module=libc...Delete
Jul 09 08:05:30 ip-192-168-85-212.ap-south-1.compute.internal dockerd[3964]: time="2020-07-09T08:05:30.489866328Z" level=info msg="ignoring event" module=libc...Delete
Jul 09 08:05:31 ip-192-168-85-212.ap-south-1.compute.internal dockerd[3964]: time="2020-07-09T08:05:31.508950762Z" level=info msg="ignoring event" module=libc...Delete
Jul 09 08:05:32 ip-192-168-85-212.ap-south-1.compute.internal dockerd[3964]: time="2020-07-09T08:05:32.523186266Z" level=info msg="ignoring event" module=libc...Delete
Jul 09 08:05:33 ip-192-168-85-212.ap-south-1.compute.internal dockerd[3964]: time="2020-07-09T08:05:33.572789446Z" level=info msg="ignoring event" module=libc...Delete
Jul 09 08:05:34 ip-192-168-85-212.ap-south-1.compute.internal dockerd[3964]: time="2020-07-09T08:05:34.610436339Z" level=info msg="ignoring event" module=libc...Delete
Hint: Some lines were ellipsized, use -l to show in full.
[root@ip-192-168-85-212 ~]#
```

```
C:\Program Files\Kubernetes\Minikube>aws eks update-kubeconfig --name=mncluster
Added new context arn:aws:eks:ap-south-1:819612740240:cluster/mncluster to C:\Users\arvind.ramugade\.kube\config
```

```
D:\EKS\D2>kubectl create -k .
secret/mysql-pass-md7m2684d9 created
service/wordpress-mysql created
service/wordpress created
deployment.apps/wordpress-mysql created
deployment.apps/wordpress created
```

```
D:\EKS\D2>kubectl get pods
NAME                              READY   STATUS    RESTARTS   AGE
wordpress-7b5d46f9b9-9clhq        0/1     Pending   0          3m52s
wordpress-mysql-85b46f8cd7-jj2jp  0/1     Pending   0          3m52s
```

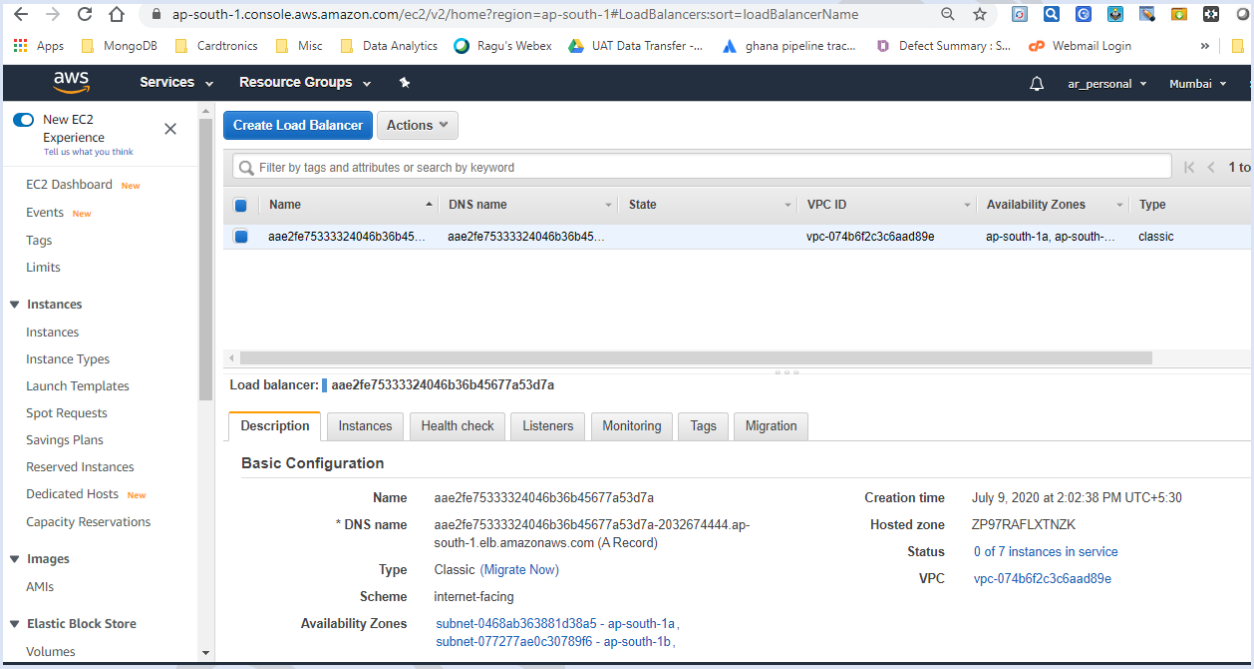
```
D:\EKS\D2>kubectl create -f D:\EKS\D2\create-storage.yaml
storageclass.storage.k8s.io/aws-efs created
persistentvolumeclaim/efs-wp created
persistentvolumeclaim/efs-sql created

D:\EKS\D2>kubectl get pvc
NAME      STATUS    VOLUME   CAPACITY   ACCESS MODES   STORAGECLASS   AGE
efs-sql   Pending                                      aws-efs        27s
efs-wp    Pending                                      aws-efs        27s

D:\EKS\D2>kubectl get pv
No resources found in mnns namespace.

D:\EKS\D2>kubectl get pods
NAME                              READY   STATUS    RESTARTS   AGE
wordpress-7b5d46f9b9-9clhq        0/1     Pending   0          9m29s
wordpress-mysql-85b46f8cd7-jj2jp  0/1     Pending   0          9m29s
```

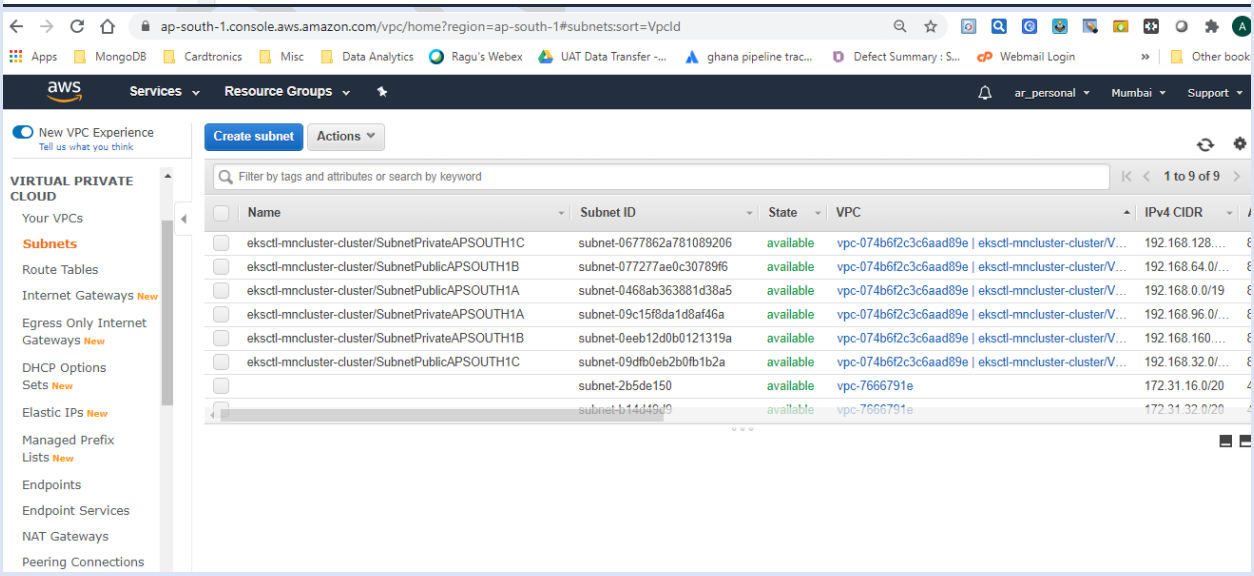
EKS Supports LoadBalancer service which automatically balances traffic on slave nodes.

Kubernetes service type is LoadBalancer & by default it uses Classic Load Balancer which are public facing.

*service.beta.kubernetes.io/aws-load-balancer-type: nlb*

Using annotations we can manage LoadBalancer configuration.

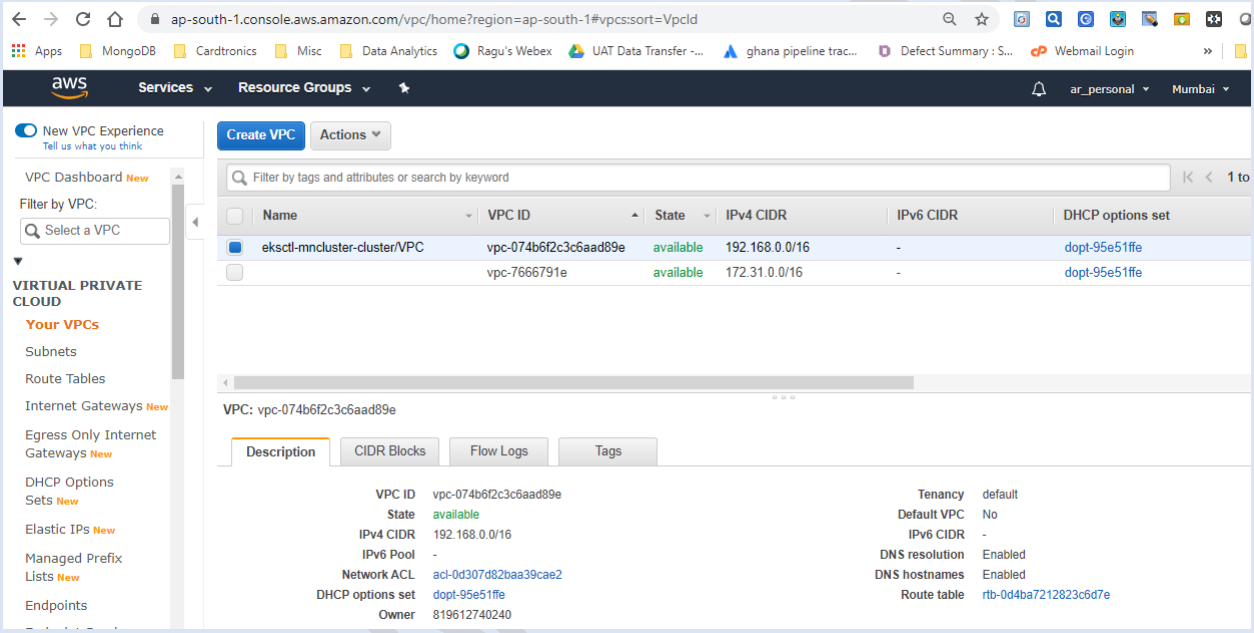


**EBS challenge:**

With EBS we can attach only one EBS volume at a time with the EC2 instance.EBS volume can't be used to connect to EC2 instance launched in different availability zones.This is a problem in a multi node cluster set up where multiple slave nodes can't be attached to centralized storage.

This can be resolved by using **EFS service**. **EFS is a centralized NFS storage** & it spans subnets in VPC. However, the Security group should be assigned to all the subsets while creating EFS.In EFS multiple nodes can concurrently access the storage without any performance/operational overhead.



ap-south-1.console.aws.amazon.com/efs/home?region=ap-south-1#/wizard/1

Step 1: Configure network access
Step 2: Configure file system settings
Step 3: Configure client access
Step 4: Review and create

## Configure network access

An Amazon EFS file system is accessed by EC2 instances running inside one of your VPCs. Instances connect to a file system by using a network interface called a mount target. Each mount target has an IP address, which we assign automatically or you can specify.

VPC: vpc-074b6f2c3c6aad89...

### Create mount targets

Instances connect to a file system by using mount targets you create. We recommend creating a mount target in each of your VPC's Availability Zones so that EC2 instances across your VPC can access the file system.

| Availability Zone | Subnet | IP address | Security groups |
|---|---|---|---|
| ap-south-1a | subnet-0468ab363881d38a5 - eksctl-mncluster-cluster/SubnetPublicAPSOUTH1A | Automatic | sg-0485a4fc4485f35bd - eksctl-mncluster-nodegroup-ng1-SG-1NWXYOVX64WNS<br>sg-08fb1f78f78739104 - eksctl-mncluster-cluster-ClusterSharedNodeSecurityGroup-BLAWDRQ99Y22 |
| ap-south-1b | subnet-077277ae0c30789f6 - eksctl-mncluster-cluster/SubnetPublicAPSOUTH1B | Automatic | sg-0485a4fc4485f35bd - eksctl-mncluster-nodegroup-ng1-SG-1NWXYOVX64WNS<br>sg-08fb1f78f78739104 - eksctl-mncluster-cluster-ClusterSharedNodeSecurityGroup-BLAWDRQ99Y22 |
| | | | sg-0485a4fc4485f35bd - eksctl-mncluster-nodegroup-ng1-SG-1NWXYOVX64WNS |

ap-south-1.console.aws.amazon.com/efs/home?region=ap-south-1#/filesystems/fs-d5dc5604

aws | Services | Resource Groups | ar_personal | Mumbai | Support

File systems
AWS DataSync
AWS Backup

Reduce your storage price to $0.08/GB-month* with EFS Lifecycle Management. Learn more

* pricing in US East (N. Virginia) region, assumes 80% of your storage in EFS IA

What's New | Documentation | AWS Storage Blog

## File systems

Create file system | Actions

| Name | File system ID | Metered size | Number of mount targets | Creation date |
|---|---|---|---|---|
| EFS for Wordpress | fs-d5dc5604 | 6.0 KiB | 3 | 07/09/2020, 08:21:48 UTC |

### Other details

| | |
|---|---|
| Owner ID | 819612740240 |
| File system state | Available |
| Performance mode | General Purpose |
| Throughput mode | Bursting |
| Encrypted | No |
| Lifecycle policy | None |

### Tags

Manage tag

Name: EFS for Wordpress

**Mount targets**

| VPC | Availability Zone | Subnet | IP address | Mount target ID | Network interface ID | Security groups | Mount target state |
|---|---|---|---|---|---|---|---|
| vpc-074b6f2c3c6aad89e - eksctl-mncluster-cluster/VPC | ap-south-1c | subnet-0677862a781089206 - eksctl-mncluster-cluster/SubnetPrivateAPSOUTH1C | 192.168.158.107 | fsmt-e132e430 | eni-0a1efa54c1358d903 | sg-0485a4fc4485f35bd - eksctl-mncluster-nodegroup-ng1-SG-1NWXYOVX64WNS sg-08fb1f78f78739104 - eksctl-mncluster-cluster-ClusterSharedNodeSecurityGroup-BLAWDRQ99Y22 | Available |
| | ap-south-1b | subnet-077277ae0c30789f6 - eksctl-mncluster-cluster/SubnetPublicAPSOUTH1B | 192.168.88.52 | fsmt-e332e432 | eni-0a4e7d272a730717a | sg-0485a4fc4485f35bd - eksctl-mncluster-nodegroup-ng1-SG-1NWXYOVX64WNS sg-08fb1f78f78739104 - eksctl-mncluster-cluster-ClusterSharedNodeSecurityGroup-BLAWDRQ99Y22 | Available |
| | ap-south-1a | subnet-0468ab363881d38a5 - eksctl-mncluster-cluster/SubnetPublicAPSOUTH1A | 192.168.3.160 | fsmt-e232e433 | eni-0b515ffb71fa6e4e6 | sg-0485a4fc4485f35bd - eksctl-mncluster-nodegroup-ng1-SG-1NWXYOVX64WNS sg-08fb1f78f78739104 - eksctl-mncluster-cluster-ClusterSharedNodeSecurityGroup-BLAWDRQ99Y22 | Available |

```
D:\EKS\D2>kubectl create -f D:\EKS\D2\create-efs-provisioner.yaml
deployment.apps/efs-provisioner created
```

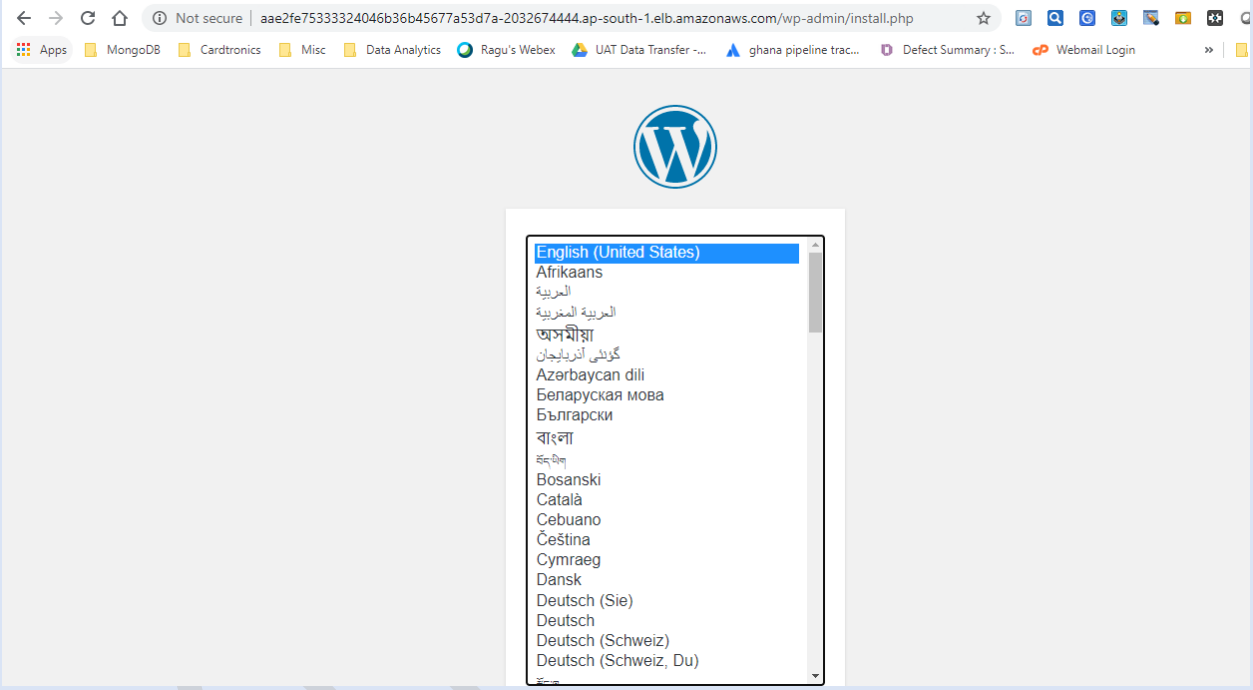
```
D:\EKS\D2>kubectl create -f D:\EKS\D2\create-rbac.yaml
clusterrolebinding.rbac.authorization.k8s.io/nfs-provisioner-role-binding created

D:\EKS\D2>kubectl get pods
NAME                              READY   STATUS              RESTARTS   AGE
efs-provisioner-6f54dcd88b-4bh2p  1/1     Running             0          6m14s
wordpress-7b5d46f9b9-9clhq        0/1     ContainerCreating   0          18m
wordpress-mysql-85b46f8cd7-jj2jp  0/1     ContainerCreating   0          18m

D:\EKS\D2>kubectl get pvc
NAME      STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
efs-sql   Bound    pvc-61787c66-27ec-41bd-b19d-e721fefb1fcc   10Gi       RWX            aws-efs        10m
efs-wp    Bound    pvc-2946c355-1a9d-4199-93a7-eb0e7e5e240a   10Gi       RWX            aws-efs        10m

D:\EKS\D2>kubectl get pv
NAME                                       CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS   CLAIM          STORAGECLASS   REASON   AGE
pvc-2946c355-1a9d-4199-93a7-eb0e7e5e240a   10Gi       RWX            Delete           Bound    mnns/efs-wp    aws-efs                 13m
pvc-61787c66-27ec-41bd-b19d-e721fefb1fcc   10Gi       RWX            Delete           Bound    mnns/efs-sql   aws-efs                 13m

D:\EKS\D2>kubectl get pods
NAME                              READY   STATUS    RESTARTS   AGE
efs-provisioner-6f54dcd88b-4bh2p  1/1     Running   0          20m
wordpress-7b5d46f9b9-9clhq        1/1     Running   0          32m
wordpress-mysql-85b46f8cd7-jj2jp  1/1     Running   0          32m
```

```
D:\EKS\D2>kubectl get secret
NAME                  TYPE                                  DATA   AGE
default-token-jnxdp   kubernetes.io/service-account-token   3      42m
mysql-pass-md7m2684d9 Opaque                                1      34m
```

```
D:\EKS\D2>
D:\EKS\D2>kubectl get sc
NAME            PROVISIONER             AGE
aws-efs         lw-course/aws-efs       27m
gp2 (default)   kubernetes.io/aws-ebs   69m
```

```
D:\EKS\D2>kubectl get all
NAME                                     READY   STATUS    RESTARTS   AGE
pod/efs-provisioner-6f54dcd88b-4bh2p     1/1     Running   0          23m
pod/wordpress-7b5d46f9b9-9clhq           1/1     Running   0          35m
pod/wordpress-mysql-85b46f8cd7-jj2jp     1/1     Running   0          35m

NAME                      TYPE           CLUSTER-IP      EXTERNAL-IP                                                              PORT(S)        AGE
service/wordpress         LoadBalancer   10.100.24.156   aae2fe75333324046b36b45677a53d7a-2032674444.ap-south-1.elb.amazonaws.com   80:31597/TCP   35m
service/wordpress-mysql   ClusterIP      None            <none>                                                                   3306/TCP       35m

NAME                                 READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/efs-provisioner      1/1     1            1           23m
deployment.apps/wordpress            1/1     1            1           35m
deployment.apps/wordpress-mysql      1/1     1            1           35m

NAME                                            DESIRED   CURRENT   READY   AGE
replicaset.apps/efs-provisioner-6f54dcd88b      1         1         1       23m
replicaset.apps/wordpress-7b5d46f9b9            1         1         1       35m
replicaset.apps/wordpress-mysql-85b46f8cd7      1         1         1       35m
```

## WordPress Application



Not secure | aae2fe75333324046b36b45677a53d7a-2032674444.ap-south-1.elb.amazonaws.com/wp-admin/install...

Apps | MongoDB | Cardtronics | Misc | Data Analytics | Ragu's Webex | UAT Data Transfer -... | ghana pipeline trac... | Defect Summary : S... | Webmail Login

## Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

## Information needed

Please provide the following information. Don't worry, you can always change these settings later.

**Site Title** AWS-EKS

**Username** Arvind

Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

**Password** &#7ZZGR9cu)^YeY!fB Hide

Strong

**Important:** You will need this password to log in. Please store it in a secure location.

**Your Email** arvindbramugade@gmail.com

Double-check your email address before continuing.

**Search Engine Visibility** ☐ Discourage search engines from indexing this site

Not secure | aae2fe75333324046b36b45677a53d7a-2032674444.ap-south-1.elb.amazonaws.com/wp-admin/install...

Apps | MongoDB | Cardtronics | Misc | Data Analytics | Ragu's Webex | UAT Data Transfer -... | ghana pipeline trac... | Defect Summary : S... | Webmail Lo

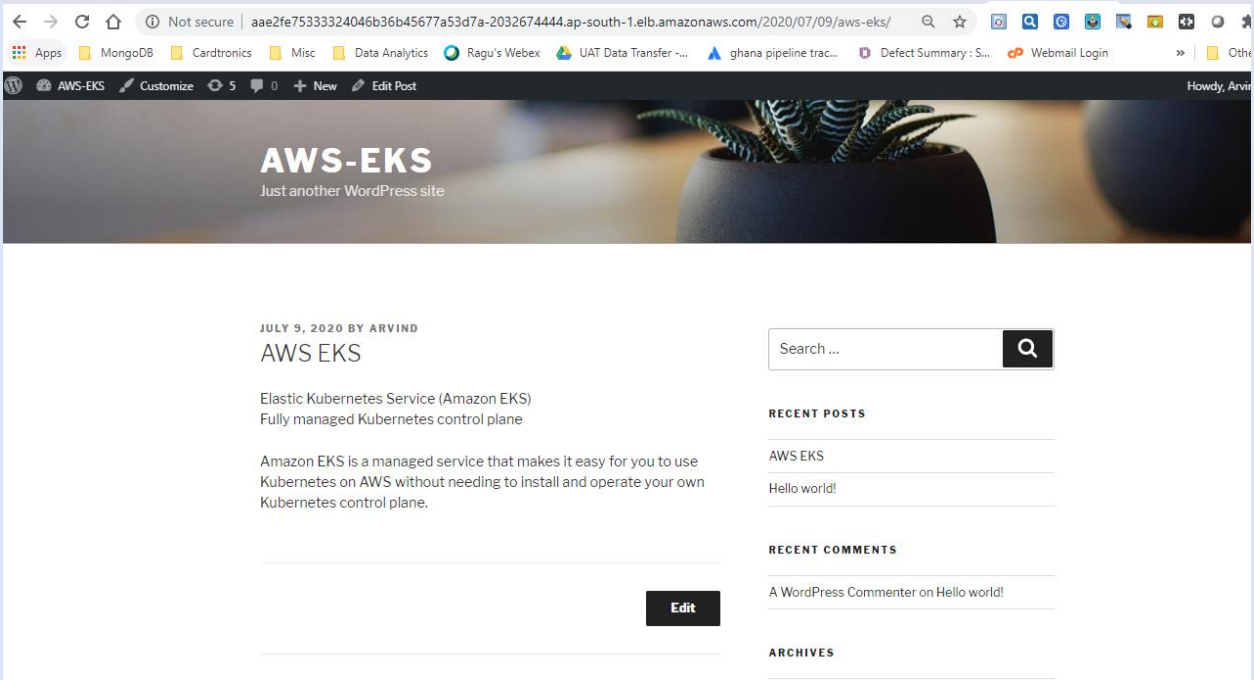
## Success!

WordPress has been installed. Thank you, and enjoy!

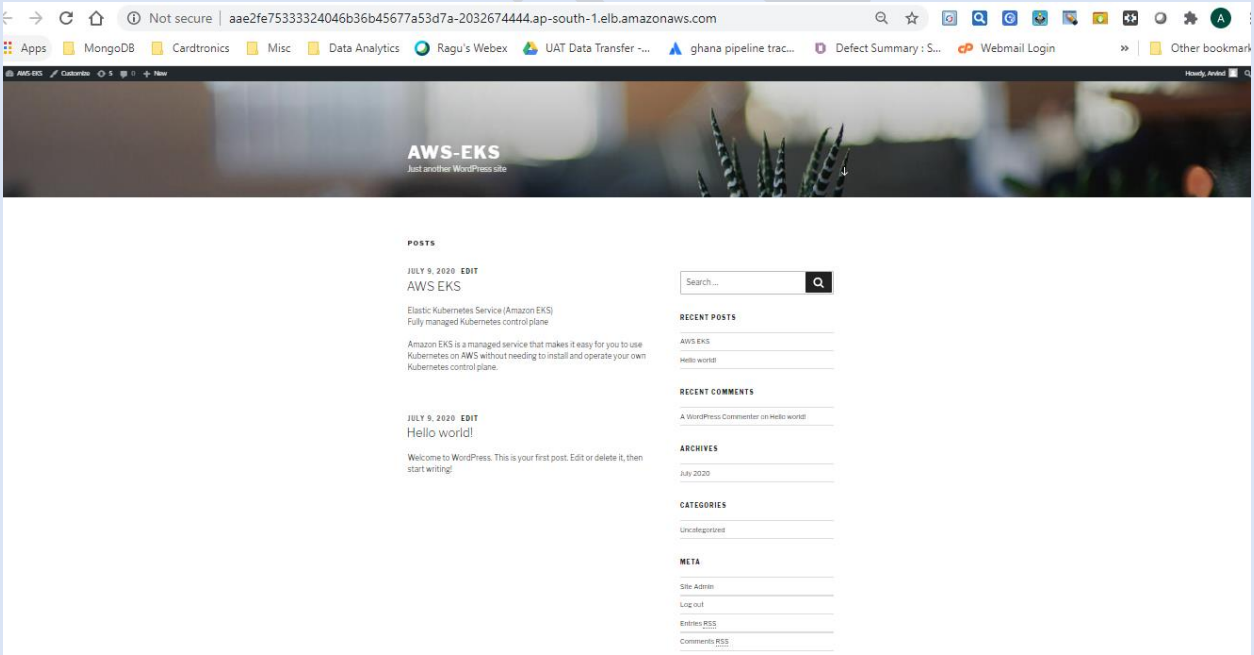
**Username** Arvind

**Password** *Your chosen password.*

Log In




After clicking on LoadBalancer following page will be opened



```
D:\EKS\D2>kubectl get pods -n kube-system -o wide
NAME                      READY   STATUS    RESTARTS   AGE   IP               NODE                                          NOMINATED NODE   READINESS GATES
aws-node-4pnsn            1/1     Running   0          82m   192.168.85.212   ip-192-168-85-212.ap-south-1.compute.internal   <none>           <none>
aws-node-b6kr7            1/1     Running   0          70m   192.168.65.84    ip-192-168-65-84.ap-south-1.compute.internal    <none>           <none>
aws-node-fgz5v            1/1     Running   0          71m   192.168.16.140   ip-192-168-16-140.ap-south-1.compute.internal   <none>           <none>
aws-node-l5d9t            1/1     Running   0          80m   192.168.14.197   ip-192-168-14-197.ap-south-1.compute.internal   <none>           <none>
aws-node-ng5c9            1/1     Running   0          80m   192.168.95.56    ip-192-168-95-56.ap-south-1.compute.internal    <none>           <none>
aws-node-t9dn6            1/1     Running   0          81m   192.168.74.110   ip-192-168-74-110.ap-south-1.compute.internal   <none>           <none>
aws-node-wxp72            1/1     Running   0          82m   192.168.25.50    ip-192-168-25-50.ap-south-1.compute.internal    <none>           <none>
coredns-6856799b8d-9nfjw  1/1     Running   0          89m   192.168.82.90    ip-192-168-85-212.ap-south-1.compute.internal   <none>           <none>
coredns-6856799b8d-txffd  1/1     Running   0          89m   192.168.82.102   ip-192-168-85-212.ap-south-1.compute.internal   <none>           <none>
kube-proxy-6rnj9          1/1     Running   0          80m   192.168.95.56    ip-192-168-95-56.ap-south-1.compute.internal    <none>           <none>
kube-proxy-9hfvk          1/1     Running   0          71m   192.168.16.140   ip-192-168-16-140.ap-south-1.compute.internal   <none>           <none>
kube-proxy-d9n5n          1/1     Running   0          82m   192.168.85.212   ip-192-168-85-212.ap-south-1.compute.internal   <none>           <none>
kube-proxy-mn8rt          1/1     Running   0          70m   192.168.65.84    ip-192-168-65-84.ap-south-1.compute.internal    <none>           <none>
kube-proxy-qjwsq          1/1     Running   0          81m   192.168.74.110   ip-192-168-74-110.ap-south-1.compute.internal   <none>           <none>
kube-proxy-tv6wv          1/1     Running   0          82m   192.168.25.50    ip-192-168-25-50.ap-south-1.compute.internal    <none>           <none>
kube-proxy-wncw9          1/1     Running   0          80m   192.168.14.197   ip-192-168-14-197.ap-south-1.compute.internal   <none>           <none>

D:\EKS\D2>
```

**Helm**

Helm is a client side program that provides the k8s software packages where we can launch the whole application in the kubernetes cluster.

**Tiller**

Tiller is a server side program to help the helm to set up the whole infrastructure.

Initializing Helm

```
D:\EKS\D2>helm init
Creating C:\Users\arvind.ramugade\.helm
Creating C:\Users\arvind.ramugade\.helm\repository
Creating C:\Users\arvind.ramugade\.helm\repository\cache
Creating C:\Users\arvind.ramugade\.helm\repository\local
Creating C:\Users\arvind.ramugade\.helm\plugins
Creating C:\Users\arvind.ramugade\.helm\starters
Creating C:\Users\arvind.ramugade\.helm\cache\archive
Creating C:\Users\arvind.ramugade\.helm\repository\repositories.yaml
Adding stable repo with URL: https://kubernetes-charts.storage.googleapis.com
Adding local repo with URL: http://127.0.0.1:8879/charts
$HELM_HOME has been configured at C:\Users\arvind.ramugade\.helm.

Tiller (the Helm server-side component) has been installed into your Kubernetes Cluster.

Please note: by default, Tiller is deployed with an insecure 'allow unauthenticated users' policy.
To prevent this, run `helm init` with the --tiller-tls-verify flag.
For more information on securing your installation see: https://v2.helm.sh/docs/securing_installation/
```

```
D:\EKS\D2>helm repo list
NAME    URL
stable  https://kubernetes-charts.storage.googleapis.com
local   http://127.0.0.1:8879/charts

D:\EKS\D2>
```

Creating Tiller Service

```
Run 'kubectl --help' for usage.

D:\EKS\D2>
D:\EKS\D2>kubectl -n kube-system create service account tiller
error: unknown command "account tiller"
See 'kubectl create service -h' for help and examples

D:\EKS\D2>kubectl -n kube-system create serviceaccount tiller
serviceaccount/tiller created

D:\EKS\D2>kubectl create clusterrolebinding tiller --clusterrole cluster-admin --serviceaccount=kube-system:tiller
clusterrolebinding.rbac.authorization.k8s.io/tiller created
```

```
D:\EKS\D2>helm init --service-account tiller --upgrade
$HELM_HOME has been configured at C:\Users\arvind.ramugade\.helm.

Tiller (the Helm server-side component) has been updated to gcr.io/kubernetes-helm/tiller:v2.16.9 .
```

**Installation of Prometheus using helm**

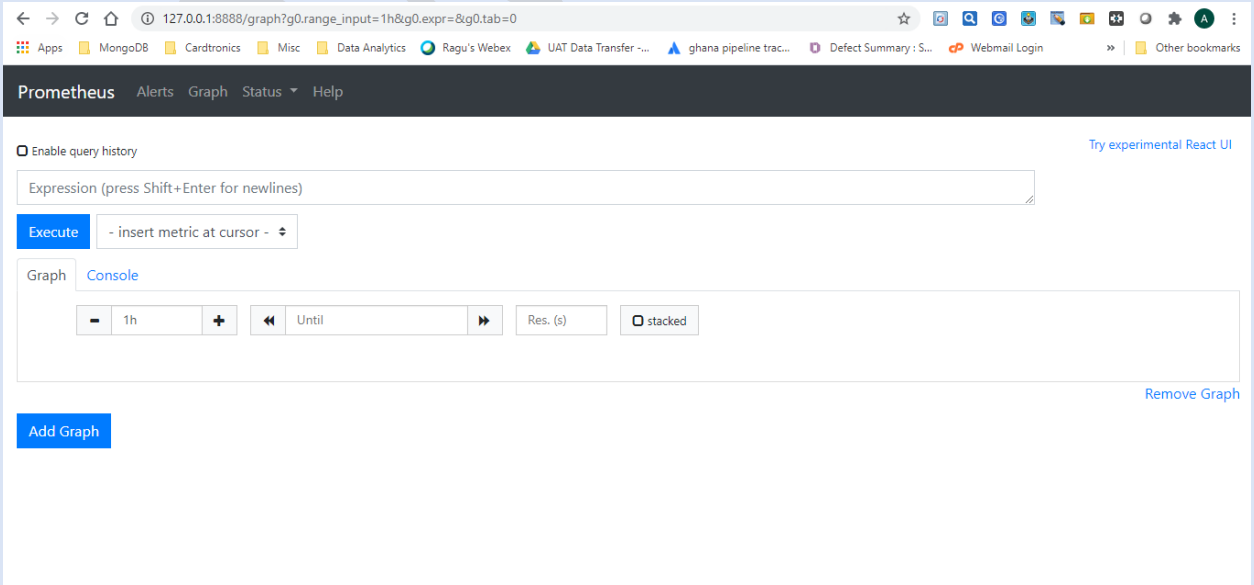
```
D:\EKS\D2>helm install stable/prometheus --namespace prometheus --set alertmanager.persistentVolume.storageClass="gp2" --set server.persistentVolume.storageClass="gp2"
NAME:   sullen-manatee
LAST DEPLOYED: Thu Jul  9 15:14:12 2020
NAMESPACE: prometheus
STATUS: DEPLOYED

RESOURCES:
==> v1/ConfigMap
NAME                                  DATA  AGE
sullen-manatee-prometheus-alertmanager  1     1s
sullen-manatee-prometheus-server        5     1s

==> v1/DaemonSet
NAME                                   DESIRED  CURRENT  READY  UP-TO-DATE  AVAILABLE  NODE SELECTOR  AGE
sullen-manatee-prometheus-node-exporter  7        7        0      7           0          <none>         1s

==> v1/Deployment
NAME                                  READY  UP-TO-DATE  AVAILABLE  AGE
sullen-manatee-kube-state-metrics       0/1    1           0          1s
sullen-manatee-prometheus-alertmanager  0/1    1           0          1s
sullen-manatee-prometheus-pushgateway   0/1    1           0          1s
sullen-manatee-prometheus-server        0/1    1           0          1s

==> v1/PersistentVolumeClaim
NAME                                  STATUS   VOLUME  CAPACITY  ACCESS MODES  STORAGECLASS  AGE
sullen-manatee-prometheus-alertmanager  Pending  gp2     1s
sullen-manatee-prometheus-server        Pending  gp2     1s

==> v1/Pod(related)
NAME                                                  READY  STATUS            RESTARTS  AGE
sullen-manatee-kube-state-metrics-5f6d4997d7-qgsmv     0/1    ContainerCreating  0         1s
sullen-manatee-prometheus-alertmanager-88f6655cf-njwj9  0/2    Pending           0         1s
sullen-manatee-prometheus-node-exporter-4mst5          0/1    ContainerCreating  0         1s
sullen-manatee-prometheus-node-exporter-4rdhp          0/1    ContainerCreating  0         1s
sullen-manatee-prometheus-node-exporter-cpgrm          0/1    ContainerCreating  0         1s
sullen-manatee-prometheus-node-exporter-gcdtj          0/1    ContainerCreating  0         1s
sullen-manatee-prometheus-node-exporter-jdncm          0/1    ContainerCreating  0         1s
sullen-manatee-prometheus-node-exporter-n6n4d          0/1    Pending           0         1s
sullen-manatee-prometheus-node-exporter-whc5f          0/1    ContainerCreating  0         1s
sullen-manatee-prometheus-pushgateway-97679b887-jmds6   0/1    ContainerCreating  0         1s
sullen-manatee-prometheus-server-64d5bdbdc4-wmkfg       0/2    Pending           0         1s
```

```
sullen-manatee-prometheus-node-exporter-whc5f           0/1    ContainerCreating  0         1s
sullen-manatee-prometheus-pushgateway-97679b887-jmds6   0/1    ContainerCreating  0         1s
sullen-manatee-prometheus-server-64d5bdbdc4-wmkfg       0/2    Pending            0         1s

==> v1/Service
NAME                                  TYPE       CLUSTER-IP      EXTERNAL-IP  PORT(S)   AGE
sullen-manatee-kube-state-metrics       ClusterIP  10.100.67.81    <none>       8080/TCP  1s
sullen-manatee-prometheus-alertmanager  ClusterIP  10.100.147.205  <none>       80/TCP    1s
sullen-manatee-prometheus-node-exporter ClusterIP  None            <none>       9100/TCP  1s
sullen-manatee-prometheus-pushgateway   ClusterIP  10.100.98.165   <none>       9091/TCP  1s
sullen-manatee-prometheus-server        ClusterIP  10.100.53.51    <none>       80/TCP    1s

==> v1/ServiceAccount
NAME                                  SECRETS  AGE
sullen-manatee-kube-state-metrics       1        1s
sullen-manatee-prometheus-alertmanager  1        1s
sullen-manatee-prometheus-node-exporter 1        1s
sullen-manatee-prometheus-pushgateway   1        1s
sullen-manatee-prometheus-server        1        1s

==> v1beta1/ClusterRole
NAME                                  AGE
sullen-manatee-prometheus-server        1s
sullen-manatee-kube-state-metrics       1s
sullen-manatee-prometheus-alertmanager  1s
sullen-manatee-prometheus-pushgateway   1s

==> v1beta1/ClusterRoleBinding
NAME                                  AGE
sullen-manatee-prometheus-server        1s
sullen-manatee-kube-state-metrics       1s
sullen-manatee-prometheus-alertmanager  1s
sullen-manatee-prometheus-pushgateway   1s


NOTES:
The Prometheus server can be accessed via port 80 on the following DNS name from within your cluster:
sullen-manatee-prometheus-server.prometheus.svc.cluster.local


Get the Prometheus server URL by running these commands in the same shell:
  export POD_NAME=$(kubectl get pods --namespace prometheus -l "app=prometheus,component=server" -o jsonpath="{.items[0].metadata.name}")
  kubectl --namespace prometheus port-forward $POD_NAME 9090
```

```
The Prometheus alertmanager can be accessed via port 80 on the following DNS name from within your cluster:
sullen-manatee-prometheus-alertmanager.prometheus.svc.cluster.local


Get the Alertmanager URL by running these commands in the same shell:
  export POD_NAME=$(kubectl get pods --namespace prometheus -l "app=prometheus,component=alertmanager" -o jsonpath="{.items[0].metadata.name}")
  kubectl --namespace prometheus port-forward $POD_NAME 9093
#################################################################################
######   WARNING: Pod Security Policy has been moved to a global property.  #####
######            use .Values.podSecurityPolicy.enabled with pod-based      #####
######            annotations                                               #####
######            (e.g. .Values.nodeExporter.podSecurityPolicy.annotations) #####
#################################################################################


The Prometheus PushGateway can be accessed via port 9091 on the following DNS name from within your cluster:
sullen-manatee-prometheus-pushgateway.prometheus.svc.cluster.local


Get the PushGateway URL by running these commands in the same shell:
  export POD_NAME=$(kubectl get pods --namespace prometheus -l "app=prometheus,component=pushgateway" -o jsonpath="{.items[0].metadata.name}")
  kubectl --namespace prometheus port-forward $POD_NAME 9091

For more information on running Prometheus, visit:
https://prometheus.io/
```

```
D:\EKS\D2>kubectl get pods -n prometheus
NAME                                                READY   STATUS    RESTARTS   AGE
sullen-manatee-kube-state-metrics-5f6d4997d7-qgsmv  1/1     Running   0          2m33s
sullen-manatee-prometheus-alertmanager-88f6655cf-njwj9  2/2     Running   0          2m33s
sullen-manatee-prometheus-node-exporter-4mst5       1/1     Running   0          2m33s
sullen-manatee-prometheus-node-exporter-4rdhp       1/1     Running   0          2m33s
sullen-manatee-prometheus-node-exporter-cpgrm       1/1     Running   0          2m33s
sullen-manatee-prometheus-node-exporter-gcdtj       1/1     Running   0          2m33s
sullen-manatee-prometheus-node-exporter-jdncm       1/1     Running   0          2m33s
sullen-manatee-prometheus-node-exporter-n6n4d       0/1     Pending   0          2m33s
sullen-manatee-prometheus-node-exporter-whc5f       1/1     Running   0          2m33s
sullen-manatee-prometheus-pushgateway-97679b887-jmds6   1/1     Running   0          2m33s
sullen-manatee-prometheus-server-64d5bdbdc4-wmkfg   2/2     Running   0          2m33s
```

```
D:\EKS\D2>kubectl get svc -n prometheus
NAME                                  TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)   AGE
sullen-manatee-kube-state-metrics     ClusterIP   10.100.67.81     <none>        8080/TCP  3m45s
sullen-manatee-prometheus-alertmanager  ClusterIP   10.100.147.205   <none>        80/TCP    3m45s
sullen-manatee-prometheus-node-exporter  ClusterIP   None             <none>        9100/TCP  3m45s
sullen-manatee-prometheus-pushgateway ClusterIP   10.100.98.165    <none>        9091/TCP  3m45s
sullen-manatee-prometheus-server      ClusterIP   10.100.53.51     <none>        80/TCP    3m45s
```



127.0.0.1:8888/graph?g0.range_input=1h&g0.expr=apiserver_request_count&g0.tab=1&g1.range_input=1h&g1.expr=&g1.ta...

Apps | MongoDB | Cardtronics | Misc | Data Analytics | Ragu's Webex | UAT Data Transfer -... | ghana pipeline trac... | Defect Summary : S... | Webmail Login | Other bookma

**Prometheus** Alerts Graph Status Help

apiserver_request_count

Load time: 109ms
Resolution: 14s
Total time series: 398

Execute | apiserver_request_count

Graph | Console

Moment

| Element | Value |
|---|---|
| apiserver_request_count{code="0",component="apiserver",instance="192.168.123.140:443",job="kubernetes-apiservers",resource="pods",scope="namespace",subresource="portforward",verb="CONNECT",version="v1"} | 1 |
| apiserver_request_count{code="0",component="apiserver",contentType="application/json",instance="192.168.123.140:443",job="kubernetes-apiservers",resource="configmaps",scope="namespace",verb="WATCH",version="v1"} | 2 |
| apiserver_request_count{code="0",component="apiserver",contentType="application/json",instance="192.168.123.140:443",job="kubernetes-apiservers",resource="events",scope="namespace",verb="WATCH",version="v1"} | 4 |
| apiserver_request_count{code="0",component="apiserver",contentType="application/json",instance="192.168.123.140:443",job="kubernetes-apiservers",resource="nodes",scope="cluster",verb="WATCH",version="v1"} | 18 |
| apiserver_request_count{code="0",component="apiserver",contentType="application/json",instance="192.168.123.140:443",job="kubernetes-apiservers",resource="persistentvolumeclaims",scope="cluster",verb="WATCH",version="v1"} | 10 |
| apiserver_request_count{code="0",component="apiserver",contentType="application/json",instance="192.168.123.140:443",job="kubernetes-apiservers",resource="persistentvolumes",scope="cluster",verb="WATCH",version="v1"} | 8 |
| apiserver_request_count{code="0",component="apiserver",contentType="application/json",instance="192.168.123.140:443",job="kubernetes-apiservers",resource="pods",scope="cluster",verb="WATCH",version="v1"} | 62 |
| apiserver_request_count{code="0",component="apiserver",contentType="application/json",instance="192.168.123.140:443",job="kubernetes-apiservers",resource="serviceaccounts",scope="cluster",verb="WATCH",version="v1"} | 16 |
| apiserver_request_count{code="0",component="apiserver",contentType="application/json",group="storage.k8s.io",instance="192.168.123.140:443",job="kubernetes- | 9 |

LinkedIn | Inbox (413) - a | My Drive - Go | 5thJuly2020 - | (3) LinkedIn | Elastic File Sys | Load Balancer | Prometheus Ti

127.0.0.1:8888/graph?g0.range_input=1h&g0.expr=apiserver_request_count&g0.tab=0&g1.range_input=1h&g1.expr=&g1.ta...

Apps | MongoDB | Cardtronics | Misc | Data Analytics | Ragu's Webex | UAT Data Transfer -... | ghana pipeline trac... | Defect Summary : S... | Webmail Login | Other bookmarks

**Prometheus** Alerts Graph Status Help

Execute | apiserver_request_count

Graph | Console

1h | Until | Res. (s) | stacked


Type here to search

3:23 PM
7/9/2020

127.0.0.1:8888/targets

Apps | MongoDB | Cardtronics | Misc | Data Analytics | Ragu's Webex | UAT Data Transfer -... | ghana pipeline trac... | Defect Summary : S... | Webmail Login

Prometheus Alerts Graph Status Help

# Targets

All Unhealthy

## kubernetes-apiservers (2/2 up) show less

| Endpoint | State | Labels | Last Scrape | Scrape Duration | Error |
|---|---|---|---|---|---|
| https://192.168.123.140:443/metrics | UP | instance="192.168.123.140:443" job="kubernetes-apiservers" | 50.718s ago | 58.7ms | |
| https://192.168.154.113:443/metrics | UP | instance="192.168.154.113:443" job="kubernetes-apiservers" | 18.371s ago | 97.63ms | |

## kubernetes-nodes (7/7 up) show less

| Endpoint | State | Labels | Last Scrape | Scrape Duration | Error |
|---|---|---|---|---|---|
| https://kubernetes.default.svc:443/api/v1/nodes/ip-192-168-14-197.ap-south-1.compute.internal/proxy/metrics | UP | alpha_eksctl_io_cluster_name="mncluster" alpha_eksctl_io_instance_id="i-0ade0546263c2781d" alpha_eksctl_io_nodegroup_name="ng-mixed" beta_kubernetes_io_arch="amd64" beta_kubernetes_io_instance_type="t3.small" | 33.91s ago | 12.19ms | |

## prometheus (1/1 up) show less

| Endpoint | State | Labels | Last Scrape | Scrape Duration | Error |
|---|---|---|---|---|---|
| http://localhost:9090/metrics | UP | instance="localhost:9090" job="prometheus" | 30.057s ago | 6.273ms | |

## prometheus-pushgateway (1/1 up) show less

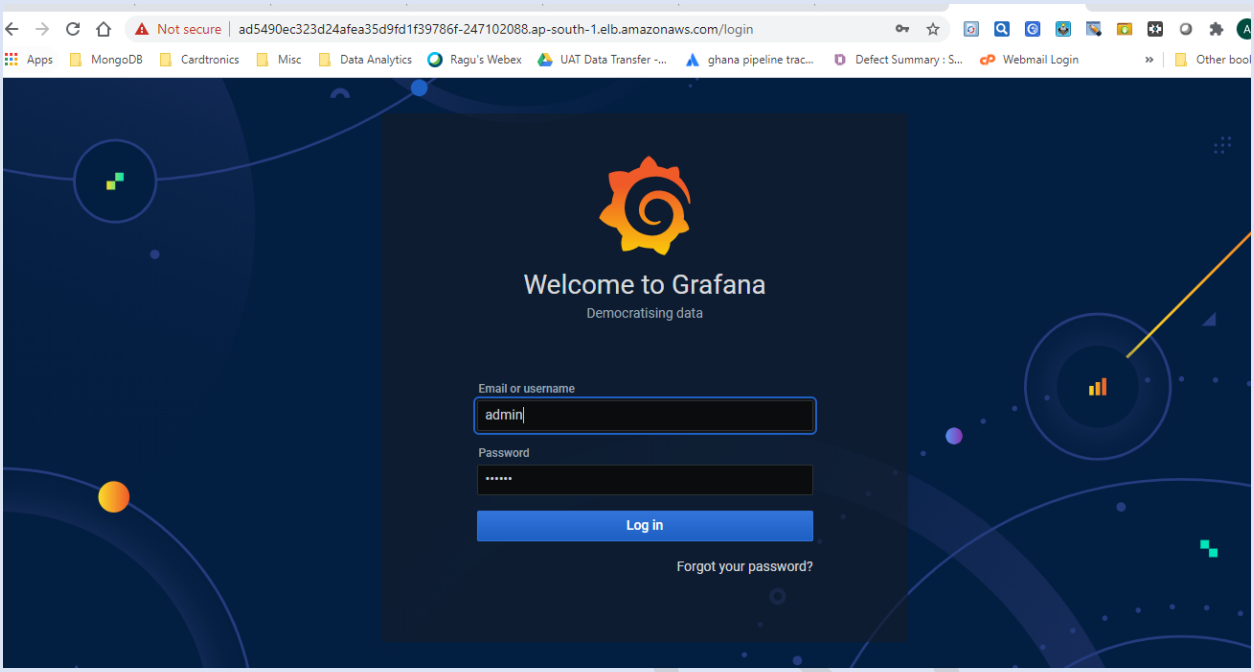
| Endpoint | State | Labels | Last Scrape | Scrape Duration | Error |
|---|---|---|---|---|---|
| http://sullen-manatee-prometheus-pushgateway.prometheus.svc:9091/metrics | UP | instance="sullen-manatee-prometheus-pushgateway.prometheus.svc:9091" job="prometheus-pushgateway" | 36.592s ago | 2.861ms | |

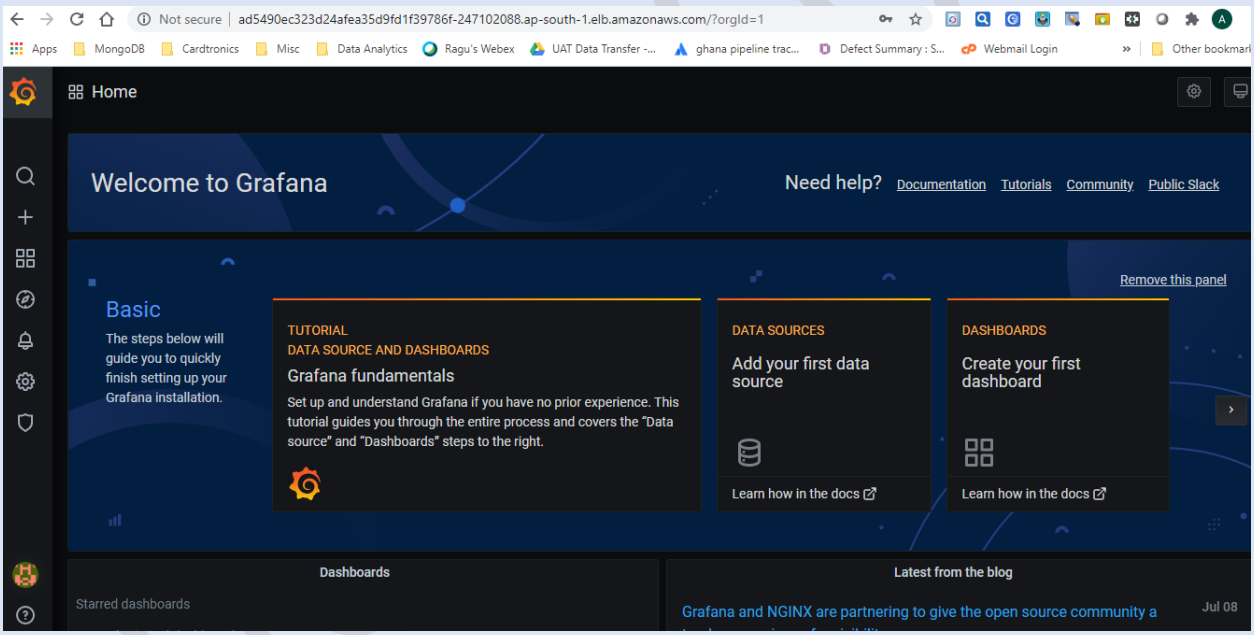
**Install Grafana.**

```
D:\EKS\D2>helm install stable/grafana --namespace grafana --set persistent.storageClassName="gp2" --set adminPassword=bingos --set service.type=LoadBalancer
NAME:   smelly-hummingbird
LAST DEPLOYED: Thu Jul  9 15:34:32 2020
NAMESPACE: grafana
STATUS: DEPLOYED

RESOURCES:
==> v1/ClusterRole
NAME                                   AGE
smelly-hummingbird-grafana-clusterrole  0s

==> v1/ClusterRoleBinding
NAME                                           AGE
smelly-hummingbird-grafana-clusterrolebinding  0s

==> v1/ConfigMap
NAME                            DATA  AGE
smelly-hummingbird-grafana      1     0s
smelly-hummingbird-grafana-test 1     0s

==> v1/Deployment
NAME                      READY  UP-TO-DATE  AVAILABLE  AGE
smelly-hummingbird-grafana  0/1    1           0          0s

==> v1/Pod(related)
NAME                                       READY  STATUS             RESTARTS  AGE
smelly-hummingbird-grafana-697cbdf855-v69xx  0/1    ContainerCreating  0         0s

==> v1/Role
NAME                            AGE
smelly-hummingbird-grafana-test  0s

==> v1/RoleBinding
NAME                            AGE
smelly-hummingbird-grafana-test  0s

==> v1/Secret
NAME                      TYPE    DATA  AGE
smelly-hummingbird-grafana  Opaque  3     0s

==> v1/Service
NAME                      TYPE          CLUSTER-IP      EXTERNAL-IP  PORT(S)       AGE
smelly-hummingbird-grafana  LoadBalancer  10.100.158.200  <pending>    80:31570/TCP  0s
```

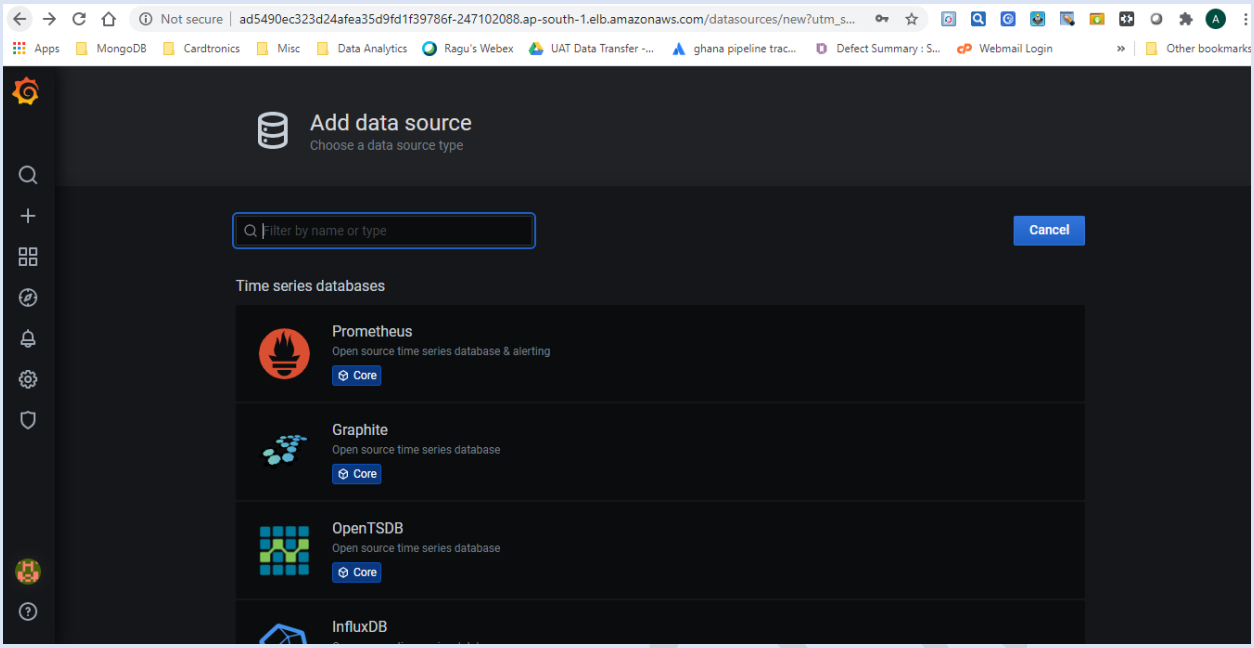
```
==> v1/ServiceAccount
NAME                            SECRETS  AGE
smelly-hummingbird-grafana      1        0s
smelly-hummingbird-grafana-test 1        0s

==> v1beta1/PodSecurityPolicy
NAME                            PRIV   CAPS      SELINUX   RUNASUSER  FSGROUP   SUPGROUP  READONLYROOTFS  VOLUMES
smelly-hummingbird-grafana      false  RunAsAny  RunAsAny  RunAsAny   RunAsAny  false     configMap,emptyDir,projected,secret,downwardAPI,persistentVolumeClaim
smelly-hummingbird-grafana-test false  RunAsAny  RunAsAny  RunAsAny   RunAsAny  false     configMap,downwardAPI,emptyDir,projected,secret

==> v1beta1/Role
NAME                      AGE
smelly-hummingbird-grafana  0s

==> v1beta1/RoleBinding
NAME                      AGE
smelly-hummingbird-grafana  0s


NOTES:
1. Get your 'admin' user password by running:

   kubectl get secret --namespace grafana smelly-hummingbird-grafana -o jsonpath="{.data.admin-password}" | base64 --decode ; echo

2. The Grafana server can be accessed via port 80 on the following DNS name from within your cluster:

   smelly-hummingbird-grafana.grafana.svc.cluster.local

   Get the Grafana URL to visit by running these commands in the same shell:
NOTE: It may take a few minutes for the LoadBalancer IP to be available.
        You can watch the status of by running 'kubectl get svc --namespace grafana -w smelly-hummingbird-grafana'
     export SERVICE_IP=$(kubectl get svc --namespace grafana smelly-hummingbird-grafana -o jsonpath='{.status.loadBalancer.ingress[0].ip}')
     http://$SERVICE_IP:80

3. Login with the password from step 1 and the username: admin
#################################################################################
######   WARNING: Persistence is disabled!!! You will lose your data when   #####
######            the Grafana pod is terminated.                             #####
#################################################################################
```

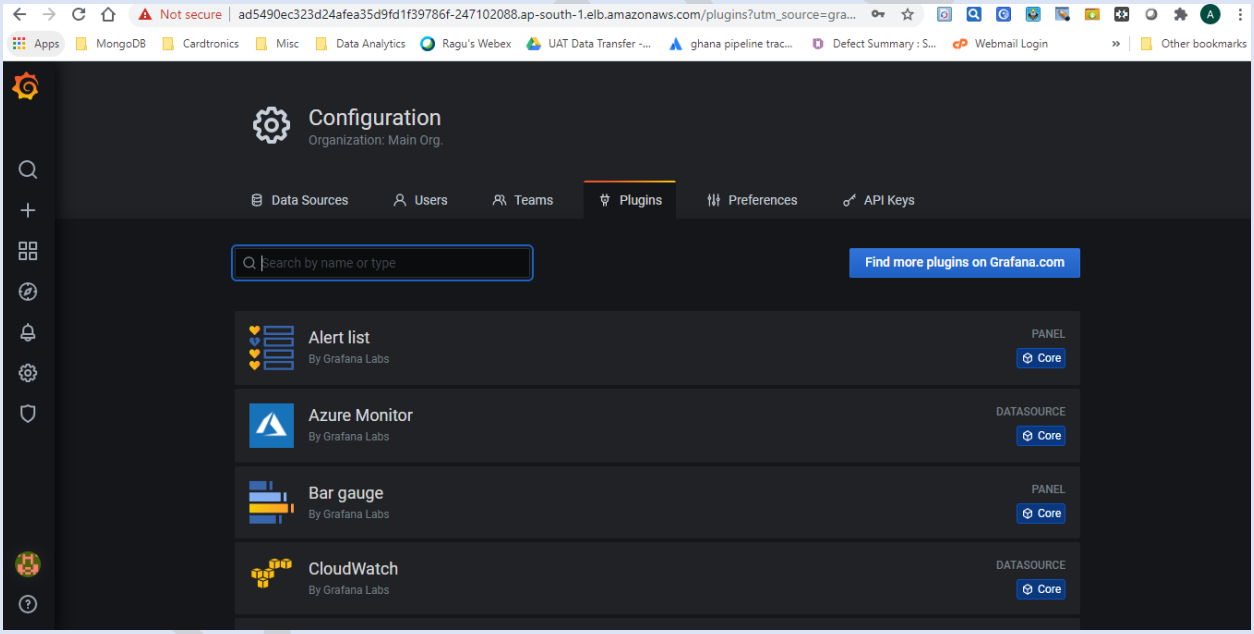
```
D:\EKS\D2>kubectl get all -n grafana
NAME                                       READY   STATUS    RESTARTS   AGE
pod/smelly-hummingbird-grafana-697cbdf855-v69xx   1/1     Running   0          5m36s

NAME                                 TYPE           CLUSTER-IP       EXTERNAL-IP                                                              PORT(S)        AGE
service/smelly-hummingbird-grafana   LoadBalancer   10.100.158.200   ad5490ec323d24afea35d9fd1f39786f-247102088.ap-south-1.elb.amazonaws.com   80:31570/TCP   5m36s

NAME                                   READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/smelly-hummingbird-grafana   1/1     1            1           5m36s

NAME                                              DESIRED   CURRENT   READY   AGE
replicaset.apps/smelly-hummingbird-grafana-697cbdf855   1         1         1       5m36s
```

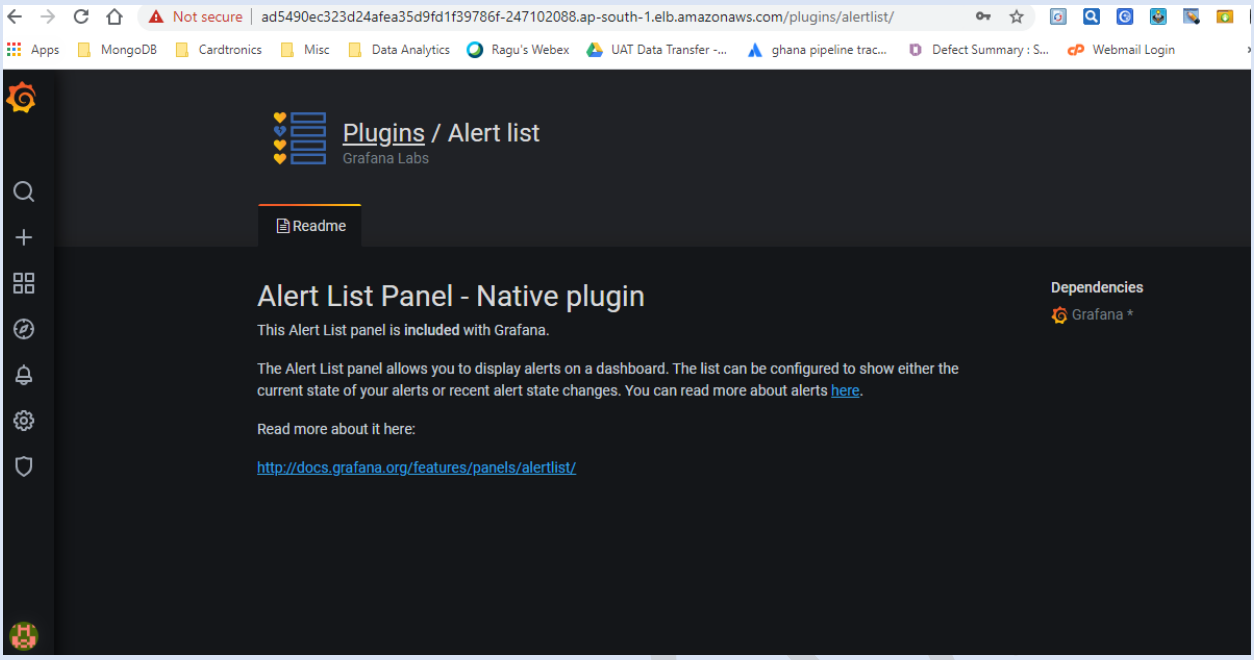



Select Prometheus Data Source


Not secure | ad5490ec323d24afea35d9fd1f39786f-247102088.ap-south-1.elb.amazonaws.com/datasources/new?utm_s...
Apps
MongoDB
Cardtronics
Misc
Data Analytics
Ragu's Webex
UAT Data Transfer -...
ghana pipeline trac...
Defect Summary : S...
Webmail Login
Other bookmarks
Add data source
Choose a data source type
Filter by name or type
Cancel
Time series databases
Prometheus
Open source time series database & alerting
Core
Graphite
Open source time series database
Core
OpenTSDB
Open source time series database
Core
InfluxDB


Not secure | ad5490ec323d24afea35d9fd1f39786f-247102088.ap-south-1.elb.amazonaws.com/plugins?utm_source=gra...
Apps
MongoDB
Cardtronics
Misc
Data Analytics
Ragu's Webex
UAT Data Transfer -...
ghana pipeline trac...
Defect Summary : S...
Webmail Login
Other bookmarks
Configuration
Organization: Main Org.
Data Sources
Users
Teams
Plugins
Preferences
API Keys
Search by name or type
Find more plugins on Grafana.com
Alert list
By Grafana Labs
PANEL
Core
Azure Monitor
By Grafana Labs
DATASOURCE
Core
Bar gauge
By Grafana Labs
PANEL
Core
CloudWatch
By Grafana Labs
DATASOURCE
Core



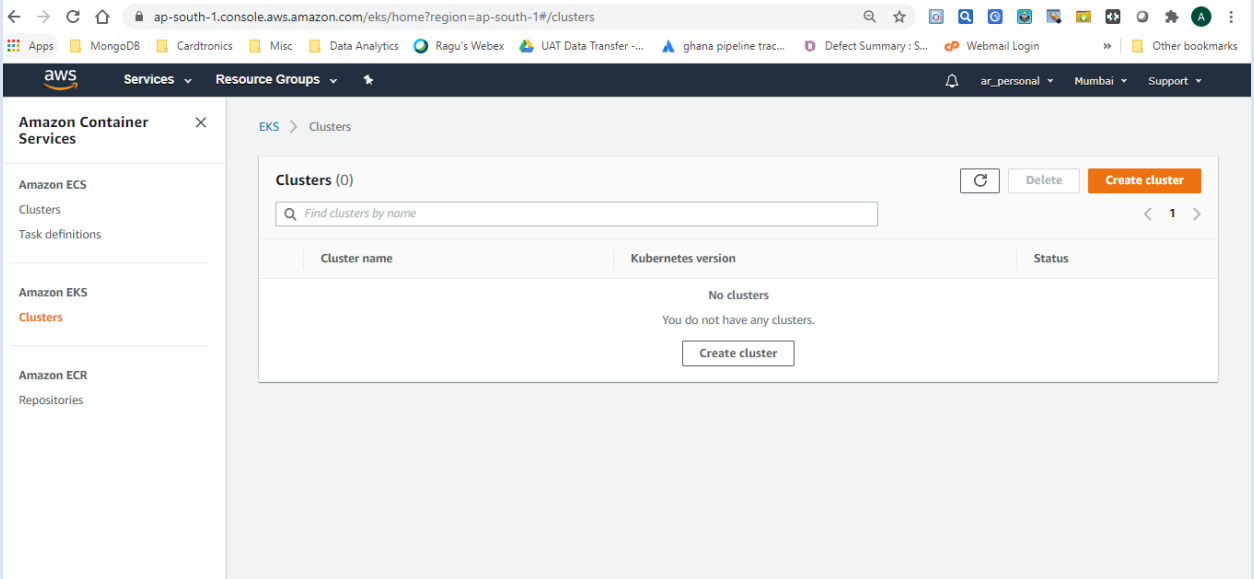

---

To delete entire EKS Cluster using below command

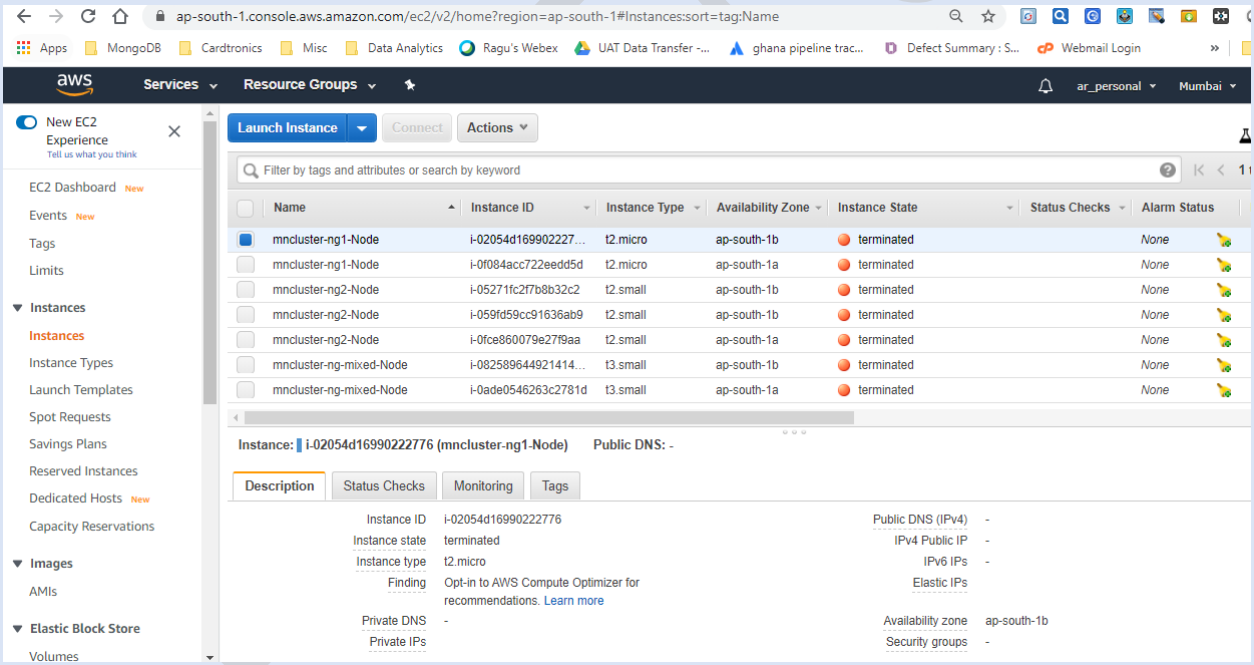
```
D:\EKS\D2>eksctl delete cluster -f cluster.yml
[ℹ]  eksctl version 0.21.0
[ℹ]  using region ap-south-1
[ℹ]  deleting EKS cluster "mncluster"
[ℹ]  either account is not authorized to use Fargate or region ap-south-1 is not supported. Ignoring error
[ℹ]  cleaning up LoadBalancer services
[ℹ]  2 sequential tasks: { 3 parallel sub-tasks: { delete nodegroup "ng2", delete nodegroup "ng1", delete nodegroup "ng-mixed" }, delete cluster control plane "mncluste
r" [async] }
[ℹ]  will delete stack "eksctl-mncluster-nodegroup-ng2"
[ℹ]  waiting for stack "eksctl-mncluster-nodegroup-ng2" to get deleted
[ℹ]  will delete stack "eksctl-mncluster-nodegroup-ng-mixed"
[ℹ]  waiting for stack "eksctl-mncluster-nodegroup-ng1"
o get deleted
 tack "eksctl-mnclusstack "eksctl-mncluster-nodegroup-ng1"
[ℹ]  waiting for stack "eksctl-mncluster-nodegroup-ng1" to get deleted
[ℹ]  will delete stack "eksctl-mncluster-cluster"
[ℹ]  all cluster resources were deleted
```

```
D:\EKS\D2>
D:\EKS\D2>eksctl get cluster
No clusters found

D:\EKS\D2>
```

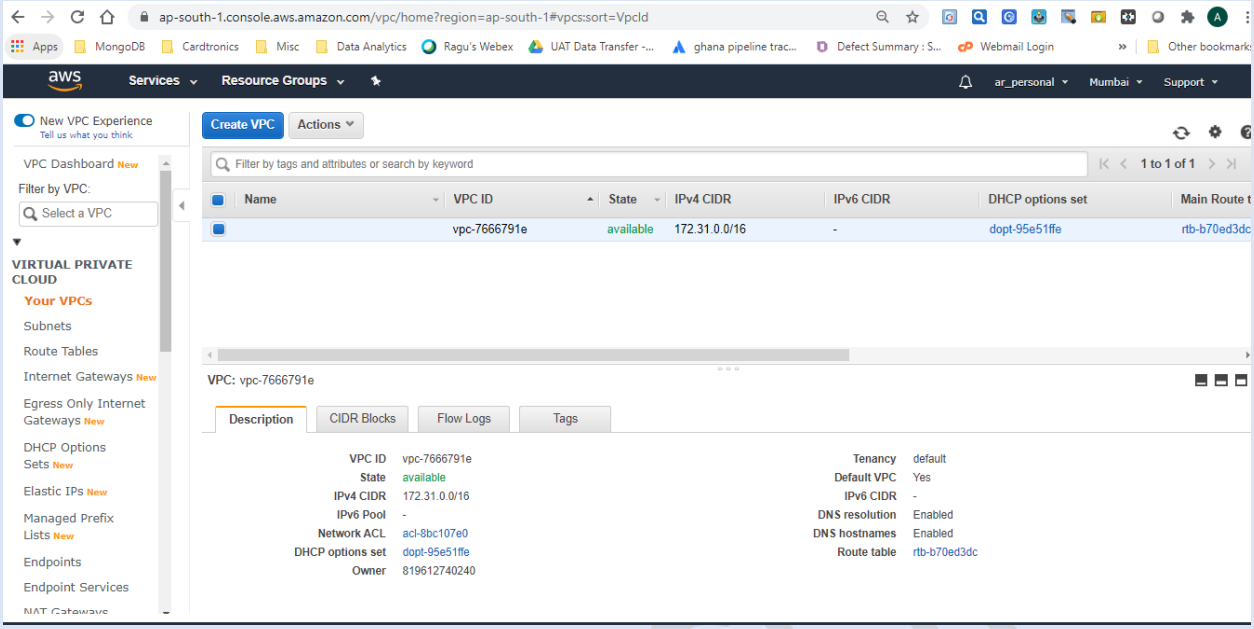
Verify deletion of cluster in AWS Console



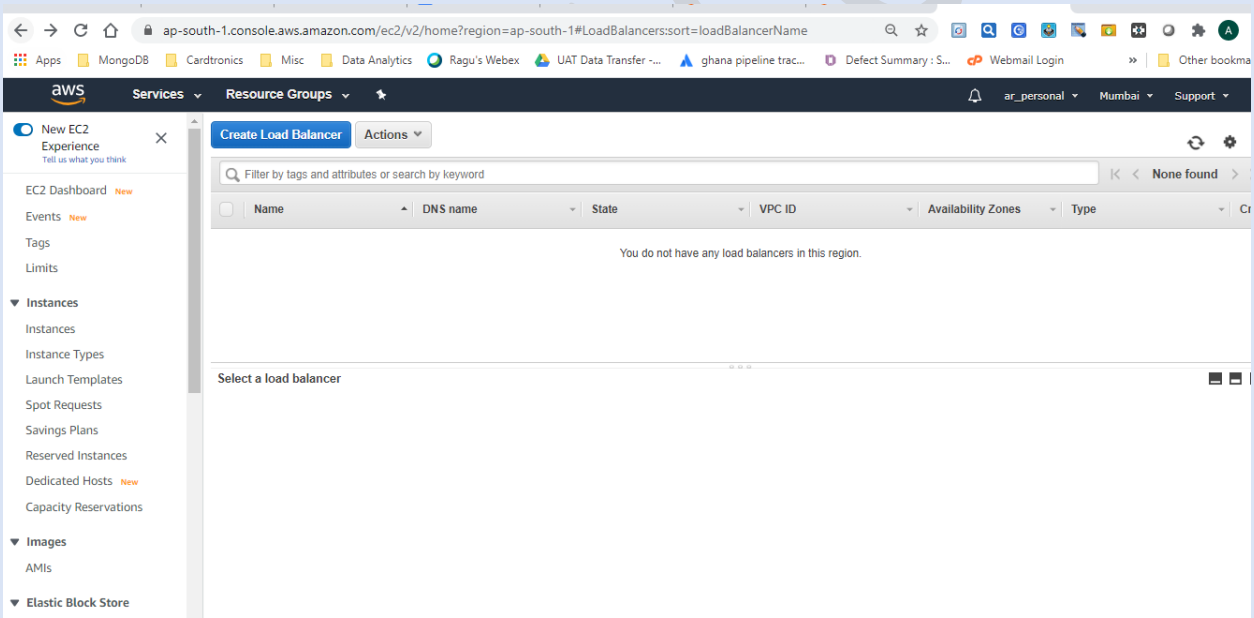

Verify that all EC2 instances are terminated.




Verify that the VPC which was created by EKS Cluster also gets removed
Below is the default VPC which is created by AWS in a region.




Load Balancer also gets removed as part of cluster deletion.




We can remove EFS volume finally which is not required.

```
C:\Program Files\Kubernetes\Minikube>eksctl get cluster
No clusters found
```

## AWS Fargate

Fargate is a server-less architecture & it integrates with EKS.

It's a black box which dynamically manages slave nodes without worrying about capacity such as CPU, RAM etc.

It launches slaves at run time & manages internally. We can't see those slaves in the AWS console though.

```
C:\Program Files\Kubernetes\Minikube>eksctl create cluster -f D:\EKS\D2\fcluster.yml
[ℹ]  eksctl version 0.21.0
[ℹ]  using region ap-southeast-1
[ℹ]  setting availability zones to [ap-southeast-1b ap-southeast-1c ap-southeast-1a]
[ℹ]  subnets for ap-southeast-1b - public:192.168.0.0/19 private:192.168.96.0/19
[ℹ]  subnets for ap-southeast-1c - public:192.168.32.0/19 private:192.168.128.0/19
[ℹ]  subnets for ap-southeast-1a - public:192.168.64.0/19 private:192.168.160.0/19
[ℹ]  using Kubernetes version 1.16
[ℹ]  creating EKS cluster "far-lwcluster" in "ap-southeast-1" region with Fargate profile
[ℹ]  will create a CloudFormation stack for cluster itself and 0 nodegroup stack(s)
[ℹ]  will create a CloudFormation stack for cluster itself and 0 managed nodegroup stack(s)
[ℹ]  if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=ap-southeast-1 --cluster=far-lwcluster'
[ℹ]  CloudWatch logging will not be enabled for cluster "far-lwcluster" in "ap-southeast-1"
[ℹ]  you can enable it with 'eksctl utils update-cluster-logging --region=ap-southeast-1 --cluster=far-lwcluster'
[ℹ]  Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "far-lwcluster" in "ap-southeast-1"
[ℹ]  2 sequential tasks: { create cluster control plane "far-lwcluster", no tasks }
[ℹ]  building cluster stack "eksctl-far-lwcluster-cluster"
[ℹ]  deploying stack "eksctl-far-lwcluster-cluster"
[ℹ]  waiting for the control plane availability...
[!]  unable to write kubeconfig , please retry with 'eksctl utils write-kubeconfig -n far-lwcluster': unable to read existing kubeconfig file "C:\\Users\\arvind
.ramugade/.kube/config": error loading config file "C:\Users\arvind.ramugade/.kube/config": read C:\Users\arvind.ramugade/.kube/config: The process cannot acces
s the file because another process has locked a portion of the file.
[ℹ]  no tasks
[ℹ]  all EKS cluster resources for "far-lwcluster" have been created
[ℹ]  creating Fargate profile "fargate-default" on EKS cluster "far-lwcluster"
[ℹ]  created Fargate profile "fargate-default" on EKS cluster "far-lwcluster"
[ℹ]  "coredns" is now schedulable onto Fargate
[ℹ]  "coredns" is now scheduled onto Fargate
[ℹ]  "coredns" pods are now scheduled onto Fargate
[ℹ]  EKS cluster "far-lwcluster" in "ap-southeast-1" region is ready

C:\Program Files\Kubernetes\Minikube>
```

```
C:\Program Files\Kubernetes\Minikube>eksctl get cluster --region ap-southeast-1
NAME            REGION
far-lwcluster   ap-southeast-1

C:\Program Files\Kubernetes\Minikube>aws eks --region ap-southeast-1 update-kubeconfig --name far-lwcluster
Added new context arn:aws:eks:ap-southeast-1:819612740240:cluster/far-lwcluster to C:\Users\arvind.ramugade\.kube\config

C:\Program Files\Kubernetes\Minikube>
```

```
C:\Program Files\Kubernetes\Minikube>kubectl config view
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: DATA+OMITTED
    server: https://B281A5F495624F50E70A2F5BB0BBBBF5.yl4.ap-south-1.eks.amazonaws.com
  name: arn:aws:eks:ap-south-1:819612740240:cluster/lwcluster
- cluster:
    certificate-authority-data: DATA+OMITTED
    server: https://F1D498ED792166C8A8DC0775B26535A2.sk1.ap-southeast-1.eks.amazonaws.com
  name: arn:aws:eks:ap-southeast-1:819612740240:cluster/far-lwcluster
- cluster:
    certificate-authority: C:\Users\arvind.ramugade\.minikube\ca.crt
    server: https://192.168.99.100:8443
  name: minikube
contexts:
- context:
    cluster: arn:aws:eks:ap-south-1:819612740240:cluster/lwcluster
    user: arn:aws:eks:ap-south-1:819612740240:cluster/lwcluster
  name: arn:aws:eks:ap-south-1:819612740240:cluster/lwcluster
- context:
    cluster: arn:aws:eks:ap-southeast-1:819612740240:cluster/far-lwcluster
    user: arn:aws:eks:ap-southeast-1:819612740240:cluster/far-lwcluster
  name: arn:aws:eks:ap-southeast-1:819612740240:cluster/far-lwcluster
- context:
    cluster: minikube
    user: minikube
  name: minikube
current-context: arn:aws:eks:ap-southeast-1:819612740240:cluster/far-lwcluster
kind: Config
preferences: {}
users:
- name: arn:aws:eks:ap-south-1:819612740240:cluster/lwcluster
  user:
    exec:
      apiVersion: client.authentication.k8s.io/v1alpha1
      args:
      - --region
      - ap-south-1
      - eks
      - get-token
      - --cluster-name
      - lwcluster
      command: aws
      env: null
- name: arn:aws:eks:ap-southeast-1:819612740240:cluster/far-lwcluster
  user:
    exec:
      apiVersion: client.authentication.k8s.io/v1alpha1
      args:
      - --region
      - ap-southeast-1
      - eks
      - get-token
      - --cluster-name
      - far-lwcluster
      command: aws
      env: null
- name: minikube
  user:
    client-certificate: C:\Users\arvind.ramugade\.minikube\profiles\minikube\client.crt
    client-key: C:\Users\arvind.ramugade\.minikube\profiles\minikube\client.key
```

```
C:\Program Files\Kubernetes\Minikube>kubectl get nodes
NAME                                                STATUS   ROLES    AGE   VERSION
fargate-ip-192-168-118-123.ap-southeast-1.compute.internal   Ready    <none>   11m   v1.16.8-eks-e16311
fargate-ip-192-168-189-110.ap-southeast-1.compute.internal   Ready    <none>   11m   v1.16.8-eks-e16311
```

To delete Fargate cluster use below command.

```
C:\Program Files\Kubernetes\Minikube>eksctl delete cluster -f D:\EKS\D2\fcluster.yml
[ℹ]  eksctl version 0.21.0
[ℹ]  using region ap-southeast-1
[ℹ]  deleting EKS cluster "far-lwcluster"
[ℹ]  deleting Fargate profile "fargate-default"
[ℹ]  deleted Fargate profile "fargate-default"
[ℹ]  deleted 1 Fargate profile(s)
[ℹ]  cleaning up LoadBalancer services
[ℹ]  1 task: { delete cluster control plane "far-lwcluster" [async] }
[ℹ]  will delete stack "eksctl-far-lwcluster-cluster"
[ℹ]  all cluster resources were deleted
```