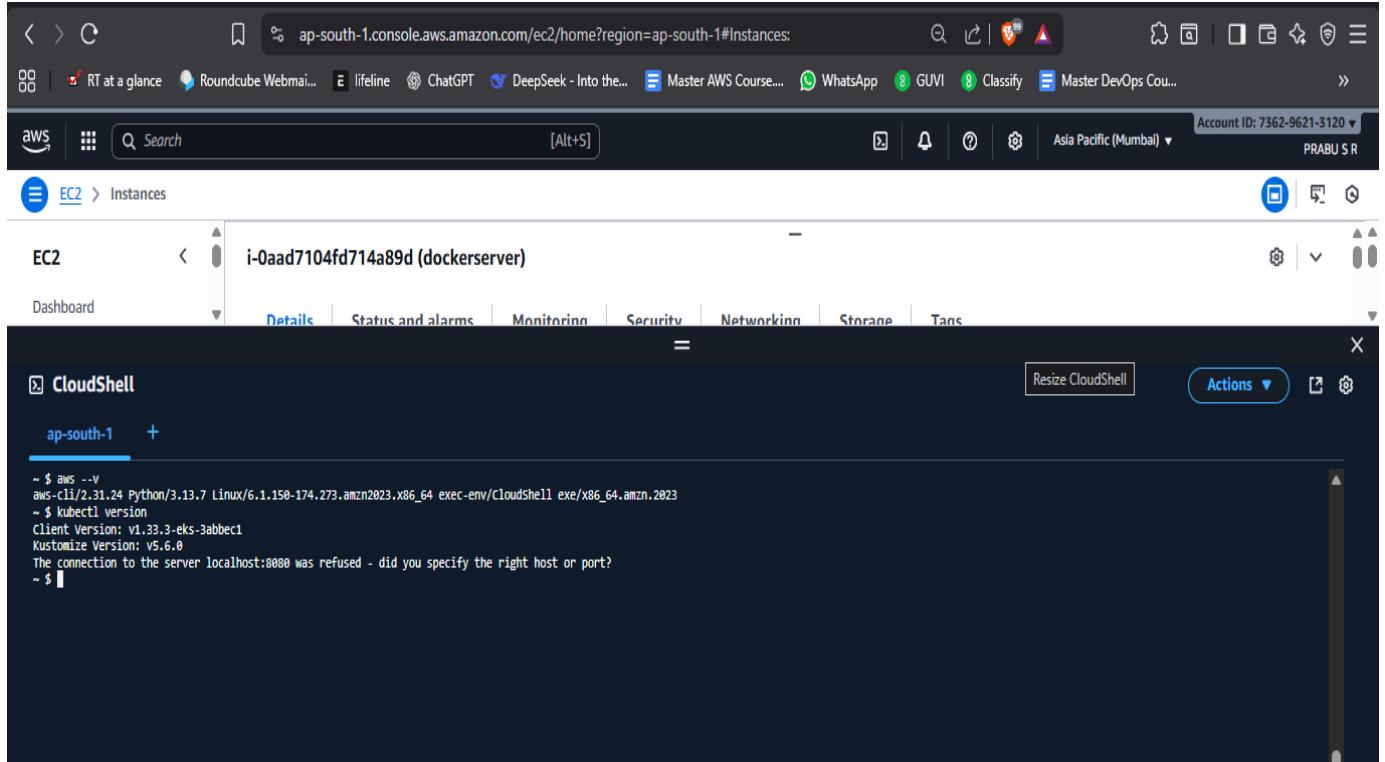


Kubernetes Task-2

Task Description:

Create the K8s EKS,further you have to do the deployment of the Nginx application and access the application outside the cluster.

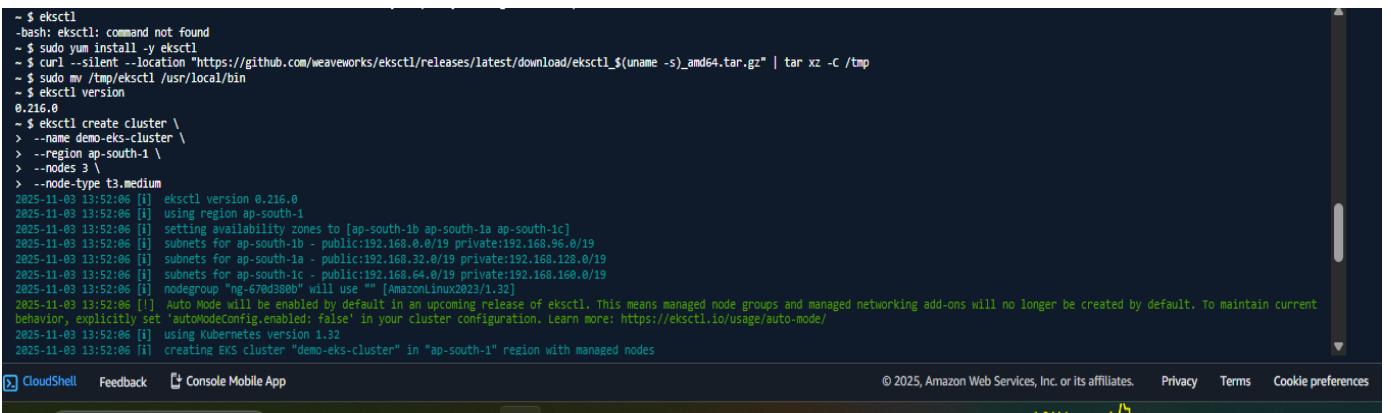
Kubernetes version check



A screenshot of the AWS CloudShell interface within the AWS Management Console. The URL in the address bar is `ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#Instances`. The CloudShell tab is active, showing a terminal window with the following command history:

```
- $ aws --v
aws-cli/2.31.24 Python/3.13.7 Linux/6.1.158-174.273.amzn2023.x86_64 exec-env/CloudShell exe/x86_64.amzn.2023
~ $ kubectl version
Client Version: v1.33.3-eks-3abbec1
Kustomize Version: v5.6.0
The connection to the server localhost:8080 was refused - did you specify the right host or port?
~ $
```

eksctl version check and if not found , download manually and move to bin location and create cluster with 3 nodes, here t3.medium (free aws: t3.small only supported) i have given t3.small



A screenshot of the AWS CloudShell interface within the AWS Management Console. The terminal window shows the following command history for creating an EKS cluster:

```
- $ eksctl
-bash: eksctl: command not found
~ $ sudo yum install -y eksctl
~ $ curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_$(uname -s)_amd64.tar.gz" | tar xz -c /tmp
~ $ sudo mv /tmp/eksctl /usr/local/bin
0.216.0
~ $ eksctl create cluster \
> --name demo-eks-cluster \
> --region ap-south-1 \
> --nodes 3 \
> --node-type t3.medium
2025-11-03 13:52:06 [i] eksctl version 0.216.0
2025-11-03 13:52:06 [i] using region ap-south-1
2025-11-03 13:52:06 [i] setting availability zones to [ap-south-1b ap-south-1a ap-south-1c]
2025-11-03 13:52:06 [i] subnets for ap-south-1b - public:192.168.0.0/19 private:192.168.96.0/19
2025-11-03 13:52:06 [i] subnets for ap-south-1a - public:192.168.32.0/19 private:192.168.128.0/19
2025-11-03 13:52:06 [i] subnets for ap-south-1c - public:192.168.64.0/19 private:192.168.160.0/19
2025-11-03 13:52:06 [i] nodegroup "ng-670d380b" will use "" [AmazonLinux2023/1.32]
2025-11-03 13:52:06 [i] Auto Node will be enabled by default in an upcoming release of eksctl. This means managed node groups and managed networking add-ons will no longer be created by default. To maintain current behavior, explicitly set 'autoNodeConfig.enabled: false' in your cluster configuration. Learn more: https://eksctl.io/usage/auto-node/
2025-11-03 13:52:06 [i] using Kubernetes version 1.32
2025-11-03 13:52:06 [i] creating EKS cluster "demo-eks-cluster" in "ap-south-1" region with managed nodes
```

CloudShell

ap-south-1 +

```
2025-11-03 13:52:06 [i] default addons coredns, metrics-server, vpc-cni, kube-proxy were not specified, will install them as EKS addons
2025-11-03 13:52:06 [i]
2 sequential tasks: { create cluster control plane "demo-eks-cluster",
  2 sequential sub-tasks: {
    2 sequential sub-tasks: {
      1 task: { create addons },
      wait for control plane to become ready,
    },
    create managed nodegroup "ng-670d380b",
  }
}
2025-11-03 13:52:06 [i] building cluster stack "eksctl-demo-eks-cluster-cluster"
2025-11-03 13:52:06 [i] deploying stack "eksctl-demo-eks-cluster-cluster"
2025-11-03 13:52:36 [i] waiting for CloudFormation stack "eksctl-demo-eks-cluster-cluster"
2025-11-03 13:53:06 [i] waiting for CloudFormation stack "eksctl-demo-eks-cluster-cluster"
2025-11-03 13:54:06 [i] waiting for CloudFormation stack "eksctl-demo-eks-cluster-cluster"
2025-11-03 13:55:06 [i] waiting for CloudFormation stack "eksctl-demo-eks-cluster-cluster"
2025-11-03 13:56:06 [i] waiting for CloudFormation stack "eksctl-demo-eks-cluster-cluster"
2025-11-03 13:57:07 [i] waiting for CloudFormation stack "eksctl-demo-eks-cluster-cluster"
2025-11-03 13:58:07 [i] waiting for CloudFormation stack "eksctl-demo-eks-cluster-cluster"
2025-11-03 13:59:07 [i] waiting for CloudFormation stack "eksctl-demo-eks-cluster-cluster"
2025-11-03 14:00:07 [i] waiting for CloudFormation stack "eksctl-demo-eks-cluster-cluster"
```

CloudShell

ap-south-1 +

```
Actions ▾ X
```

```
2025-11-03 14:44:48 [i] no tasks
2025-11-03 14:44:48 [v] all EKS cluster resources for "demo-eks-cluster" have been created
2025-11-03 14:44:49 [i] nodegroup "ng-bc9e0219" has 3 node(s)
2025-11-03 14:44:49 [i] node "ip-192-168-19-38.ap-south-1.compute.internal" is ready
2025-11-03 14:44:49 [i] node "ip-192-168-48-43.ap-south-1.compute.internal" is ready
2025-11-03 14:44:49 [i] node "ip-192-168-87-82.ap-south-1.compute.internal" is ready
2025-11-03 14:44:49 [i] waiting for at least 3 node(s) to become ready in "ng-bc9e0219"
2025-11-03 14:44:49 [i] nodegroup "ng-bc9e0219" has 3 node(s)
2025-11-03 14:44:49 [i] node "ip-192-168-19-38.ap-south-1.compute.internal" is ready
2025-11-03 14:44:49 [i] node "ip-192-168-48-43.ap-south-1.compute.internal" is ready
2025-11-03 14:44:49 [i] node "ip-192-168-87-82.ap-south-1.compute.internal" is ready
2025-11-03 14:44:49 [v] created 1 managed nodegroup(s) in cluster "demo-eks-cluster"
2025-11-03 14:44:49 [i] creating addon: metrics-server
2025-11-03 14:44:49 [i] successfully created addon: metrics-server
2025-11-03 14:44:50 [i] kubectl command should work with "/home/cloudshell-user/.kube/config", try 'kubectl get nodes'
2025-11-03 14:44:50 [v] EKS cluster "demo-eks-cluster" in "ap-south-1" region is ready
~ $
```

CloudShell Feedback Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Screenshot of the AWS EC2 Instances page showing three running t3.small instances in the ap-south-1b availability zone. The instance IDs are i-0b9b9aecc99a77923, i-0c1416699ac12212f, and i-001dee657788f01a6.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
demo-eks-cluster-ng-bc9e0219-N...	i-0b9b9aecc99a77923	Running	t3.small	Initializing	View alarms +	ap-south-1b
demo-eks-cluster-ng-bc9e0219-N...	i-0c1416699ac12212f	Running	t3.small	Initializing	View alarms +	ap-south-1a
demo-eks-cluster-ng-bc9e0219-N...	i-001dee657788f01a6	Running	t3.small	Initializing	View alarms +	ap-south-1c

create nginx-deployment.yaml file

Screenshot of the AWS CloudShell interface on the ap-south-1 region. A new terminal window is open, and the user is creating an nginx-deployment.yaml file using nano editor.

```
GNU nano 8.3
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  selector:
    matchLabels:
      app: nginx
  replicas: 2 # tells deployment to run 2 pods matching the template
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:1.14.2
          ports:
            - containerPort: 80
```

The user successfully creates the deployment:

```
2025-11-03 14:44:49 [i] successfully created addon: metrics-server
2025-11-03 14:44:50 [i] kubectl command should work with "/home/cloudshell-user/.kube/config", try 'kubectl get nodes'
2025-11-03 14:44:50 [✓] EKS cluster "demo-eks-cluster" in "ap-south-1" region is ready
~ $ nano nginx-deployment.yaml
~ $ kubectl apply -f nginx-deployment.yaml
deployment.apps/nginx-deployment created
~ $
```

```
deployment.apps/nginx-deployment created
~ $ kubectl get pods
NAME                      READY   STATUS    RESTARTS   AGE
nginx-deployment-647677fc66-bn9gv   1/1     Running   0          20s
nginx-deployment-647677fc66-wpxbg   1/1     Running   0          20s
~ $
```

create nginx-service.yaml

```
~ $ kubectl apply -f nginx-service.yaml
error: error parsing nginx-service.yaml: error converting YAML to JSON: yaml: line 12: found a tab character that violates indentation
~ $ nano nginx-service.yaml
~ $ kubectl apply -f nginx-service.yaml
service/nginx-service created
~ $
```

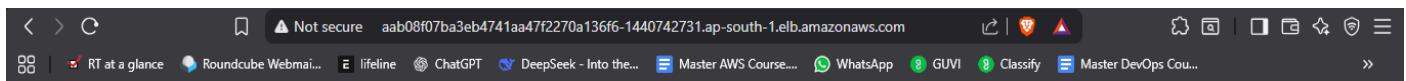
CloudShell Feedback Console Mobile App

© 2025, Amazon Web Services

run the service url(external ip)

```
~ $ kubectl get svc nginx-service --watch
NAME      TYPE        CLUSTER-IP   EXTERNAL-IP
nginx-service  LoadBalancer  10.100.199.108  aab08f07ba3eb4741aa47f2270a136f6-1440742731.ap-south-1.elb.amazonaws.com
PORT(S)    AGE
80:30911/TCP  58s
```

paste in browser and see nginx webpage



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

common cmds to check logs and datas inside clusters

```
^C~ $ kubectl get pods -o wide
NAME          READY   STATUS    RESTARTS   AGE   IP           NODE   NOMINATED-NODE   READINESS   GATES
nginx-deployment-647677fc66-bn9gv  1/1    Running   0          6m21s  192.168.74.59  ip-192-168-87-82.ap-south-1.compute.internal  <none>        <none>
nginx-deployment-647677fc66-wpxbg  1/1    Running   0          6m21s  192.168.20.211 ip-192-168-19-38.ap-south-1.compute.internal  <none>        <none>
~ $
```

```
~ $ kubectl get all
NAME          READY   STATUS    RESTARTS   AGE
pod/nginx-deployment-647677fc66-bn9gv  1/1    Running   0          6m53s
pod/nginx-deployment-647677fc66-wpxbg  1/1    Running   0          6m53s

NAME            TYPE        CLUSTER-IP   EXTERNAL-IP
service/kubernetes  ClusterIP   10.100.0.1   <none>
service/nginx-service  LoadBalancer  10.100.199.108  aab08f07ba3eb4741aa47f2270a136f6-1440742731.ap-south-1.elb.amazonaws.com
PORT(S)    AGE
443/TCP  27m
80:30911/TCP  3m14s

NAME          READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nginx-deployment  2/2     2           2           6m53s

NAME          DESIRED   CURRENT   READY   AGE
replicaset.apps/nginx-deployment-647677fc66  2        2         2       6m53s
~ $
```

```
~ $ kubectl describe svc nginx-service
Name:           nginx-service
Namespace:      default
Labels:         <none>
Annotations:    <none>
Selector:       app=nginx
Type:          LoadBalancer
IP Family Policy: SingleStack
IP Families:   IPv4
IP:            10.100.199.108
IPs:           10.100.199.108
LoadBalancer Ingress: aab08f07ba3eb4741aa47f2270a136f6-1440742731.ap-south-1.elb.amazonaws.com
Port:          <unset>  80/TCP
TargetPort:    80/TCP
NodePort:      <unset>  30911/TCP
Endpoints:     192.168.20.211:80,192.168.74.59:80
Session Affinity: None
External Traffic Policy: Cluster
Internal Traffic Policy: Cluster
Events:
  Type    Reason             Age    From            Message
  ----  -----            ----  ----            -----
Normal  EnsuringLoadBalancer  4m2s  service-controller  Ensuring load balancer
Normal  EnsuredLoadBalancer  3m59s service-controller  Ensured load balancer
~ $
```

resource cleanup process

```
CloudShell
ap-south-1 + =
```

```
~ $ eksctl delete cluster --name demo-eks-cluster --region ap-south-1
2025-11-03 15:07:01 [i] deleting EKS cluster "demo-eks-cluster"
2025-11-03 15:07:01 [i] will drain 0 unmanaged nodegroup(s) in cluster "demo-eks-cluster"
2025-11-03 15:07:01 [i] starting parallel draining, max in-flight of 1
2025-11-03 15:07:01 [i] deleted 0 Fargate profile(s)
2025-11-03 15:07:01 [✓] kubeconfig has been updated
2025-11-03 15:07:01 [i] cleaning up AWS load balancers created by Kubernetes objects of Kind Service or Ingress
```