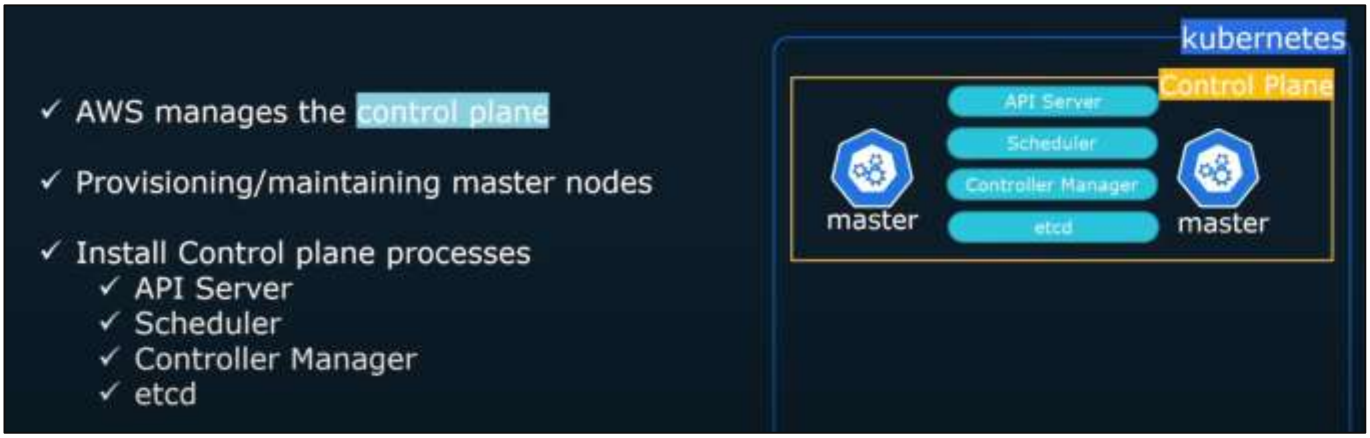
EKS – Managed Kubernetes Service

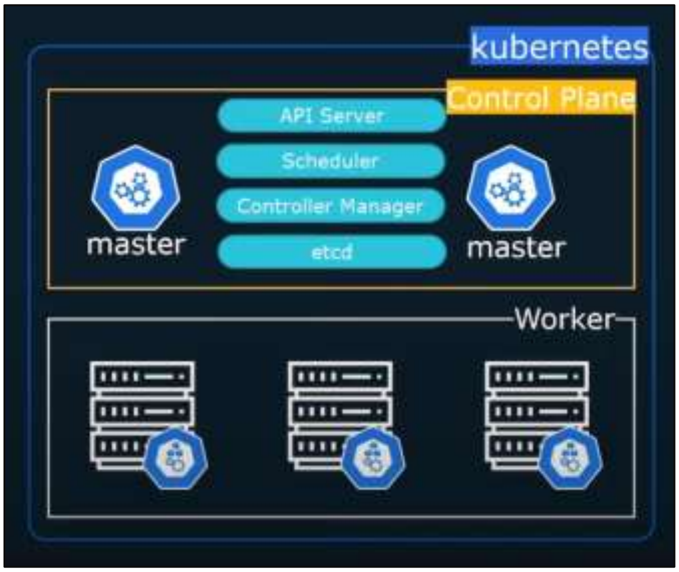
What is EKS Service do?

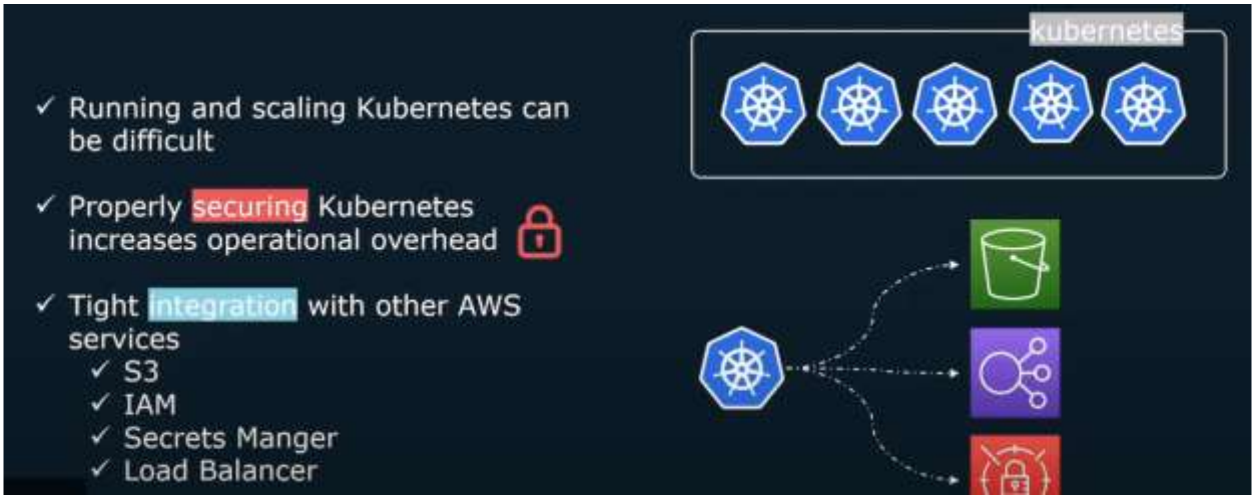


- ✓ AWS manages the control plane
- ✓ Provisioning/maintaining master nodes
- ✓ Install Control plane processes
  - ✓ API Server
  - ✓ Scheduler
  - ✓ Controller Manager
  - ✓ etcd

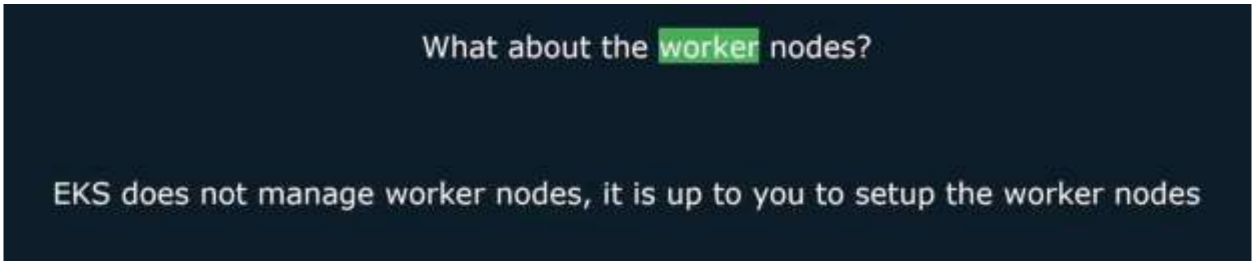
AWS will control all Control Plane (Master Node) of Kubernetes.

As a user, you only need to set up worker nodes controlled by the CONTROL PLANE.





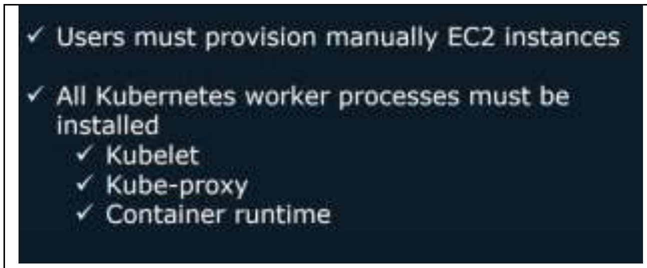
- ✓ Running and scaling Kubernetes can be difficult
- ✓ Properly securing Kubernetes increases operational overhead
- ✓ Tight integration with other AWS services
  - ✓ S3
  - ✓ IAM
  - ✓ Secrets Manger
  - ✓ Load Balancer



What about the worker nodes?

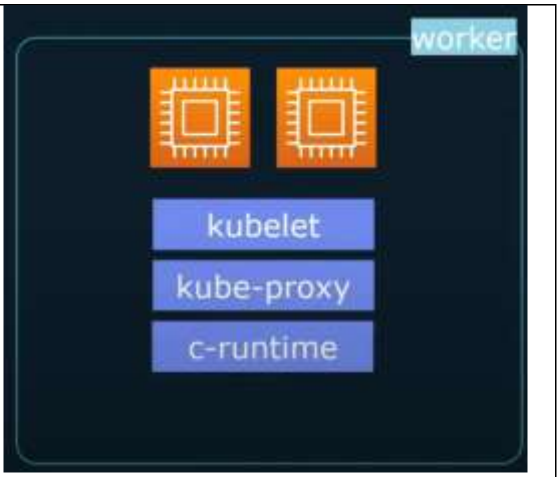
EKS does not manage worker nodes, it is up to you to setup the worker nodes

Self-managed Nodes

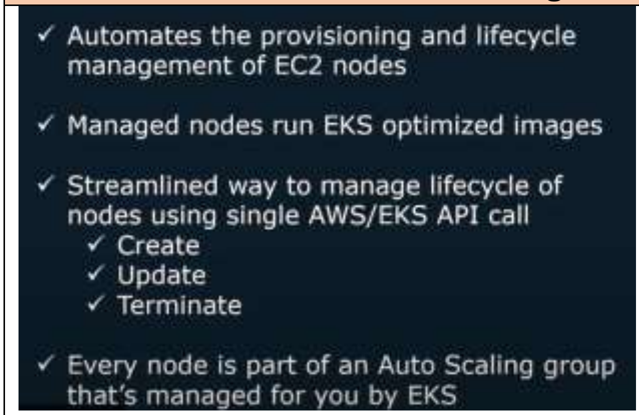


- ✓ Users must provision manually EC2 instances
- ✓ All Kubernetes worker processes must be installed
  - ✓ Kubelet
  - ✓ Kube-proxy
  - ✓ Container runtime

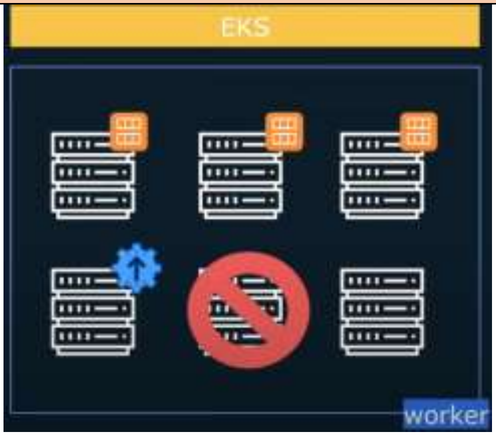
- You need to update machines and security patches.
- You need to register node (s) with the control plane.



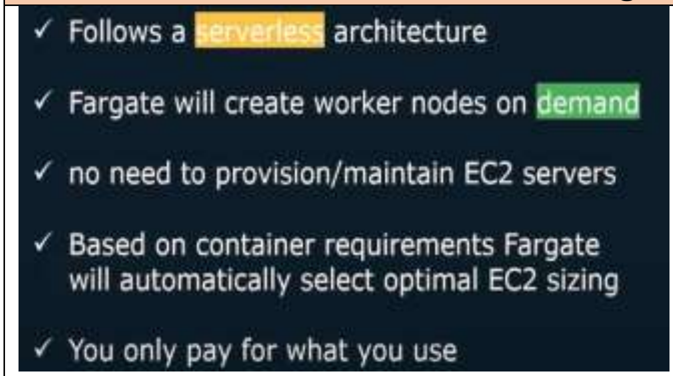
Managed Node Group



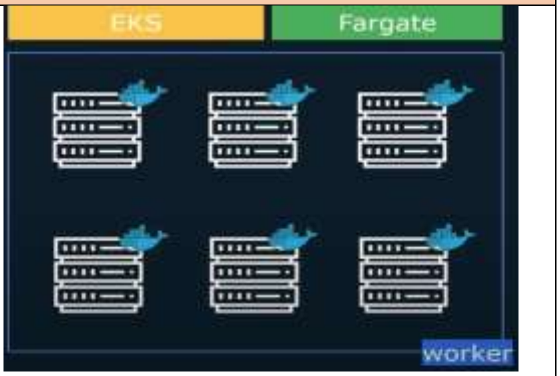
- ✓ Automates the provisioning and lifecycle management of EC2 nodes
- ✓ Managed nodes run EKS optimized images
- ✓ Streamlined way to manage lifecycle of nodes using single AWS/EKS API call
  - ✓ Create
  - ✓ Update
  - ✓ Terminate
- ✓ Every node is part of an Auto Scaling group that's managed for you by EKS



Fargate

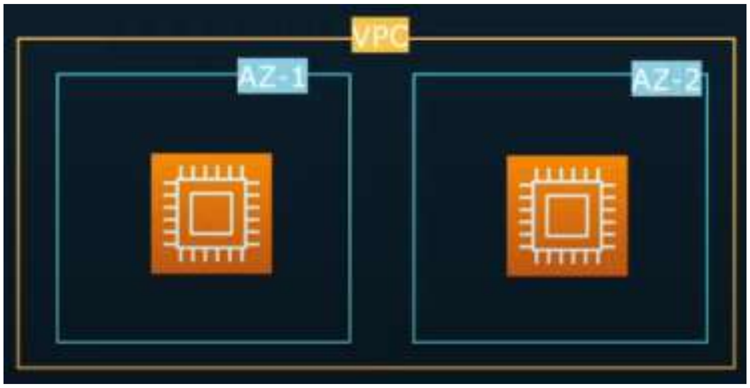


- ✓ Follows a serverless architecture
- ✓ Fargate will create worker nodes on demand
- ✓ no need to provision/maintain EC2 servers
- ✓ Based on container requirements Fargate will automatically select optimal EC2 sizing
- ✓ You only pay for what you use



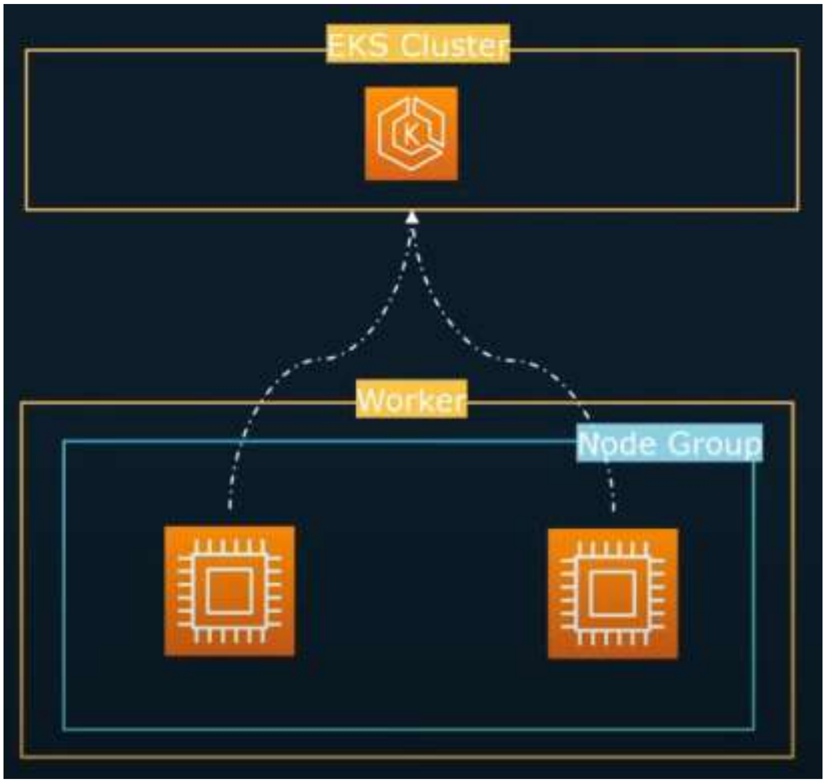
Creating EKS Cluster

- Cluster Name
- K8 Version
- IAM Role for cluster
- Select VPC and Subnets
- Define Security Group for cluster



Creating Worker Nodes

- Create Node Group
- Select Instance Type
- Define Max and Min number of Nodes
- Specify EKS cluster to connect to



Methods to create EKS Cluster

- AWS Console
- Eksctl [Command line tool to create EKS cluster and worker nodes]
- IaC – Terraform/ Pulumi

In this demo, we will use Eksctl command line tool that will make the process easy to setup EKS cluster.

https://eksctl.io/installation/

https://kubernetes.io/docs/tasks/tools/install-kubectl-windows/

Install both tools on windows.

After installation on Windows, open CMD and type this command.

```
C:\users\priyanka>eksctl version
C:\users\priyanka>eksctl --help
C:\users\priyanka>kubectl
```

Before you use eksctl command tool, you can configure access key and secret key of IAM user with admin credentials using > **aws configure**

Let’s create a cluster and 2 worker nodes with eksctl command.

- **eksctl create cluster -n cluster1 --nodegroup-name ng1 --region ap-south-1 --node-type t2.micro --nodes 2**

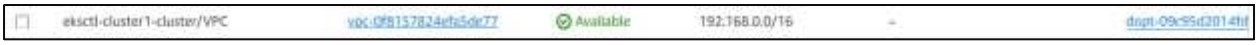
Please Note: It will take a long time to create the cluster and nodes on the AWS cloud.

Take a coffee break ☺

Once it is completed successfully, let us see what is created on AWS.

- **Check VPC - A new VPC is created for the cluster and nodes**
- **2 public and 2 private subnets**
- **Route Table**

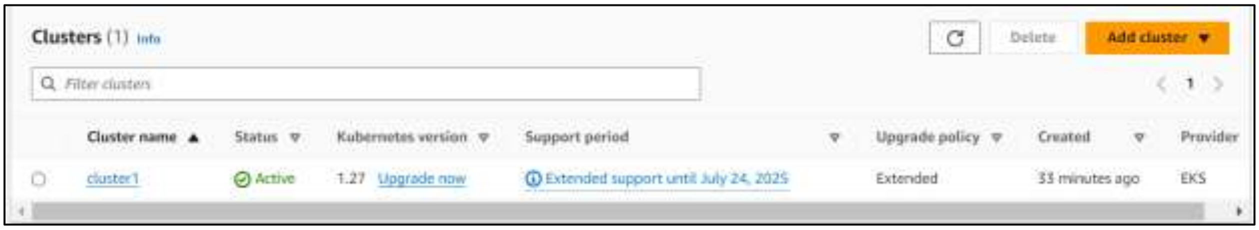
- **Internet Gateway**



□	eksctl-cluster1-cluster/VPC	vpc-0f8157824efa5de77	Available	192.168.0.0/16	-	dopt-09c95d2014fd

Go to VPC service to check all these.

Go to EKS service and you will find it has created a cluster.

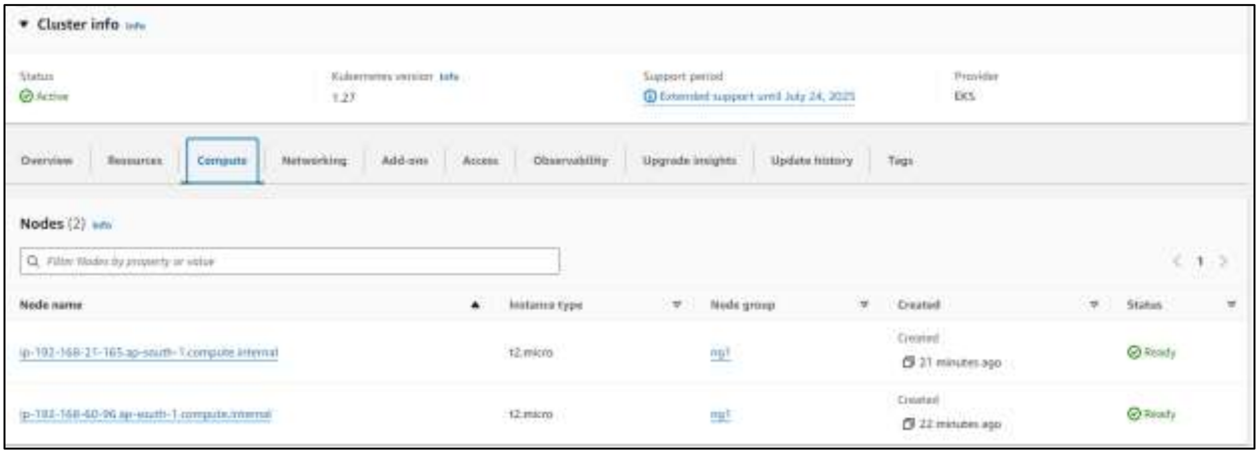


Clusters (1) Info

Filter clusters

	Cluster name	Status	Kubernetes version	Support period	Upgrade policy	Created	Provider
○	cluster1	Active	1.27 Upgrade now	Extended support until July 24, 2025	Extended	33 minutes ago	EKS

Click **cluster1** and under **Compute** you will find 2 worker nodes under node group (ng1).



Cluster info Info

Status	Kubernetes version Info	Support period	Provider
Active	1.27	Extended support until July 24, 2025	EKS

Overview | Resources | Compute | Networking | Add-ons | Access | Observability | Upgrade insights | Update history | Tags

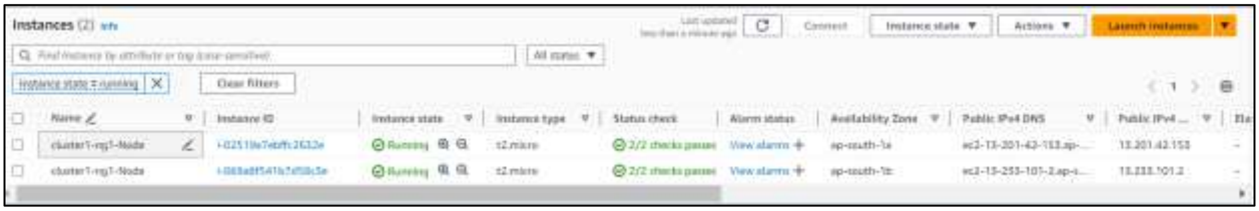
Nodes (2) Info

Filter Nodes by property or value

Node name	Instance type	Node group	Created	Status
ip-192-168-21-165.ap-south-1.compute.internal	t2.micro	ng1	Created 21 minutes ago	Ready
ip-192-168-60-96.ap-south-1.compute.internal	t2.micro	ng1	Created 22 minutes ago	Ready

These worker nodes are EC2 instances.

Go to the EC2 service to check these instances.



Instances (2) Info

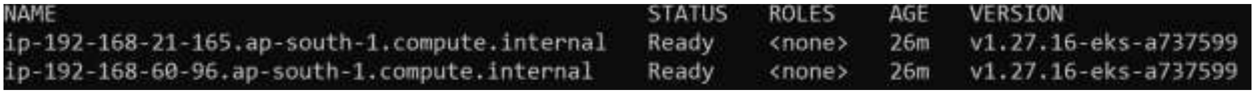
Find Instance by attribute or tag (case-sensitive)

Instance state = running

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Ela
□	cluster1-ng1-Node	i-02519a7ebffc2632e	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1a	ec2-13-201-42-153.ap-...	13.201.42.153	-
□	cluster1-ng1-Node	i-083a8f541b7af58c5e	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1b	ec2-13-233-101-2.ap-s...	13.233.101.2	-

Come back to CMD.

- kubectl get nodes

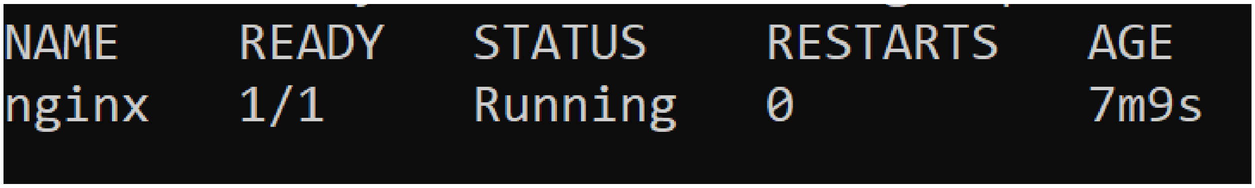


```
NAME                                          STATUS   ROLES    AGE   VERSION
ip-192-168-21-165.ap-south-1.compute.internal Ready    <none>   26m   v1.27.16-eks-a737599
ip-192-168-60-96.ap-south-1.compute.internal  Ready    <none>   26m   v1.27.16-eks-a737599
```

Now let us run an NGINX Pod and access it from cluster.

- kubectl run nginx --image=nginx --port=80
- kubectl expose pod nginx --type=LoadBalancer --port=80 --target-port=80

- kubectl get pods

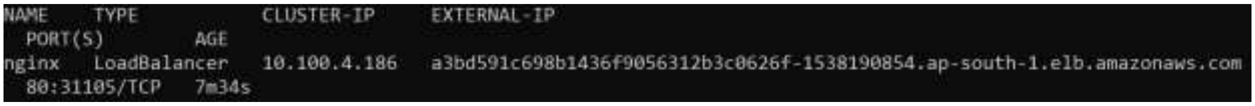


```
NAME    READY   STATUS    RESTARTS   AGE
nginx   1/1     Running   0          7m9s
```

- kubectl get svc

or

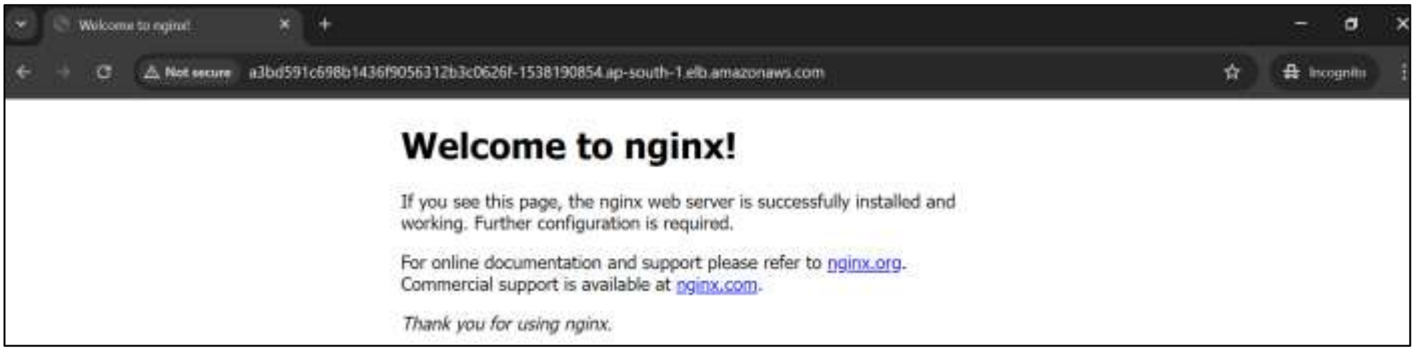
- kubectl get svc nginx



```
NAME    TYPE           CLUSTER-IP     EXTERNAL-IP
  PORT(S)        AGE
nginx   LoadBalancer   10.100.4.186   a3bd591c698b1436f9056312b3c0626f-1538190854.ap-south-1.elb.amazonaws.com
  80:31105/TCP   7m34s
```

Look under the EXTERNAL-IP column. That's the IP address you can use to access your application.

Copy the external IP address of cluster running on aws and run it in the browser.


Welcome to nginx!
Not secure
a3bd591c698b1436f9056312b3c0626f-1538190854.ap-south-1.elb.amazonaws.com
Incognito

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.

To delete the cluster and nodes

- eksctl delete cluster –n cluster1

It will delete cluster as well as nodes from aws.