

```
vignesh@Vignesh eks-setup % aws configure
AWS Access Key ID [*****7SF5]: AKIAVIOZFUX6IFZD7SF5
AWS Secret Access Key [*****cbvB]: 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
Default region name [ap-south-1]:
Default output format [json]:
vignesh@Vignesh eks-setup %
```

```
vignesh@Vignesh eks-setup % aws configure
AWS Access Key ID [*****7SF5]: AKIAVIOZFUX6IFZD7SF5
AWS Secret Access Key [*****cbvB]: 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
Default region name [ap-south-1]:
Default output format [json]:
vignesh@Vignesh eks-setup % aws eks list-clusters
{
  "clusters": []
}
vignesh@Vignesh eks-setup %
```

```
Default output format [json]:
vignesh@Vignesh eks-setup % aws eks list-clusters
```

```
{
  "clusters": []
}
vignesh@Vignesh eks-setup %
```

```
vignesh@Vignesh Desktop % eksctl create cluster -f eks-cluster.yaml
```

```
2025-06-08 08:45:45 [i] eksctl version 0.210.0
2025-06-08 08:45:45 [i] using region ap-south-1
2025-06-08 08:45:45 [!] Amazon EKS will no longer publish EKS-optimized Amazon Linux 2 (AL2) AMIs after November 26th, 2025. Additionally, Kubernetes version 1.32 is the last
onwards, Amazon EKS will continue to release AL2023 and Bottlerocket based AMIs. The default AMI family when creating clusters and nodegroups in Eksctl will be changed to AL20
2025-06-08 08:45:46 [i] setting availability zones to [ap-south-1a ap-south-1c ap-south-1b]
2025-06-08 08:45:46 [i] subnets for ap-south-1a - public:192.168.8.0/19 private:192.168.96.0/19
2025-06-08 08:45:46 [i] subnets for ap-south-1c - public:192.168.32.0/19 private:192.168.128.0/19
2025-06-08 08:45:46 [i] subnets for ap-south-1b - public:192.168.64.0/19 private:192.168.160.0/19
2025-06-08 08:45:46 [i] nodegroup "ng-nodes" will use "ami-0eb04f1ac6539e319" [AmazonLinux2/1.32]
2025-06-08 08:45:47 [i] using EC2 key pair "eks-key"
2025-06-08 08:45:47 [i] using Kubernetes version 1.32
2025-06-08 08:45:47 [i] creating EKS cluster "nginx-cluster-v2" in "ap-south-1" region with un-managed nodes
2025-06-08 08:45:47 [i] 1 nodegroup (ng-nodes) was included (based on the include/exclude rules)
2025-06-08 08:45:47 [i] will create a CloudFormation stack for cluster itself and 1 nodegroup stack(s)
2025-06-08 08:45:47 [i] if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=ap-south-1 --cluster=nginx-cluster-v2'
2025-06-08 08:45:47 [i] Kubernetes API endpoint access will use default of (publicAccess=true, privateAccess=false) for cluster "nginx-cluster-v2" in "ap-south-1"
2025-06-08 08:45:47 [i] CloudWatch logging will not be enabled for cluster "nginx-cluster-v2" in "ap-south-1"
2025-06-08 08:45:47 [i] you can enable it with 'eksctl utils update-cluster-logging --enable-types={SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)} --region=ap-south-1 --cluster=nginx
2025-06-08 08:45:47 [i] default addons metrics-server, vpc-cni, kube-proxy, coredns were not specified, will install them as EKS addons
2025-06-08 08:45:47 [i]
2 sequential tasks: { create cluster control plane "nginx-cluster-v2",
2 sequential sub-tasks: {
2 sequential sub-tasks: {
1 task: { create addons },
wait for control plane to become ready,
},
create nodegroup "ng-nodes",
}
}
2025-06-08 08:45:47 [i] building cluster stack "eksctl-nginx-cluster-v2-cluster"
2025-06-08 08:45:47 [i] deploying stack "eksctl-nginx-cluster-v2-cluster"
2025-06-08 08:46:17 [i] waiting for CloudFormation stack "eksctl-nginx-cluster-v2-cluster"
2025-06-08 08:46:48 [i] waiting for CloudFormation stack "eksctl-nginx-cluster-v2-cluster"
2025-06-08 08:47:49 [i] waiting for CloudFormation stack "eksctl-nginx-cluster-v2-cluster"
2025-06-08 08:48:49 [i] waiting for CloudFormation stack "eksctl-nginx-cluster-v2-cluster"
2025-06-08 08:49:50 [i] waiting for CloudFormation stack "eksctl-nginx-cluster-v2-cluster"
2025-06-08 08:50:51 [i] waiting for CloudFormation stack "eksctl-nginx-cluster-v2-cluster"
2025-06-08 08:51:51 [i] waiting for CloudFormation stack "eksctl-nginx-cluster-v2-cluster"
2025-06-08 08:52:52 [i] waiting for CloudFormation stack "eksctl-nginx-cluster-v2-cluster"
2025-06-08 08:53:52 [i] waiting for CloudFormation stack "eksctl-nginx-cluster-v2-cluster"
2025-06-08 08:53:55 [i] creating addon: metrics-server
2025-06-08 08:53:56 [i] successfully created addon: metrics-server
2025-06-08 08:53:56 [!] recommended policies were found for "vpc-cni" addon, but since OIDC is disabled on the cluster, eksctl cannot configure the requested permissions; the
pod identity associations; after addon creation is completed, add all recommended policies to the config file, under 'addon.PodIdentityAssociations', and run 'eksctl update ad
2025-06-08 08:53:56 [i] creating addon: vpc-cni
2025-06-08 08:53:57 [i] successfully created addon: vpc-cni
2025-06-08 08:53:57 [i] creating addon: kube-proxy
2025-06-08 08:53:57 [i] successfully created addon: kube-proxy
2025-06-08 08:53:58 [i] creating addon: coredns
2025-06-08 08:53:58 [i] successfully created addon: coredns
2025-06-08 08:56:00 [i] building nodegroup stack "eksctl-nginx-cluster-v2-nodegroup-ng-nodes"
2025-06-08 08:56:00 [i] --nodes-min=2 was set automatically for nodegroup ng-nodes
2025-06-08 08:56:00 [i] --nodes-max=2 was set automatically for nodegroup ng-nodes
2025-06-08 08:56:02 [i] deploying stack "eksctl-nginx-cluster-v2-nodegroup-ng-nodes"
2025-06-08 08:56:02 [i] waiting for CloudFormation stack "eksctl-nginx-cluster-v2-nodegroup-ng-nodes"
2025-06-08 08:56:32 [i] waiting for CloudFormation stack "eksctl-nginx-cluster-v2-nodegroup-ng-nodes"
2025-06-08 08:57:31 [i] waiting for CloudFormation stack "eksctl-nginx-cluster-v2-nodegroup-ng-nodes"
2025-06-08 08:59:10 [i] waiting for CloudFormation stack "eksctl-nginx-cluster-v2-nodegroup-ng-nodes"
2025-06-08 08:59:10 [i] waiting for the control plane to become ready
2025-06-08 08:59:11 [✓] saved kubeconfig as "/Users/vignesh/.kube/config"
2025-06-08 08:59:11 [i] no tasks
2025-06-08 08:59:11 [✓] all EKS cluster resources for "nginx-cluster-v2" have been created
2025-06-08 08:59:11 [i] nodegroup "ng-nodes" has 2 node(s)
2025-06-08 08:59:11 [i] node "ip-192-168-19-130.ap-south-1.compute.internal" is ready
2025-06-08 08:59:11 [i] node "ip-192-168-82-135.ap-south-1.compute.internal" is ready
2025-06-08 08:59:11 [i] waiting for at least 2 node(s) to become ready in "ng-nodes"
2025-06-08 08:59:11 [i] nodegroup "ng-nodes" has 2 node(s)
2025-06-08 08:59:11 [i] node "ip-192-168-19-130.ap-south-1.compute.internal" is ready
2025-06-08 08:59:11 [i] node "ip-192-168-82-135.ap-south-1.compute.internal" is ready
2025-06-08 08:59:11 [✓] created 1 nodegroup(s) in cluster "nginx-cluster-v2"
2025-06-08 08:59:13 [i] kubect command should work with "/Users/vignesh/.kube/config", try 'kubectl get nodes'
2025-06-08 08:59:13 [✓] EKS cluster "nginx-cluster-v2" in "ap-south-1" region is ready
```

```
vignesh@Vignesh Desktop %
```

```
2025-06-08 08:59:11 [i] nodegroup "ng-nodes" has 2 node(s)
2025-06-08 08:59:11 [i] node "ip-192-168-19-130.ap-south-1.compute.internal" is ready
2025-06-08 08:59:11 [i] node "ip-192-168-82-135.ap-south-1.compute.internal" is ready
2025-06-08 08:59:11 [i] waiting for at least 2 node(s) to become ready in "ng-nodes"
2025-06-08 08:59:11 [i] nodegroup "ng-nodes" has 2 node(s)
2025-06-08 08:59:11 [i] node "ip-192-168-19-130.ap-south-1.compute.internal" is ready
2025-06-08 08:59:11 [i] node "ip-192-168-82-135.ap-south-1.compute.internal" is ready
2025-06-08 08:59:11 [✓] created 1 nodegroup(s) in cluster "nginx-cluster-v2"
2025-06-08 08:59:13 [i] kubect command should work with "/Users/vignesh/.kube/config", try 'kubectl get nodes'
2025-06-08 08:59:13 [✓] EKS cluster "nginx-cluster-v2" in "ap-south-1" region is ready
```

```
vignesh@Vignesh Desktop % nano eks-cluster.yaml
```

```
vignesh@Vignesh Desktop % kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
ip-192-168-19-130.ap-south-1.compute.internal	Ready	<none>	9m21s	v1.32.3-eks-473151a
ip-192-168-82-135.ap-south-1.compute.internal	Ready	<none>	9m17s	v1.32.3-eks-473151a

```
vignesh@Vignesh Desktop %
```

```
2025-06-08 08:59:13 [✓] EKS cluster "nginx-cluster-v2" in "ap-south-1" region is ready
```

```
vignesh@Vignesh Desktop % nano eks-cluster.yaml
```

```
vignesh@Vignesh Desktop % kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
ip-192-168-19-130.ap-south-1.compute.internal	Ready	<none>	9m21s	v1.32.3-eks-473151a
ip-192-168-82-135.ap-south-1.compute.internal	Ready	<none>	9m17s	v1.32.3-eks-473151a

```
vignesh@Vignesh Desktop % nano nginx-deployment.yaml
```

```
vignesh@Vignesh Desktop % kubectl apply -f nginx-deployment.yaml
```

```
deployment.apps/nginx-deployment created
```

```
vignesh@Vignesh Desktop %
```

```
vignesh@vignesh Desktop % kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
ip-192-168-19-130.ap-south-1.compute.internal	Ready	<none>	9m21s	v1.32.3-eks-473151a
ip-192-168-82-135.ap-south-1.compute.internal	Ready	<none>	9m17s	v1.32.3-eks-473151a

```
vignesh@Vignesh Desktop % nano nginx-deployment.yaml  
vignesh@Vignesh Desktop % kubectl apply -f nginx-deployment.yaml
```

```
deployment.apps/nginx-deployment created
```

```
vignesh@Vignesh Desktop % kubectl get deployments
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
nginx-deployment	0/2	2	0	22s

```
vignesh@Vignesh Desktop %
```

```
nginx-deployment 0/2 2 0 22s
```

```
vignesh@Vignesh Desktop % kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
nginx-deployment-96b9d695-4clwk	0/1	Pending	0	39s
nginx-deployment-96b9d695-ppqtj	0/1	Pending	0	39s

```
vignesh@Vignesh Desktop %
```

```
nginx-deployment-96b9d695-ppqtj 0/1 Pending 0 39s
```

```
vignesh@Vignesh Desktop % kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
nginx-deployment-96b9d695-4clwk	0/1	Pending	0	69s
nginx-deployment-96b9d695-ppqtj	0/1	Pending	0	69s

```
vignesh@Vignesh Desktop % nano nginx-service.yaml
```

```
vignesh@Vignesh Desktop % kubectl apply -f nginx-service.yaml
```

```
service/nginx-service created
```

```
vignesh@Vignesh Desktop %
```

```
2025-06-08 09:33:38 [i]
2 sequential tasks: { delete nodegroup "ng-nodes", delete cluster control plane "nginx-cluster-v2" [async]
}
2025-06-08 09:33:38 [i] will delete stack "eksctl-nginx-cluster-v2-nodegroup-ng-nodes"
2025-06-08 09:33:38 [i] waiting for stack "eksctl-nginx-cluster-v2-nodegroup-ng-nodes" to get deleted
2025-06-08 09:33:38 [i] waiting for CloudFormation stack "eksctl-nginx-cluster-v2-nodegroup-ng-nodes"
2025-06-08 09:34:09 [i] waiting for CloudFormation stack "eksctl-nginx-cluster-v2-nodegroup-ng-nodes"
2025-06-08 09:34:58 [i] waiting for CloudFormation stack "eksctl-nginx-cluster-v2-nodegroup-ng-nodes"
2025-06-08 09:36:12 [i] waiting for CloudFormation stack "eksctl-nginx-cluster-v2-nodegroup-ng-nodes"
2025-06-08 09:36:12 [i] will delete stack "eksctl-nginx-cluster-v2-cluster"
2025-06-08 09:36:13 [✓] all cluster resources were deleted
```

```
vignesh@Vignesh Desktop % eksctl create cluster \
```

```
--name nginx-cluster \
```

```
--region ap-south-1 \
```

```
--nodegroup-name ng-nodes \
```

```
--nodes 2 \
```

```
--node-type t3.medium \
```

```
--nodes-min 2 \
```

```
--nodes-max 3 \
```

```
--managed
```

```
vignesh@Vignesh Desktop % eksctl create cluster \
--name nginx-cluster \
--region ap-south-1 \
--nodegroup-name ng-nodes \
--nodes 2 \
--node-type t3.medium \
--nodes-min 2 \
--nodes-max 3 \
--managed

2025-06-08 09:39:38 [i] eksctl version 0.210.0
2025-06-08 09:39:38 [i] using region ap-south-1
2025-06-08 09:39:39 [i] setting availability zones to [ap-south-1c ap-south-1b ap-south-1a]
2025-06-08 09:39:39 [i] subnets for ap-south-1c - public:192.168.0.0/19 private:192.168.96.0/19
2025-06-08 09:39:39 [i] subnets for ap-south-1b - public:192.168.32.0/19 private:192.168.128.0/19
2025-06-08 09:39:39 [i] subnets for ap-south-1a - public:192.168.64.0/19 private:192.168.160.0/19
2025-06-08 09:39:39 [i] nodegroup "ng-nodes" will use "" [AmazonLinux2023/1.32]
2025-06-08 09:39:39 [i] using Kubernetes version 1.32
2025-06-08 09:39:39 [i] creating EKS cluster "nginx-cluster" in "ap-south-1" region with managed nodes
2025-06-08 09:39:39 [i] will create 2 separate CloudFormation stacks for cluster itself and the initial managed nodegroup
2025-06-08 09:39:39 [i] if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=ap-south-1
2025-06-08 09:39:39 [i] Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "nginx-clu
2025-06-08 09:39:39 [i] CloudWatch logging will not be enabled for cluster "nginx-cluster" in "ap-south-1"
2025-06-08 09:39:39 [i] you can enable it with 'eksctl utils update-cluster-logging --enable-types={SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)}
2025-06-08 09:39:39 [i] default addons kube-proxy, coredns, metrics-server, vpc-cni were not specified, will install them as EKS addons
2025-06-08 09:39:39 [i]
2 sequential tasks: { create cluster control plane "nginx-cluster",
  2 sequential sub-tasks: {
    2 sequential sub-tasks: {
      1 task: { create addons },
      wait for control plane to become ready,
    },
    create managed nodegroup "ng-nodes",
  },
}
2025-06-08 09:39:39 [i] building cluster stack "eksctl-nginx-cluster-cluster"
2025-06-08 09:39:40 [i] deploying stack "eksctl-nginx-cluster-cluster"
2025-06-08 09:40:10 [i] waiting for CloudFormation stack "eksctl-nginx-cluster-cluster"
2025-06-08 09:40:40 [i] waiting for CloudFormation stack "eksctl-nginx-cluster-cluster"
2025-06-08 09:41:41 [i] waiting for CloudFormation stack "eksctl-nginx-cluster-cluster"
2025-06-08 09:42:41 [i] waiting for CloudFormation stack "eksctl-nginx-cluster-cluster"
2025-06-08 09:43:42 [i] waiting for CloudFormation stack "eksctl-nginx-cluster-cluster"
2025-06-08 09:44:42 [i] waiting for CloudFormation stack "eksctl-nginx-cluster-cluster"
2025-06-08 09:45:43 [i] waiting for CloudFormation stack "eksctl-nginx-cluster-cluster"
2025-06-08 09:56:56 [i] waiting for CloudFormation stack "eksctl-nginx-cluster-cluster"
2025-06-08 09:56:58 [i] creating addon: kube-proxy
2025-06-08 09:56:58 [i] successfully created addon: kube-proxy
2025-06-08 09:56:59 [i] creating addon: coredns
2025-06-08 09:56:59 [i] successfully created addon: coredns
2025-06-08 09:57:00 [i] creating addon: metrics-server
2025-06-08 09:57:00 [i] successfully created addon: metrics-server
2025-06-08 09:57:01 [i] recommended policies were found for "vpc-cni" addon, but since OIDC is disabled on the cluster, eksctl cannot conf
pod identity associations; after addon creation is completed, add all recommended policies to the config file, under 'addon.PodIdentityAss
2025-06-08 09:57:01 [i] creating addon: vpc-cni
2025-06-08 09:57:01 [i] successfully created addon: vpc-cni
2025-06-08 09:59:03 [i] building managed nodegroup stack "eksctl-nginx-cluster-nodegroup-ng-nodes"
2025-06-08 09:59:04 [i] deploying stack "eksctl-nginx-cluster-nodegroup-ng-nodes"
2025-06-08 09:59:04 [i] waiting for CloudFormation stack "eksctl-nginx-cluster-nodegroup-ng-nodes"
2025-06-08 09:59:34 [i] waiting for CloudFormation stack "eksctl-nginx-cluster-nodegroup-ng-nodes"
2025-06-08 10:00:33 [i] waiting for CloudFormation stack "eksctl-nginx-cluster-nodegroup-ng-nodes"
2025-06-08 10:04:07 [i] waiting for CloudFormation stack "eksctl-nginx-cluster-nodegroup-ng-nodes"
2025-06-08 10:04:08 [i] waiting for the control plane to become ready
2025-06-08 10:04:08 [✓] saved kubeconfig as "/Users/vignesh/.kube/config"
2025-06-08 10:04:08 [i] no tasks
2025-06-08 10:04:08 [✓] all EKS cluster resources for "nginx-cluster" have been created
2025-06-08 10:04:08 [i] nodegroup "ng-nodes" has 2 node(s)
2025-06-08 10:04:08 [i] node "ip-192-168-28-81.ap-south-1.compute.internal" is ready
2025-06-08 10:04:08 [i] node "ip-192-168-87-92.ap-south-1.compute.internal" is ready
2025-06-08 10:04:08 [i] waiting for at least 2 node(s) to become ready in "ng-nodes"
2025-06-08 10:04:08 [i] nodegroup "ng-nodes" has 2 node(s)
2025-06-08 10:04:08 [i] node "ip-192-168-28-81.ap-south-1.compute.internal" is ready
2025-06-08 10:04:08 [i] node "ip-192-168-87-92.ap-south-1.compute.internal" is ready
2025-06-08 10:04:08 [✓] created 1 managed nodegroup(s) in cluster "nginx-cluster"
2025-06-08 10:04:09 [i] kubectl command should work with "/Users/vignesh/.kube/config", try 'kubectl get nodes'
2025-06-08 10:04:09 [✓] EKS cluster "nginx-cluster" in "ap-south-1" region is ready
vignesh@Vignesh Desktop %
```

```
2025-06-08 10:04:08 [i] node "ip-192-168-87-92.ap-south-1.compute.internal" is ready
2025-06-08 10:04:08 [✓] created 1 managed nodegroup(s) in cluster "nginx-cluster"
2025-06-08 10:04:09 [i] kubectl command should work with "/Users/vignesh/.kube/config", try 'kubectl get nodes'
2025-06-08 10:04:09 [✓] EKS cluster "nginx-cluster" in "ap-south-1" region is ready
vignesh@Vignesh Desktop % kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
ip-192-168-28-81.ap-south-1.compute.internal	Ready	<none>	3m54s	v1.32.3-eks-473151a
ip-192-168-87-92.ap-south-1.compute.internal	Ready	<none>	3m55s	v1.32.3-eks-473151a

```
vignesh@Vignesh Desktop % nano nginx-deployment.yaml
vignesh@Vignesh Desktop % kubectl apply -f nginx-deployment.yaml
```

```
deployment.apps/nginx-deployment created
service/nginx-service created
vignesh@Vignesh Desktop %
```


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.

```
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
vignesh@Vignesh Desktop %
```