



Kops for Kubernetes Setup

Kops, short for Kubernetes Operations, is a command-line tool used for deploying, managing, and operating Kubernetes clusters. It is specifically designed to simplify the process of setting up and managing production-grade Kubernetes clusters on various cloud platforms, such as AWS (Amazon Web Services).

Key features of Kops include:

1. **Cluster Provisioning:** Kops automates the provisioning of Kubernetes clusters, handling tasks such as creating the necessary infrastructure components (e.g., virtual machines, networking, storage) and configuring Kubernetes control plane components (e.g., API server, scheduler, controller manager).
2. **Customization and Configuration:** Kops provides extensive options for customizing and configuring Kubernetes clusters according to specific requirements. Users can specify parameters such as instance types, networking configurations, Kubernetes versions, and add-ons.
3. **High Availability:** Kops supports the creation of highly available Kubernetes clusters with redundant control plane components and worker nodes. This ensures resilience and fault tolerance for production workloads.
4. **Upgrades and Maintenance:** Kops simplifies the process of upgrading Kubernetes clusters to newer versions. It provides commands to perform rolling upgrades of control plane components and worker nodes, minimizing downtime and disruption to running applications.
5. **Cluster Lifecycle Management:** Kops supports various lifecycle management operations for Kubernetes clusters, including scaling clusters up or down, adding or removing nodes, and performing cluster backups and restores.
6. **Integration with Cloud Providers:** Kops integrates seamlessly with cloud providers' APIs to provision and manage Kubernetes clusters on platforms such as AWS, GCP (Google Cloud Platform), and Azure. It leverages cloud-specific features and resources to optimize cluster deployment and operation.
7. **Infrastructure as Code (IaC):** Kops allows users to define Kubernetes cluster configurations as code using YAML manifests. This enables infrastructure automation and version control, facilitating reproducibility and consistency in cluster deployments.

Overall, Kops is a powerful tool for automating the deployment and management of Kubernetes clusters, particularly in production environments where reliability, scalability, and ease of operation are paramount. It abstracts away the complexities of setting up Kubernetes infrastructure, allowing users to focus on deploying and running containerized applications effectively.



Use cases of Kops:

Kops (Kubernetes Operations) is commonly used across various use cases for deploying, managing, and operating Kubernetes clusters in production environments. Here are some common use cases:

1. **Cloud-Native Applications:** Organizations building cloud-native applications often use Kops to deploy and manage Kubernetes clusters on cloud platforms such as AWS, GCP, or Azure. Kops simplifies the process of setting up infrastructure for containerized workloads, enabling developers to focus on building and deploying applications.
2. **Microservices Architecture:** Kops is well-suited for managing microservices-based architectures where applications are composed of multiple loosely coupled services. It allows teams to deploy and scale individual microservices independently within Kubernetes clusters, ensuring flexibility and agility in development and operations.
3. **Continuous Integration/Continuous Deployment (CI/CD):** Kops integrates seamlessly with CI/CD pipelines to automate the deployment and testing of applications. Organizations use Kops to provision Kubernetes clusters dynamically, deploy application updates, and perform rolling upgrades as part of their CI/CD workflows, enabling rapid and reliable software delivery.
4. **Highly Available Production Workloads:** Kops supports the creation of highly available Kubernetes clusters with redundant control plane components and worker nodes. Organizations leverage Kops to deploy production workloads with stringent availability requirements, ensuring resilience and fault tolerance against node failures or infrastructure issues.
5. **Scalable Infrastructure:** Kops enables organizations to scale Kubernetes clusters up or down based on workload demands. It provides features for dynamically adding or removing worker nodes, adjusting resource allocations, and optimizing cluster capacity utilization, allowing applications to scale seamlessly as demand fluctuates.
6. **Multi-Cloud Deployments:** Organizations operating in multi-cloud environments use Kops to deploy Kubernetes clusters across different cloud providers. Kops abstracts away cloud-specific details and provides a consistent deployment experience, enabling applications to run seamlessly across hybrid or multi-cloud environments.
7. **DevOps and Infrastructure Automation:** Kops facilitates DevOps practices by automating infrastructure provisioning and management tasks. Organizations use Kops as part of their infrastructure automation efforts to define Kubernetes cluster configurations as code, enabling version-controlled deployments, reproducibility, and consistency across environments.
8. **Big Data and Analytics:** Kops is used for deploying Kubernetes clusters to run big data and analytics workloads, leveraging tools and frameworks such as Apache Spark, Apache Hadoop, and Elasticsearch. Kubernetes provides a scalable and flexible platform for processing large datasets and running analytics applications efficiently.
9. **Edge Computing:** In edge computing scenarios, where applications are deployed closer to end-users or IoT devices, Kops enables organizations to deploy Kubernetes clusters at the edge. Kops provides centralized management and orchestration capabilities, allowing organizations to run containerized workloads in distributed edge environments.

10. Research and Development: Researchers and developers use Kops to set up Kubernetes clusters for experimenting with new technologies, testing applications, and exploring advanced Kubernetes features. Kops simplifies the process of creating isolated development environments, enabling rapid iteration and innovation.

That's a comprehensive guide to setting up Kubernetes clusters using Kops on AWS! It covers everything from initial setup with EC2, S3, and IAM to installing and configuring Kops and Kubernetes components. The step-by-step instructions are clear and easy to follow, making it accessible for users with varying levels of experience.

One thing to note is the importance of ensuring security throughout the process, especially when dealing with IAM credentials and exposing services to the internet. It's essential to follow best practices and implement proper security measures to protect the cluster and its resources.

Overall, this guide provides a solid foundation for deploying and managing Kubernetes clusters on AWS using Kops, enabling users to leverage the power of Kubernetes for their containerized applications. It's a valuable resource for anyone looking to get started with Kubernetes orchestration on AWS.

To begin with the Lab:

1. There is a prerequisite for this lab which is you should be having a domain with you from any domain service provider.
2. Now to start this lab, open your Amazon console and navigate to EC2. Here we are going to launch an EC2 instance with Ubuntu OS.
3. Then create a key pair, after that while creating your security group, name it as per your convenience.
4. Then just launch your instance. After that, you should **navigate to S3** and create a bucket.
5. In S3 give your bucket a unique name, and it should be in the same region as your instance is. Then just create your bucket.

General configuration

AWS Region
Asia Pacific (Singapore) ap-southeast-1

Bucket name [Info](#)

Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#) 

Copy settings from existing bucket - *optional*
Only the bucket settings in the following configuration are copied.

[Choose bucket](#)

Format: s3://bucket/prefix

6. After that you have to go to IAM. Now Kops is going to create a cluster and it will be using AWS CLI command. So, for that, you need to provide the credentials.
7. So, now you are going to create an IAM user. For that in IAM go to users and click on Create user.
8. After that give your user a name and click on next.

Specify user details

User details

User name
The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)

Provide user access to the AWS Management Console - *optional*
If you're providing console access to a person, it's a [best practice](#) to manage their access in IAM Identity Center.

If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user.
[Learn more](#)

[Cancel](#) [Next](#)

9. Then you have to choose Attach policies directly and select Administrator Access.

Set permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

Permissions options

Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.

Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

Permissions policies (1/1205)

Choose one or more policies to attach to your new user.

Policy name	Type	Attached entities
<input checked="" type="checkbox"/> AdministratorAccess	AWS managed - job function	7

[Create policy](#)

10. After that just create your user and then go inside your user. There you have to choose security credentials.

[Permissions](#) [Groups](#) [Tags](#) [Security credentials](#) [Access Advisor](#)

11. From access keys you need to click on create.

Access keys (0)

Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. [Learn more](#)

No access keys. As a best practice, avoid using long-term credentials like access keys. Instead, use tools which provide short term credentials. [Learn more](#)

[Create access key](#)

12. In there choose command line interface and click on next.

Access key best practices & alternatives [Info](#)

Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.

Use case

Command Line Interface (CLI)

You plan to use this access key to enable the AWS CLI to access your AWS account.

13. After that just create your access keys and download the CSV file.

Retrieve access keys [Info](#)

Access key

If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key

Secret access key

AKIA4ZIQ7TEGNK4DNO7T

***** [Show](#)

Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the [best practices for managing AWS access keys](#).

[Download .csv file](#)

[Done](#)

14. Now navigate to Route 53, from the dashboard of Route 53 navigate to hosted zones and click on create.

15. We are using Route 53 for Kubernetes DNS records.

16. **Kubernetes DNS records, also known as kube-dns or CoreDNS, are a built-in service that allows Kubernetes pods and services to communicate using hostnames instead of IP addresses. DNS records are important for large-scale, complex environments like Kubernetes, where it's difficult to keep track of IP addresses.**

The screenshot shows the AWS Route 53 'Hosted zones' page. At the top, there's a breadcrumb navigation 'Route 53 > Hosted zones'. Below it is a header with 'Hosted zones (0)' and several buttons: 'View details', 'Edit', 'Delete', and a prominent orange 'Create hosted zone' button. A search bar labeled 'Filter records by property or value' is present. The main content area has a heading 'No hosted zones' and a message stating 'There are no hosted zones created for this account.' Below this is another orange 'Create hosted zone' button.

17. Here you have to give the domain name and then choose public hosted zone and click on create.

The screenshot shows the 'Hosted zone configuration' page. It starts with a brief description of what a hosted zone is. The first section is 'Domain name' with an 'Info' link, containing a text input field with the value 'cloudservicesdemo.in'. Below it is a note about valid characters. The next section is 'Description - optional' with an 'Info' link, featuring a text area with placeholder text 'The hosted zone is used for...'. A note says the description can have up to 256 characters. The final section is 'Type' with an 'Info' link, showing two options: 'Public hosted zone' (selected) and 'Private hosted zone'. A note explains that a public hosted zone routes traffic on the internet.

18. Below you can see that you have the URL for ns servers. Now we are going to our domain registrar and add these ns servers URL.

Records (2)		DNSSEC signing	Hosted zone tags (0)		
Records (2) Info		Delete record Import zone file Create record			
Automatic mode is the current search behavior optimized for best filter results. To change modes go to settings.					
<input type="text"/> Filter records by property or value Type Routing policy Alias Value/Route traffic to TTL (s...) Health ...					
Record ...	Type	Routin...	Differ...		
<input type="checkbox"/> cloudservi...	NS	Simple	-		
			No		
			ns-79.awsdns-09.com. ns-927.awsdns-51.net. ns-1208.awsdns-23.org. ns-1659.awsdns-15.co.uk.		
<input type="checkbox"/> cloudservi...	SOA	Simple	-		
			No		
			ns-79.awsdns-09.com. awsdns-9...		
			900		
			-		

19. In GoDaddy navigate to your domain and then click on domain settings.

The screenshot shows the GoDaddy Domain Settings page for the domain `clouddservicesdemo.in`. The left sidebar includes links for Dashboard, Domain (which is selected), Website, Email, Ways To Sell, and Marketing. The main area displays the domain name, a 'PROTECT YOUR BRAND AND SAVE MONEY' offer for various extensions, and a search bar. At the top right, there are buttons for Forward Domain, Manage DNS, and Domain Settings, with 'Domain Settings' being the active tab.

20. In domain settings you have to open DNS management. Then you have to click on Nameservers. Now click on change Nameservers.

DNS Management

The screenshot shows the GoDaddy DNS Management page for the domain `clouddservicesdemo.in`. The top navigation bar includes a search bar for domains, a 'Domain Settings' button, and a 'Select a different domain' button. Below the search bar, tabs for DNS Records, Forwarding, Nameservers (which is selected), Premium DNS, Hostnames, and DS Records are visible. A note states that Nameservers determine where your DNS is hosted. The 'Using custom nameservers' section contains a 'Change Nameservers' button.

21. In edit nameservers you have to choose I'll use my own nameservers and then copy the ns server from route 53 hosted zone and paste it here one by one.

Edit nameservers

Choose nameservers for `cloudservicesdemo.in`

GoDaddy Nameservers (recommended)

I'll use my own nameservers

ns-79.awsdns-09.com



ns-927.awsdns-51.net



ns-1208.awsdns-23.org



ns-1659.awsdns-15.co.uk



[+ Add Nameserver](#)

Save

Cancel

22. Once it is done now we will login to our EC2 instance and here we will setup rest of the things.

23. Now you are going to generate a key which will be used by Kops.

ssh-keygen

```

ubuntu@ip-172-31-16-149:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_rsa
Your public key has been saved in /home/ubuntu/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:RaB0s50WH465639Hw5Uf0rHb5vNI20U8DD7Ke4Nt86o ubuntu@ip-172-31-16-149
The key's randomart image is:
+---[RSA 3072]---+
| . +.o .
| . o = B . .
| . . B o . +
| o . . =.
| S . .o B|
| ..ooB=|
| . +oo**+|
| ....Boo+|
| .E*o*o+|
+---[SHA256]---+
ubuntu@ip-172-31-16-149:~$ |

```

24. After that you have update your instance and install AWS CLI in it.

```

sudo apt update
sudo snap install aws-cli --classic

```

25. Once your AWS CLI has been installed now you are going to configure your CLI. Here you have to give your access key and then your secret access key. After that give the region where your resources reside then what should be the output format.

```
aws configure
```

```

ubuntu@ip-172-31-16-149:~$ aws configure
AWS Access Key ID [None]: AKIA4ZIQ7TEGNK4DN07T
AWS Secret Access Key [None]: RxHLLwn5faKAVCr50+XHLIZNNlNLYLX5A/4i42zD
Default region name [None]: ap-southeast-1
Default output format [None]: json
ubuntu@ip-172-31-16-149:~$ |

```

Installing kubectl on Linux and Kops:

- Now you are going to install kubectl on your ubuntu instance. For that you can this link below and from its official documentation copy the script and paste it inside your SSH session.

<https://kubernetes.io/docs/tasks/tools/install-kubectl-linux/>

- Or you can use the script shown below to download kubectl.

```
curl -LO https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl
```

- Once it has been downloaded now if you do a listing then you can see kubectl.

```
ubuntu@ip-172-31-16-149:~$ ls  
kubectl  
ubuntu@ip-172-31-16-149:~$ |
```

- Now we are going to give it executable permission and then we're going to move it to **usr local bin**. So if you move it over there then we can access it from anywhere.
- Use the below commands and paste them one by one in your instance. With kubectl help command you can get to know about it.

```
chmod +x ./kubectl  
sudo mv kubectl /usr/local/bin  
kubectl --help
```

```
ubuntu@ip-172-31-16-149:~$ chmod +x ./kubectl  
ubuntu@ip-172-31-16-149:~$ sudo mv kubectl /usr/local/bin  
ubuntu@ip-172-31-16-149:~$ kubectl --help  
kubectl controls the Kubernetes cluster manager.  
  
Find more information at: https://kubernetes.io/docs/reference/kubectl/  
  
Basic Commands (Beginner):  
create      Create a resource from a file or from stdin  
expose      Take a replication controller, service, deployment or pod and expose it as a new Kubernetes service  
run         Run a particular image on the cluster  
set         Set specific features on objects  
  
Basic Commands (Intermediate):  
explain     Get documentation for a resource  
get         Display one or many resources  
edit        Edit a resource on the server  
delete      Delete resources by file names, stdin, resources and names, or by resources and label selector  
  
Deploy Commands:  
rollout    Manage the rollout of a resource  
scale      Set a new size for a deployment, replica set, or replication controller  
autoscale  Auto-scale a deployment, replica set, stateful set, or replication controller  
  
Cluster Management Commands:  
certificate Modify certificate resources  
cluster-info Display cluster information  
top          Display resource (CPU/memory) usage  
cordon      Mark node as unschedulable  
uncordon    Mark node as schedulable  
drain       Drain node in preparation for maintenance  
taint       Update the taints on one or more nodes
```

- Now we will install Kops on our instance. For that, you have to navigate to this link down below. Or you can use the official documentation page for Kops to install it.

<https://github.com/kubernetes/kops/releases/tag/v1.26.4>

- From this page you have to right-click on kops-linux-amd64 and then copy its link address. Then paste it into your instance.

- Below you can see that Kops has been successfully installed.

```
wget https://github.com/kubernetes/kops/releases/download/v1.26.4/kops-linux-amd64
```

```
ubuntu@ip-172-31-16-149:~$ wget https://github.com/kubernetes/kops/releases/download/v1.26.4/kops-linux-amd64
--2024-04-19 10:10:52-- https://github.com/kubernetes/kops/releases/download/v1.26.4/kops-linux-amd64
Resolving github.com (github.com)... 20.205.243.166
Connecting to github.com (github.com)|20.205.243.166|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/62091339/b6609eac-fe3d-42a9-913d-017b771cb536?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAVCODYLSA53PQK4ZA%2F20240419%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20240419T101052Z&X-Amz-Expires=300&X-Amz-Signature=9b9a5340d103cb8c972a7eb7c874d9e1d9cf7e01126c1330008db9ab68b5cb69&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=62091339&response-content-disposition=attachment%3B%20filename%3Dkops-linux-amd64&response-content-type=application%2Foctet-stream [following]
--2024-04-19 10:10:52-- https://objects.githubusercontent.com/github-production-release-asset-2e65be/62091339/b6609eac-fe3d-42a9-913d-017b771cb536?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAVCODYLSA53PQK4ZA%2F20240419%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20240419T101052Z&X-Amz-Expires=300&X-Amz-Signature=9b9a5340d103cb8c972a7eb7c874d9e1d9cf7e01126c1330008db9ab68b5cb69&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=62091339&response-content-disposition=attachment%3B%20filename%3Dkops-linux-amd64&response-content-type=application%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.108.133, 185.199.109.133, 185.199.110.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 175179407 (167M) [application/octet-stream]
Saving to: ‘kops-linux-amd64’

kops-linux-amd64                                     100%[=====] 167.06M  16.4MB/s    in 10s

2024-04-19 10:11:04 (16.1 MB/s) - ‘kops-linux-amd64’ saved [175179407/175179407]

ubuntu@ip-172-31-16-149:~$ |
```

9. Now if you do listing then you can see the binary.

```
ubuntu@ip-172-31-16-149:~$ ls
kops-linux-amd64
ubuntu@ip-172-31-16-149:~$ |
```

10. First we are going to make it executable.

```
chmod +x kops-linux-amd64
sudo mv kops-linux-amd64 /usr/local/bin/kops
```

```
ubuntu@ip-172-31-16-149:~$ chmod +x kops-linux-amd64
ubuntu@ip-172-31-16-149:~$ sudo mv kops-linux-amd64 /usr/local/bin/kops
ubuntu@ip-172-31-16-149:~$ kops
kOps is Kubernetes Operations.

kOps is the easiest way to get a production grade Kubernetes cluster up and running. We like to think of it as kubectl for clusters.

kOps helps you create, destroy, upgrade and maintain production-grade, highly available, Kubernetes clusters from the command line. AWS (Amazon Web Services) is currently officially supported, with Digital Ocean and OpenStack in beta support.

Usage:
  kops [command]

Available Commands:
  completion  Generate the autocompletion script for the specified shell
  create      Create a resource by command line, filename or stdin.
  delete      Delete clusters, instancegroups, instances, and secrets.
  distrust   Distrust keypairs.
  edit        Edit clusters and other resources.
  export      Export configuration.
  get         Get one or many resources.
```

11. Now we are going to verify our domain, for that run the command below and you will be able to see your domain and the your four ns servers.

```
nslookup -type=ns cloudservicesdemo.in
```

```

ubuntu@ip-172-31-16-149:~$ nslookup -type=ns clouddemo.in
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
clouddemo.in    nameserver = ns-927.awsdns-51.net.
clouddemo.in    nameserver = ns-1208.awsdns-23.org.
clouddemo.in    nameserver = ns-1659.awsdns-15.co.uk.
clouddemo.in    nameserver = ns-79.awsdns-09.com.

Authoritative answers can be found from:

ubuntu@ip-172-31-16-149:~$ |

```

12. Now we will run the Kops command which will create the Kubernetes cluster.
13. This command will create a Kubernetes cluster named "clouddemo.in" in the AWS region ap-southeast-1, spread across two availability zones. It will consist of one master node and two worker nodes, with specified instance types and volume sizes. The cluster's DNS will be configured to use the DNS zone "clouddemo.in".

kops create cluster: This is the command to create a Kubernetes cluster using Kops.

--name=clouddemo.in: This specifies the name of the Kubernetes cluster to be created. In this case, the cluster name is set to "clouddemo.in".

--state=s3://demo-kops-bucket: This specifies the state storage location for the cluster configuration. Kops uses an S3 bucket to store cluster configuration and state information. In this case, the bucket named "demo-kops-bucket" is used.

--zones=ap-southeast-1a,ap-southeast-1b: This specifies the availability zones in which the cluster's nodes will be deployed. In this case, the cluster nodes will be spread across availability zones "ap-southeast-1a" and "ap-southeast-1b" in the AWS region ap-southeast-1.

--node-count=2: This specifies the number of worker nodes to be provisioned in the cluster. In this case, the cluster will have 2 worker nodes.

--node-size=t3.small: This specifies the instance type for the worker nodes. In this case, the worker nodes will be of type "t3.small".

--master-size=t3.medium: This specifies the instance type for the master node. In this case, the master node will be of type "t3.medium".

--dns-zone=clouddemo.in: This specifies the DNS zone to be used for the cluster's DNS configuration. In this case, the DNS zone "clouddemo.in" will be used.

--node-volume-size=8: This specifies the size of the volume attached to each worker node. In this case, the volume size is set to 8 GB.

--master-volume-size=8: This specifies the size of the volume attached to the master node. In this case, the volume size is set to 8 GB.

```
kops create cluster --name=cloudservicesdemo.in \
--state=s3://demo-kops-bucket --zones=ap-southeast-1a,ap-southeast-1b \
--node-count=2 --node-size=t3.small --master-size=t3.medium \
--dns-zone=cloudservicesdemo.in \
--node-volume-size=8 --master-volume-size=8
```

```
ubuntu@ip-172-31-16-149:~$ kops create cluster --name=cloudservicesdemo.in \
--state=s3://demo-kops-bucket --zones=ap-southeast-1a,ap-southeast-1b \
--node-count=2 --node-size=t3.small --master-size=t3.medium --dns-zone=cloudservicesdemo.in \
--node-volume-size=8 --master-volume-size=8
I0419 10:29:26.999290    3386 new_cluster.go:1354] Cloud Provider ID: "aws"
I0419 10:29:27.107088    3386 subnets.go:185] Assigned CIDR 172.20.32.0/19 to subnet ap-southeast-1a
I0419 10:29:27.107382    3386 subnets.go:185] Assigned CIDR 172.20.64.0/19 to subnet ap-southeast-1b
Previewing changes that will be made:
```

14. So it created the cluster and stored the information in the S3 bucket.

15. Now you are going to run this command shown below.

```
kops update cluster --name cloudservicesdemo.in --state=s3://demo-kops-bucket --yes
```

```
Suggestions:
 * list clusters with: kops get cluster
 * edit this cluster with: kops edit cluster cloudservicesdemo.in
 * edit your node instance group: kops edit ig --name=cloudservicesdemo.in nodes-ap-southeast-1a
 * edit your control-plane instance group: kops edit ig --name=cloudservicesdemo.in control-plane-ap-southeast-1a

Finally configure your cluster with: kops update cluster --name cloudservicesdemo.in --yes --admin

ubuntu@ip-172-31-16-149:~$ |
```

16. Now once you have executed the above command it will take around 15 minutes.

```
ubuntu@ip-172-31-16-149:~$ kops update cluster --name cloudservicesdemo.in --state=s3://demo-kops-bucket --yes
I0420 05:31:01.865169   6120 executor.go:111] Tasks: 0 done / 100 total; 46 can run
I0420 05:31:02.048392   6120 keypair.go:226] Issuing new certificate: "etcd-peers-ca-events"
W0420 05:31:02.073860   6120 vfs_keystorereader.go:143] CA private key was not found
I0420 05:31:02.074447   6120 keypair.go:226] Issuing new certificate: "etcd-clients-ca"
I0420 05:31:02.089382   6120 keypair.go:226] Issuing new certificate: "etcd-manager-ca-events"
I0420 05:31:02.119899   6120 keypair.go:226] Issuing new certificate: "apiserver-aggregator-ca"
I0420 05:31:02.212771   6120 keypair.go:226] Issuing new certificate: "etcd-peers-ca-main"
I0420 05:31:02.322235   6120 keypair.go:226] Issuing new certificate: "etcd-manager-ca-main"
W0420 05:31:02.372169   6120 vfs_keystorereader.go:143] CA private key was not found
I0420 05:31:02.795296   6120 keypair.go:226] Issuing new certificate: "service-account"
I0420 05:31:02.956760   6120 keypair.go:226] Issuing new certificate: "kubernetes-ca"
I0420 05:31:04.575355   6120 executor.go:111] Tasks: 46 done / 100 total; 21 can run
I0420 05:31:05.827013   6120 executor.go:111] Tasks: 67 done / 100 total; 27 can run
I0420 05:31:06.832522   6120 executor.go:111] Tasks: 90 done / 100 total; 3 can run
I0420 05:31:07.646039   6120 executor.go:155] No progress made, sleeping before retrying 3 task(s)
I0420 05:31:17.646904   6120 executor.go:111] Tasks: 94 done / 100 total; 3 can run
I0420 05:31:19.780756   6120 executor.go:111] Tasks: 97 done / 100 total; 3 can run
I0420 05:31:19.866328   6120 executor.go:111] Tasks: 100 done / 100 total; 0 can run
I0420 05:31:22.114858   6120 dns.go:236] Pre-creating DNS records
I0420 05:31:22.828055   6120 update_cluster.go:323] Exporting kubeconfig for cluster
kOps has set your kubectl context to cloudservicesdemo.in
W0420 05:31:22.071176   6120 update_cluster.go:348] Exported kubeconfig with no user authentication; use --admin, --user or --auth-plugin flags with `kops export kubeconfig`
```

Cluster is starting. It should be ready in a few minutes.

```
Suggestions:
 * validate cluster: kops validate cluster --wait 10m
 * list nodes: kubectl get nodes --show-labels
 * ssh to a control-plane node: ssh -i ~/.ssh/id_rsa ubuntu@api.cloudservicesdemo.in
 * the ubuntu user is specific to Ubuntu. If not using Ubuntu please use the appropriate user based on your OS.
 * read about installing addons at: https://kops.sigs.k8s.io/addons.
```

17. After that you have to validate the cluster. For that, you are going to run this command below.

```
kops validate cluster --state=s3://demo-kops-bucket
```

18. Then you will this type of output on your screen.

```
ubuntu@ip-172-31-25-114:~$ kops validate cluster --state=s3://demo-kops-bucket
Using cluster from kubectl context: cloudservicesdemo.in

Validating cluster cloudservicesdemo.in

INSTANCE GROUPS
NAME          ROLE      MACHINETYPE  MIN  MAX  SUBNETS
control-plane-ap-southeast-1a  ControlPlane  t3.medium   1    1    ap-southeast-1a
nodes-ap-southeast-1a        Node       t3.small     1    1    ap-southeast-1a
nodes-ap-southeast-1b        Node       t3.small     1    1    ap-southeast-1b

NODE STATUS
NAME          ROLE      READY
i-05dc882694393b763  control-plane  True
i-05e8745507bf88015  node        True
i-0d23ea32b3873c34a  node        True

Your cluster cloudservicesdemo.in is ready
```

19. Just in case your validation is failing then you have to use these three commands and your cluster will be working fine then.

```
ubuntu@ip-172-31-25-114:~$ kops validate cluster --state=s3://demo-kops-bucket
Using cluster from kubectl context: cloudservicesdemo.in

Validating cluster cloudservicesdemo.in

Error: validation failed: unexpected error during validation: error listing nodes: Unauthorized
```

20. Below you can see three commands here you have to replace the bucket name with your bucket name. Now copy all three commands at once and paste them in your session then execute them.

```
export KOPS_STATE_STORE=s3://demo-kops-bucket
kops export kubecfg --admin
```

```
ubuntu@ip-172-31-27-17:~$ export KOPS_STATE_STORE=s3://demo-kops-bucket
ubuntu@ip-172-31-27-17:~$ kops export kubecfg --admin
Using cluster from kubectl context: cloudservicesdemo.in

kOps has set your kubectl context to cloudservicesdemo.in
```

21. Now if you will run this command below then you can get to know about your nodes.

kubectl get nodes

22. Below you can see that we have one master node and two worker nodes.

```
ubuntu@ip-172-31-25-114:~$ kubectl get nodes
NAME          STATUS  ROLES      AGE  VERSION
i-05dc882694393b763  Ready  control-plane  16m  v1.26.13
i-05e8745507bf88015  Ready  node        13m  v1.26.13
i-0d23ea32b3873c34a  Ready  node        14m  v1.26.13
```

23. Now if you will open your AWS Console then you can see that three instances have been launched.

Instances (3) Info					
		C	Connect	Instance state ▾	Actions ▾
Find Instance by attribute or tag (case-sensitive)		All states ▾			
Cloudservicesdemo.in X		Clear filters		Launch instances ▾	
<input type="checkbox"/>	Name ↴	Instance ID	Instance state	Instance type	Status check
<input type="checkbox"/>	nodes-ap-southeast-1b.cloudservicesdemo.in	i-05e8745507bf88015	Running Q Q	t3.small	2/2 checks passed View alarm
<input type="checkbox"/>	nodes-ap-southeast-1a.cloudservicesdemo.in	i-0d23ea32b3873c34a	Running Q Q	t3.small	2/2 checks passed View alarm
<input type="checkbox"/>	control-plane-ap-southeast-1a.masters.cloudservicesdemo.in	i-05dc882694393b763	Running Q Q	t3.medium	2/2 checks passed View alarm

24. These have been launched through the Auto Scaling group.

Auto Scaling groups (3) Info						
		C	Launch configurations	Launch templates ▾	Actions ▾	Create Auto Scaling group
Search your Auto Scaling groups		< 1 > ⚙️				
<input type="checkbox"/>	Name ↴	Launch template/configuration	Instances	Status	Desired capacity	Min
<input type="checkbox"/>	nodes-ap-southeast-1b.cloudservicesdemo.in	nodes-ap-southeast-1b.cloudservicesdem	1	-	1	1
<input type="checkbox"/>	control-plane-ap-southeast-1a.masters.cloudservicesdemo.in	control-plane-ap-southeast-1a.masters.cl	1	-	1	1
<input type="checkbox"/>	nodes-ap-southeast-1a.cloudservicesdemo.in	nodes-ap-southeast-1a.cloudservicesdem	1	-	1	1

25. Now if you want to see more then you can visit VPC and S3 for more information.

26. Now once you are done you can give this command and it will delete all the resources that are running.

```
kops delete cluster --name cloudservicesdemo.in --state=s3://demo-kops-bucket --yes
```