1. In the command line window, check the AWS CLI version:
2. aws --version

It should be an older version.

1. Download v2:
2. curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip" -o "awscliv2.zip"
3. Unzip the file:
4. unzip awscliv2.zip
5. See where the current AWS CLI is installed:
6. which aws

It should be /usr/bin/aws.

1. Update it:
2. sudo ./aws/install --bin-dir /usr/bin --install-dir /usr/bin/aws-cli --update
3. Check the version of AWS CLI:
4. aws --version

It should now be updated.

1. Configure the CLI:
2. aws configure
3. For AWS Access Key ID, paste in the access key ID you copied earlier.
4. For AWS Secret Access Key, paste in the secret access key you copied earlier.
5. For Default region name, enter us-east-1.
6. For Default output format, enter json.
7. Download kubectl:
8. curl -o kubectl https://amazon-eks.s3.us-west-2.amazonaws.com/1.16.8/2020-04-16/bin/linux/amd64/kubectl
9. Apply execute permissions to the binary:
10. chmod +x ./kubectl
11. Copy the binary to a directory in your path:
12. mkdir -p $HOME/bin && cp ./kubectl $HOME/bin/kubectl && export PATH=$PATH:$HOME/bin
13. Ensure kubectl is installed:
14. kubectl version --short --client
15. Download eksctl:
16. curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl\_$(uname -s)\_amd64.tar.gz" | tar xz -C /tmp
17. Move the extracted binary to /usr/bin:
18. sudo mv /tmp/eksctl /usr/bin
19. Get the version of eksctl:
20. eksctl version
21. See the options with eksctl:
22. eksctl help

**Provision an EKS Cluster**

1. Provision an EKS cluster with three worker nodes in us-east-1:
2. eksctl create cluster --name dev --region us-east-1 --nodegroup-name standard-workers --node-type t3.medium --nodes 3 --nodes-min 1 --nodes-max 4 --managed

If your EKS resources can't be deployed due to AWS capacity issues, delete your eksctl-dev-cluster CloudFormation stack and retry the command using the --zones parameter and suggested availability zones from the CREATE\_FAILED message:

AWS::EKS::Cluster/ControlPlane: CREATE\_FAILED – "Resource handler returned message: \"Cannot create cluster 'dev' because us-east-1e, the targeted availability zone, does not currently have sufficient capacity to support the cluster. Retry and choose from these availability zones: us-east-1a, us-east-1b, us-east-1c, us-east-1d, us-east-1f (Service: Eks, Status Code: 400, Request ID: 21e7e4aa-17a5-4c79-a911-bf86c4e93373)\" (RequestToken: 18b731b0-92a1-a779-9a69-f61e90b97ee1, HandlerErrorCode: InvalidRequest)"

In this example, the --zones parameter was added using the us-east-1a,us-east-1b,us-east-1c,us-east-1d,us-east-1f AZs from the message above:

eksctl create cluster --name dev --region us-east-1 --zones us-east-1a,us-east-1b,us-east-1c,us-east-1d,us-east-1f --nodegroup-name standard-workers --node-type t3.medium --nodes 3 --nodes-min 1 --nodes-max 4 --managed

It will take 10–15 minutes since it's provisioning the control plane and worker nodes, attaching the worker nodes to the control plane, and creating the VPC, security group, and Auto Scaling group.

1. In the AWS Management Console, navigate to CloudFormation and take a look at what’s going on there.
2. Select the eksctl-dev-cluster stack (this is our control plane).
3. Click **Events**, so you can see all the resources that are being created.
4. We should then see another new stack being created — this one is our node group.
5. Once both stacks are complete, navigate to **Elastic Kubernetes Service** > **Clusters**.
6. Click the listed cluster.
7. If you see a Your current user or role does not have access to Kubernetes objects on this EKS cluster message just ignore it, as it won't impact the next steps of the activity.
8. Click the **Compute** tab (under **Configuration**), and then click the listed node group. There, we'll see the Kubernetes version, instance type, status, etc.
9. Click **dev** in the breadcrumb navigation link at the top of the screen.
10. Click the **Networking** tab (under **Configuration**), where we'll see the VPC, subnets, etc.
11. Click the **Logging** tab (under **Configuration**), where we'll see the control plane logging info.
    * The control plane is abstracted — we can only interact with it using the command line utilities or the console. It’s not an EC2 instance we can log into and start running Linux commands on.
12. Navigate to **EC2** > **Instances**, where you should see the instances have been launched.
13. Close out of the existing CLI window, if you still have it open.
14. Select the original t2.micro instance, and click **Connect** at the top of the window.
15. In the **Connect to your instance** dialog, select **EC2 Instance Connect (browser-based SSH connection)**.
16. Click **Connect**.
17. In the CLI, check the cluster:
18. eksctl get cluster
19. Enable it to connect to our cluster:
20. aws eks update-kubeconfig --name dev --region us-east-1