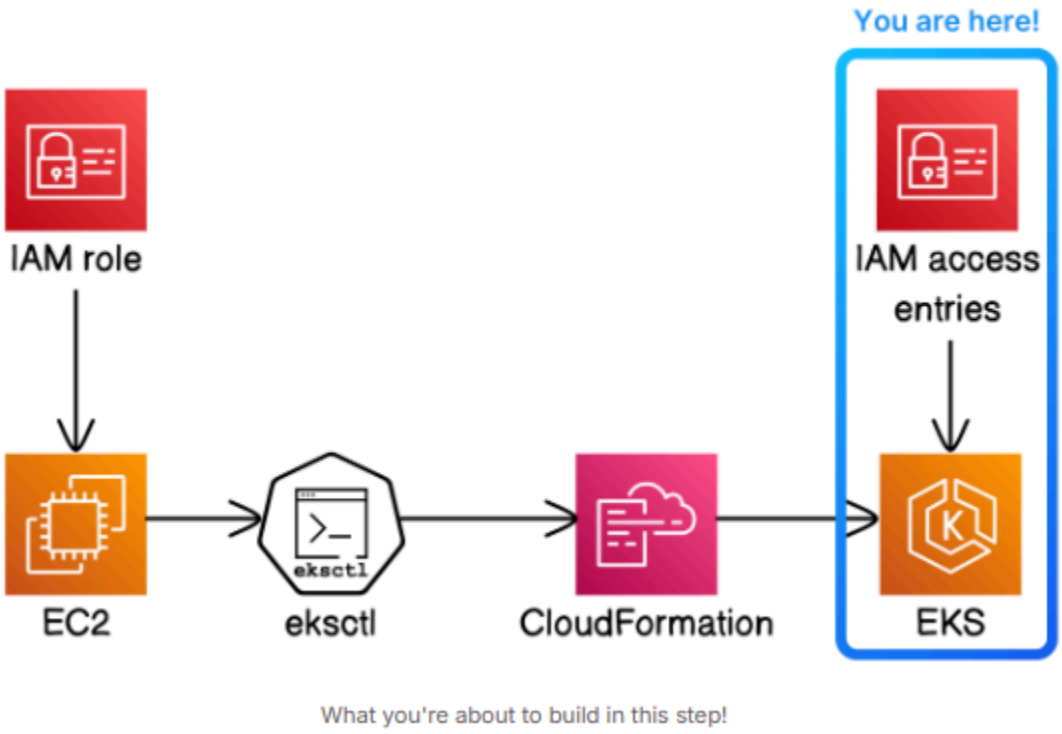
Building a Kubernetes Cluster



Launch an EC2 Instance

- Log into your AWS Management Console as your IAM Admin user.
- If you don't have an IAM user, check out Step #5 in our intro project on containers. We'd also recommend completing that project first to understand containers before diving into Kubernetes!
- Head to EC2 in your AWS Management Console.
- Select **Instances** at the left hand navigation bar.
- Check that you're using the **AWS Region** that's closest to you. Select the Region name at the top right corner of your console if you'd like to switch
- Select **Launch instances** at the top right corner of your console.
- Name your EC2 instance nextwork-eks-instance
- For the **Amazon Machine Image,** select **Amazon Linux 2023 AMI.**
- For the **Instance type,** select **t2.micro.**
- For the **Key pair (login)** panel, select **Proceed without a key pair (not recommended).**
- We won't change anything in the **Networking** section and keep the default security group.

- Notice that the security group allows SSH traffic from anywhere (0.0.0.0/0) which isn't the best security practice, but we'll leave this to make connecting via SSH easier later on.
- Select **Launch instances**.

Connect to Your Instance

Now that our EC2 instance is up and running, let's connect to it using EC2 Instance Connect.

Select the instance you just launched.

- Select the checkbox next to **nextwork-eks-instance.**
- Select the **Connect** button.

Launch an EKS cluster (and get an error)

Now for the main event – creating our Kubernetes cluster with **Amazon EKS**!

We'll use a handy command-line tool called **eksctl** to create a cluster within your EC2 instance's terminal.

Let's try creating a cluster right away.

- In your EC2 Instance Connect window, run this command:

```
eksctl create cluster \

--name nextwork-eks-cluster \

--nodegroup-name nextwork-nodegroup \

--node-type t2.micro \

--nodes 3 \

--nodes-min 1 \

--nodes-max 3 \

--version 1.31
```

... you'll likely see an error message saying something like eksctl not found

```
[ec2-user@ip-172-31-25-220 ~]$ eksctl create cluster \
--name nextwork-eks-cluster \
--region ${AWS_REGION} \
--nodegroup-name nextwork-nodegroup \
--node-type t3.micro \
--nodes 3 \
--nodes-min 1 \
--nodes-max 4
bash: eksctl: command not found
```

Don't worry, this is expected! It means we haven't installed **eksctl** yet.

Install eksctl

- Run these commands in your terminal:

curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_$(uname -s)_amd64.tar.gz" | tar xz -C /tmp

sudo mv -v /tmp/eksctl /usr/local/bin

Check that eksctl is installed correctly by running *eksctl version*. You should see the version number printed in the terminal.

Launch an EKS cluster (for real this time)

Now that eksctl is all installed, let's give launching an Kubernetes cluster on EKS another go. We'll also learn about giving your EC2 instance the permission to run AWS commands!

Now that eksctl is installed, let's try creating our cluster again.

- Run this command:

eksctl create cluster \

--name nextwork-eks-cluster \

--nodegroup-name nextwork-nodegroup \

--node-type t2.micro \

```
--nodes 3 \

--nodes-min 1 \

--nodes-max 3 \

--version 1.31 \

--region [YOUR-REGION]
```

Make sure to replace [YOUR-REGION] in the last line of the command with your AWS region code, e.g. **us-west-2**. Make sure the region you pick is same region where you launched your EC2 instance

Create an IAM role for your EC2 instance

Let's resolve this permissions error by setting up a new **Role** in AWS IAM.

- In a new tab, head to your AWS IAM console.
- Select **Roles** from the left hand sidebar.
- Select **Create role.**
- Under **Trusted entity type,** select **AWS service** to tell AWS that we're setting up this role for a AWS serice (Amazon EC2).
- Under **Use case,** select **EC2.**
- Select **Next.**
- Under **Permissions policies,** we'll grant our EC2 instance **AdministratorAccess.**
- Make sure the **AdministratorAccess** option is checked, and select **Next**.
- Let's give this role a straightforward name - nextwork-eks-instance-role
- Enter a short description:
- Grants an EC2 instance AdministratorAccess to my AWS account. Created during NextWork's Kubernetes project.
- Select **Create role.**
-

Attach IAM role to EC2 instance

Great! Your new role is born. Now we'll add this role to our EC2 instance.

- Head back to the **Amazon EC2** console.
- Select **Instances** from the left hand sidebar.

- Select the checkbox next to your **nextwork-eks-instance** EC2 instance.
- Select the **Actions** dropdown, and then **Security** -> **Modify IAM role.**
- Under **IAM role,** select your new nextwork-eks-instance-role role.
- Select **Update IAM role**.

Create your EKS cluster agains (third time lucky)!

- Still in your browser, head back to your **EC2 Instance Connect** tab.
- Let's run the command to kick off our Kubernetes cluster one more time...

  eksctl create cluster \

  --name nextwork-eks-cluster \

  --nodegroup-name nextwork-nodegroup \

  --node-type t2.micro \

  --nodes 3 \

  --nodes-min 1 \

  --nodes-max 3 \

  --version 1.31 \

  --region [YOUR-REGION]

Make sure to replace [YOUR-REGION] in the last line of the command with your AWS region code, e.g. **us-west-2**. Make sure the region you pick is same region where you launched your EC2 instance.

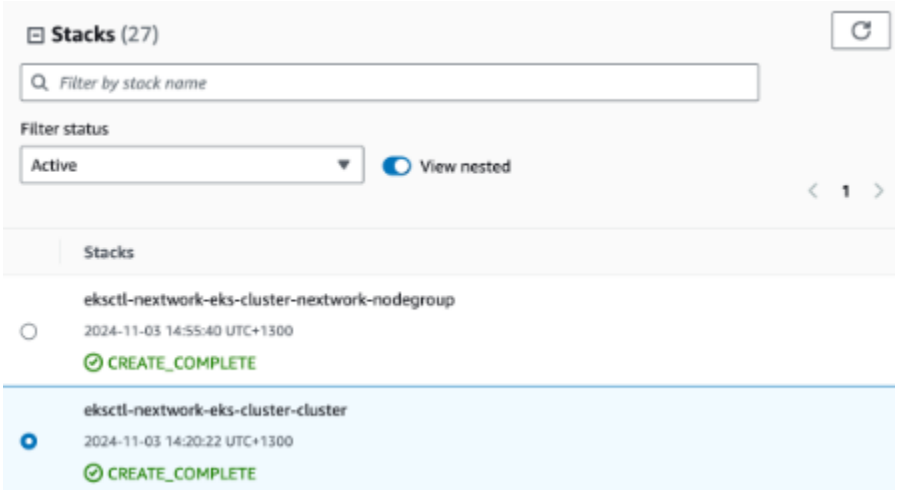
⏰ **Check the time on your computer now.** Don't forget that EKS charges $0.10 for every hour your cluster is running.

Creating a cluster can take 15-20 minutes, so grab a drink and read on to know more about Kubernetes while we wait.

In a new tab, head to the CloudFormation console.

- In the **Stacks** page, notice that there is a new stack in progress! The stack should be called **eksctl-nextwork-eks-cluster-cluster.**
- Select the stack, and track the **Events** tab.
- Now select the **Resources** tab.

- Woah! Lots of resources are getting created here.
- You might also notice a NEW stack pop up in the Stacks page. This should pop up 10-12 minutes from the time you ran the create cluster command
- Select the nodegroup stack, and open up the **Events** tab to get a preview of what being deployed.
- Wait until the first stack i.e. **eksctl-nextwork-eks-cluster-cluster** is in **CREATE_COMPLETE** status.



Accessing the EKS Console

- In a new tab, open up the EKS console.
- Select the new **nextwork-eks-cluster** cluster you just created.
- Welcome to your EKS cluster's dedicated page!
- Select the **Compute** tab.
-
- Scroll down to the **Node groups** panel - aha, it's getting created!
- If it's been some time since you ran your create cluster command, you might even notice that it's already in **Created** status.
- Scroll back up to the top of your cluster's page. Notice that there are two banners in blue.

Set up your own access to your cluster

- Select **Create access entry** at the blue banner at the top of the page.

Nice work, we can now set up an IAM access entry.

- Under **IAM Principal**, select your IAM user's ARN. Double check your IAM Admin's name at the top right corner and make sure it matches your ARN.

- Select **Next**.
- Under **Policy name,** select AmazonEKSClusterAdminPolicy.
- Select **Add policy.**
- Select **Next**.
- Select **Create**.
- Head back to your cluster's page
- Select the refresh button on the console. You should now see your nodes listed under your node group.
- Nice work creating your very first Kubernetes cluster with Amazon EKS!
- Well done on setting up the basics of Amazon EKS. You'll go through this process of creating an EKS cluster **again** in the next project of this Kubernetes series.
-