



Launch a Kubernetes Cluster

ME

Melvin J Bonner

Nodes (3) Info						
Node name	Instance type	Compute	Managed by	Created	Status	
ip-192-168-10-142.ec2.internal	t2.micro	Node group	nextwork-nodegroup	14 hours ago	Ready	
ip-192-168-2-78.ec2.internal	t2.micro	Node group	nextwork-nodegroup	14 hours ago	Ready	
ip-192-168-63-150.ec2.internal	t2.micro	Node group	nextwork-nodegroup	14 hours ago	Ready	

Introducing Today's Project!

In this project, I will deploy my first Kubernetes cluster because I want to learn what it does and why people love it.

What is Amazon EKS?

Amazon EKS is a managed service that makes it easy to run Kubernetes on AWS without needing to maintain your own Kubernetes control plane. I was able to use Amazon EKS to set up and run my Kubernetes clusters and integrate Kubernetes with other AWS services.

One thing I didn't expect

The one thing I didn't expect was the replacement of the terminated EC2 instances.

This project took me...

This project took me 2 hours. The part that took the longest was adding the access policy.

What is Kubernetes?

Kubernetes is a container orchestration platform, it coordinates containers so they're running smoothly across all your servers. It makes sure all containers are running where they should, scales containers automatically to meet demand levels, and even restarts containers if something crashes. Companies and developers use Kubernetes for keeping large, container-based applications steady and easy to scale with traffic.

I used eksctl to create a cluster within my EC2 instance's terminal. The create cluster command I ran defined the node group and the instance type.

I initially ran into two errors while using eksctl. The first one was because eksctl was not installed. The second one was because my EC2 instance didn't have the permission to create an EKS cluster.

```
[ec2-user@ip-172-31-93-118 ~]$ ^[[200~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
bash: $'\E[200~eksctl': command not found
[ec2-user@ip-172-31-93-118 ~]$ ~
bash: /home/ec2-user: Is a directory
[ec2-user@ip-172-31-93-118 ~]$ █
```

eksctl and CloudFormation

Cloudformation helped create my EKS cluster because eksctl sets up a CloudFormation stack to automate the creation of all the necessary resources for the EKS cluster. It created VPC resources to make it work for a Kubernetes cluster. eksctl creates a whole new VPC for us to let us start fresh.

There was also a second Cloudformation stack for the note group. The difference between a cluster and a node group, is that a cluster is made up of nodes and a node group lets you manage your nodes by grouping them together.

Stacks (2)			
Stack name	Status	Created time	Description
eksctl-nextwork-eks-cluster-nodegroup-nextwork-nodegroup	CREATE_COMPLETE	2025-07-17 23:53:53 UTC-0400	EKS Managed Nodes (SSH access: false) [created by eksctl]
eksctl-nextwork-eks-cluster-cluster	CREATE_COMPLETE	2025-07-17 23:42:49 UTC-0400	EKS cluster (dedicated VPC: true, dedicated IAM: true) [created and managed by eksctl]

The EKS console

I had to create an IAM access entry in order to be able to see and control all parts of my EKS cluster, including all the nodes inside. An access entry connects AWS IAM users with Kubernetes' Role-Based Access Control (RBAC), which is Kubernetes' system to manage access inside a cluster. I set up the access entry under IAM Principal, by selecting my IAM user's ARN and making sure my IAM Admin's name at the top right corner matched my ARN.

It took me 2 hours to create my cluster. Since I'll create this cluster again in the next project of this series, this process would be sped up if I didn't miss the add policy button when I initially tried to add the access policy. .

ME

Melvin J Bonner
NextWork Student

nextwork.org

Nodes (3) Info						
<input type="text"/> Filter Nodes by property or value						
Node name	▲ Instance type	▼ Compute	▼ Managed by	▼ Created	▼ Status	▼
ip-192-168-10-142.ec2.internal	t2.micro	Node group	nextwork-nodegroup	14 hours ago	Ready	
ip-192-168-2-78.ec2.internal	t2.micro	Node group	nextwork-nodegroup	14 hours ago	Ready	
ip-192-168-63-130.ec2.internal	t2.micro	Node group	nextwork-nodegroup	14 hours ago	Ready	

EXTRA: Deleting nodes

Did you know you can find your EKS cluster's nodes in Amazon EC2? This is because when you create nodes in a Kubernetes cluster on AWS, each node is actually an EC2 instance. Kubernetes uses a generic term (node) because different cloud platforms use different cloud resources to be the node; in AWS, we use EC2 instances as our nodes.

Desired size is the number of nodes I want running in my node group. Minimum size is the minimum number of nodes my group needs to have to keep my app available at all times (even in low-demand periods). Maximum size is the maximum number of nodes that I will allow inside my group. Minimum and maximum sizes are helpful for automatically adjusting the number of nodes based on the demands of my application. I can always edit these settings if I want my node group to be bigger or smaller!

When I deleted my EC2 instances, new instances were launched to replace the terminated ones. This is because my managed node group uses an Auto Scaling Group behind the scenes. AWS ensures that 3 EC2 instances (nodes) are always running as per desiredSize: 3.

ME

Melvin J Bonner

NextWork Student

nextwork.org

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...
nextwork-eks-cluster-nextwork-nodeg...	i-04edd87d011b236ac	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b	ec2-54-89-186-133.co...	54.89.186.133
<input checked="" type="checkbox"/> nextwork-eks-cluster-nextwork-nodeg...	i-03a4f230cdff18f07f	Terminated	t2.micro	-	View alarms +	us-east-1b	-	-
<input checked="" type="checkbox"/> nextwork-eks-cluster-nextwork-nodeg...	i-0cb7d3aebbb912e23	Terminated	t2.micro	-	View alarms +	us-east-1b	-	-
nextwork-eks-cluster-nextwork-nodeg...	i-0be1779b22a6102f4	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b	ec2-98-81-123-38.com...	98.81.123.58
nextwork-eks-cluster-nextwork-nodeg...	i-000935bc428106955	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-100-24-60-255.co...	100.24.60.255
nextwork-eks-instance	i-0b64f8ecc10143d03	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-34-226-138-205.co...	34.226.138.205
<input checked="" type="checkbox"/> nextwork-eks-cluster-nextwork-nodeg...	i-047b1ed8e8842c956	Terminated	t2.micro	-	View alarms +	us-east-1a	-	-



nextwork.org

The place to learn & showcase your skills

Check out nextwork.org for more projects

