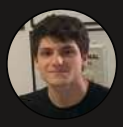
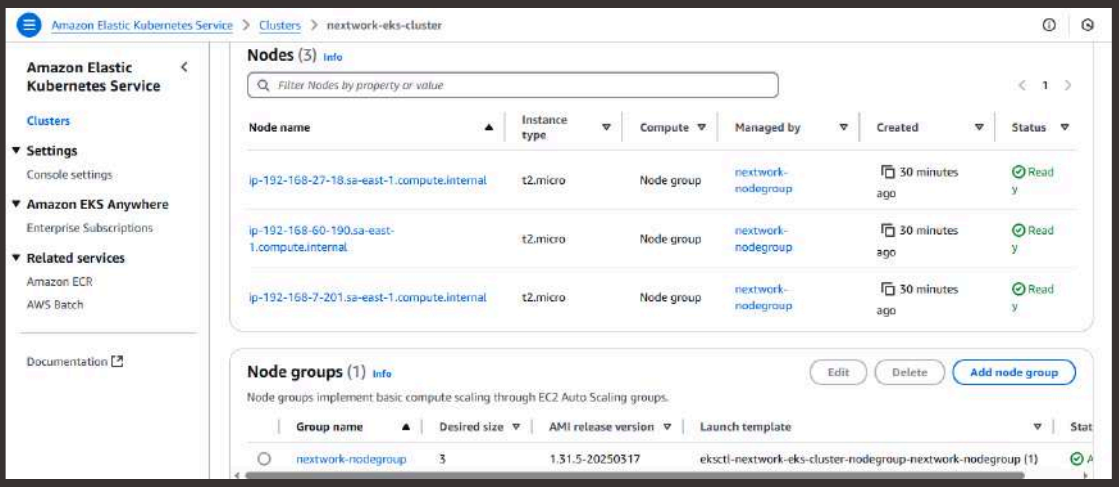
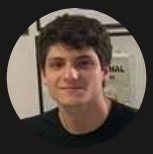
nextwork.org

Launch a Kubernetes Cluster



Nicolás Aversa





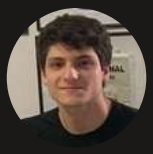
Nicolás Aversa
NextWork Student

Introducing Today's Project!

In this project, I will deploy a Kubernetes cluster using Amazon EKS to learn how to manage containerized applications in the cloud. I'll use EC2, CloudFormation, and IAM to create and access the cluster, and test its resilience.

What is Amazon EKS?

Amazon EKS (Elastic Kubernetes Service) is a managed service that simplifies running Kubernetes on AWS. It handles the setup, scaling, and maintenance of Kubernetes control planes, allowing you to focus on deploying and managing containerized applications. I used it to create a Kubernetes cluster with eksctl, set up a node group with scaling limits (min 1, max 3 nodes), automate infrastructure using CloudFormation, manage access with IAM roles, and test resilience by deleting nodes to observe Kubernetes' self-healing.



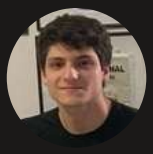
Nicolás Aversa
NextWork Student

One thing I didn't expect

One thing I didn’t expect in today's project was how straightforward it could be to create and manage Kubernetes clusters using Amazon EKS and eksctl. I was surprised by how much of the heavy lifting, like networking and resource provisioning, was automated by CloudFormation behind the scenes. It made the process much smoother than I anticipated, especially for someone new to Kubernetes!

This project took me...

This project took me 90 minutes. The part that took the longest was resolving the errors when creating the cluster with eksctl, as I needed to download it, and then set up the correct IAM roles and permissions to ensure everything worked smoothly.



Nicolás Aversa
NextWork Student

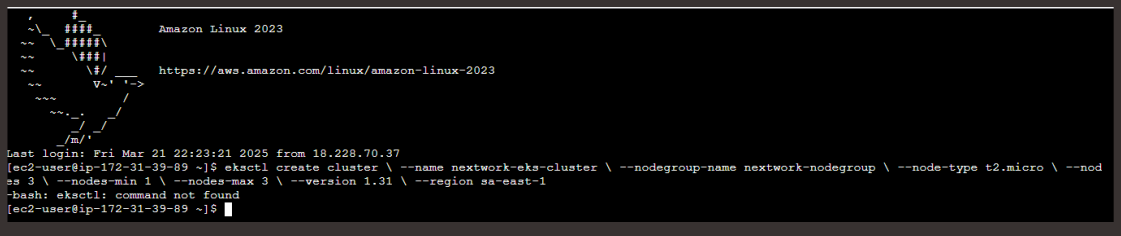
NextWork.org

What is Kubernetes?

Kubernetes is a container orchestration platform, which is a fancy way to say that it coordinates containers so they're running smoothly across all your servers. It makes sure all your 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 to create and manage containers; instead of doing it manually. Managing hundreds or thousands of containers manually would be a huge amount of work and hard to get right all the time. Kubernetes takes care of all these tasks automatically!

I used eksctl to create en EKS cluster. The create cluster command I ran defined defined the cluster's name, nodegroup name, node type, number of nodes (min and max included), version and AWS region.

I initially ran into two errors while using eksctl. The first one was because I didn't have eksctl installed. The second one was because the default IAM role for my EC2 instance lacked permissions to create an EKS cluster.

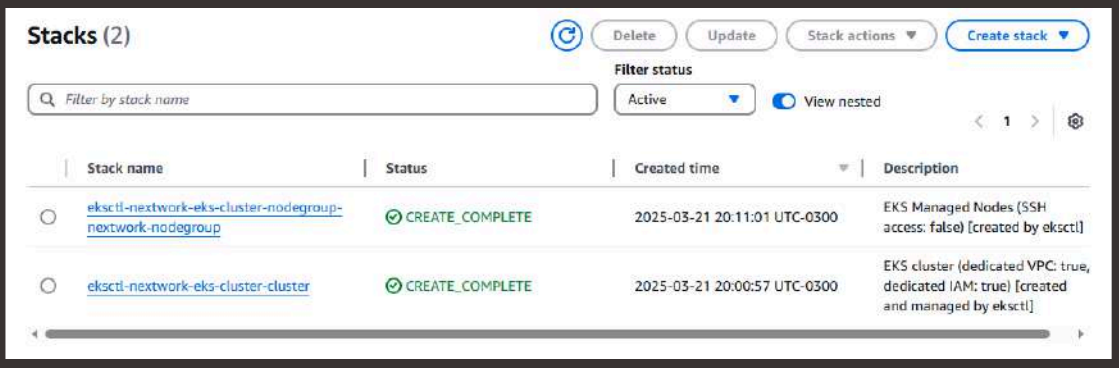


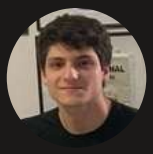
Nicolás Aversa
NextWork Student

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 because it sets up a private, secure network for the containers to connect with each other and the internet while keeping the app private.

There was also a second CloudFormation stack for my node group, which is a group of EC2 instances that will run my containers. The difference between a cluster and node group is that a cluster is the entire Kubernetes environment (including the control plane and all nodes), while a node group is a subset of nodes within the cluster, grouped together with shared configurations for easier management.





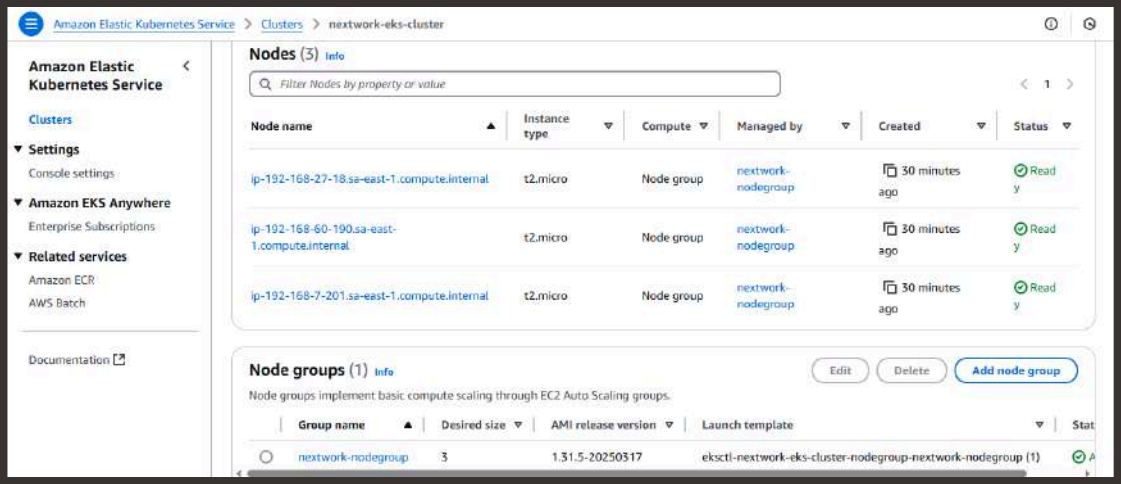
Nicolás Aversa
NextWork Student

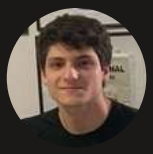
The EKS console

I had to create an IAM access entry in order to get access to the nodes in my EKS cluster. An access entry is how we connect AWS IAM users with Kubernetes' Role-Based Access Control (RBAC), which is Kubernetes' system to manage access inside a cluster. I set it up by picking my IAM user's ARN and adding an AccessPolicy called AmazonEKSClusterAdminPolicy, choosing Cluster as the scope. This policy gives you full administrative rights over your EKS clusters.

It took 30 minutes to create my cluster. Since I'll create this cluster again in the next project of this series, maybe this process could be sped up if I ensure the IAM role and permissions are set up in advance.

Nicolás Aversa
NextWork Student





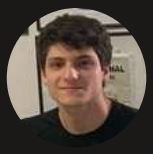
Nicolás Aversa
NextWork Student

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.

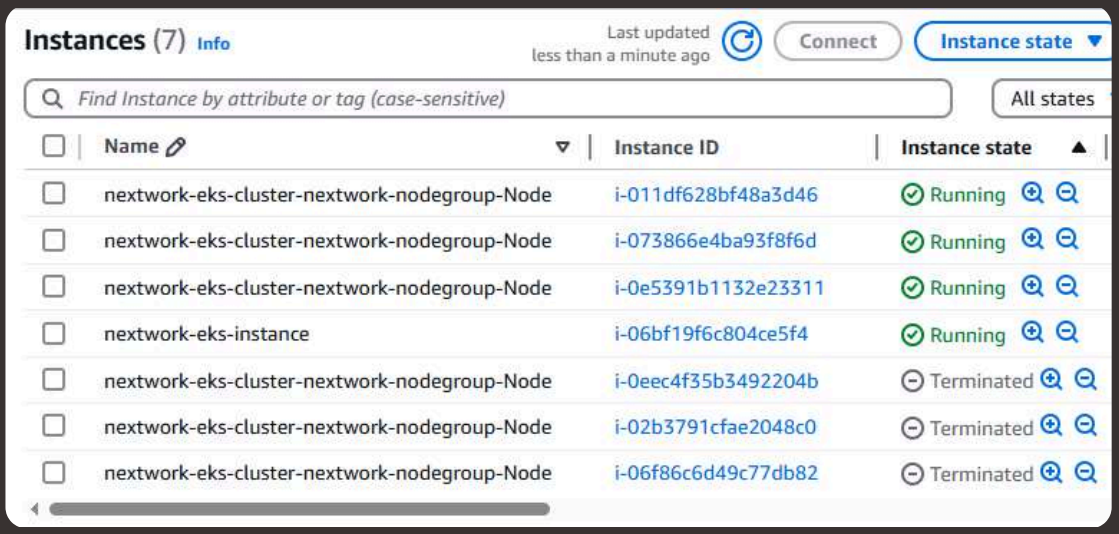
Desired size is the number of nodes you want running in your node group. Mininum and maximum sizes are helpful for ensuring your application remains available and scalable. The minimum size ensures your app stays operational during low-demand periods by maintaining a baseline number of nodes, while the maximum size allows your node group to scale up during high-demand periods, preventing overloading. Together, these settings help Kubernetes automatically adjust the number of nodes to match your app's needs, balancing performance and cost.

When I deleted my EC2 instances, new instances were launched to replace the terminated ones. This is because Kubernetes automatically detects the change and launches new nodes to keep the cluster running at its desired state (3 nodes in my case).



Nicolás Aversa
NextWork Student

NextWork.org



NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for more projects