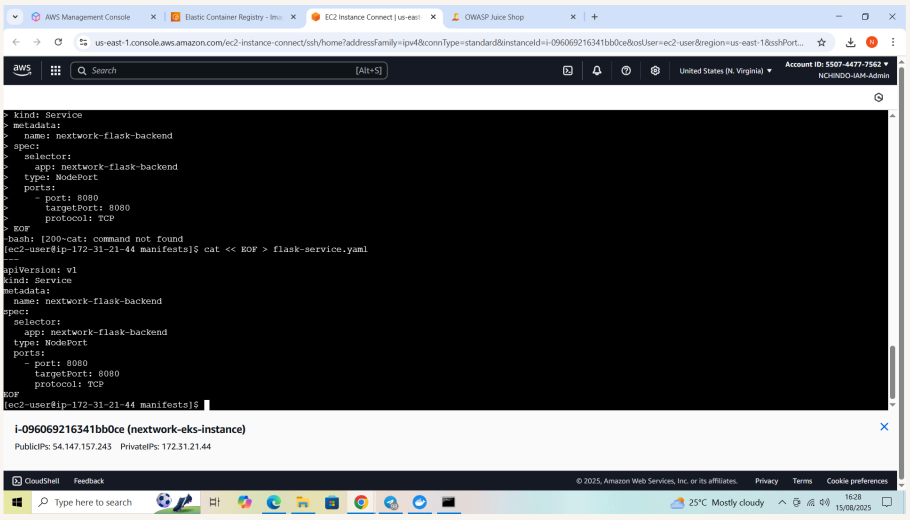
nextwork.org

Create Kubernetes Manifests

Nchindo Boris



Introducing Today's Project!

In this project, I will; - Set up a Deployment manifest that tells Kubernetes how to deploy my backend. - Set up a Service manifest that tells Kubernetes how to expose your backend to my users. these key files called manifests that tell Kubernetes how to deploy and manage my containerized backend!

Tools and concepts

I used tools like; - Amazon EKS - Amazon ECR - Git & GitHub - Amazon EC2 Key concepts include; - Build and push a container image of the backend of my app. - Write a Deployment manifests to tell Kubernetes how to manage and scale my backend app. - Write a Service manifests to tell Kubernetes how to expose my backend app to handle traffic.

Project reflection

I did this project today to deepen my understanding of deploying containerised applications using Kubernetes on AWS

This project took me approximately 1 hour to complete. The most challenging part was understanding the structure and purpose of the Kubernetes manifests, especially how Deployment and Service work together.

Project Set Up

Kubernetes cluster

To set up today's project, I launched a Kubernetes cluster. Steps I took to do this included; - First, I logged into the AWS Management Console using my IAM Admin account. I then launched an EC2 instance and named it nextwork-eks-instance. - After connecting to the instance via EC2 Instance Connect, I installed ekscti, a tool that makes it easier to create AWS EKS clusters. - Next, I attached an IAM role called nextwork-eks-instance-role to the Instance and gave it AdministratorAccess so it could create all the necessary AWS resources. - Finally, I ran a single eksctl command to create the Kubernetes cluster named nextwork-eks-cluster, along with a node group using small t2.micro instances.

Backend code

I retrieved the backend that I plan to deploy by; - First installing Git on my EC2 instance, since it wasn't pre-Installed. -After setting up Git and configuring my user details, I copied the HTTPS URL from the GitHub repository and ran the git clone command. This downloaded a full copy of the nextwork-flask-backend repository onto my EC2 instance, giving me access to all the backend files I need for deployment. The backend is the "brain" of an application. It's where your app processes user requests and stores and retrieves data.

Container image

Once I cloned the backend code, I built a container image because Kubernetes doesn't run raw code; it runs containers that are created from container images. I used Docker to do this

placeholder

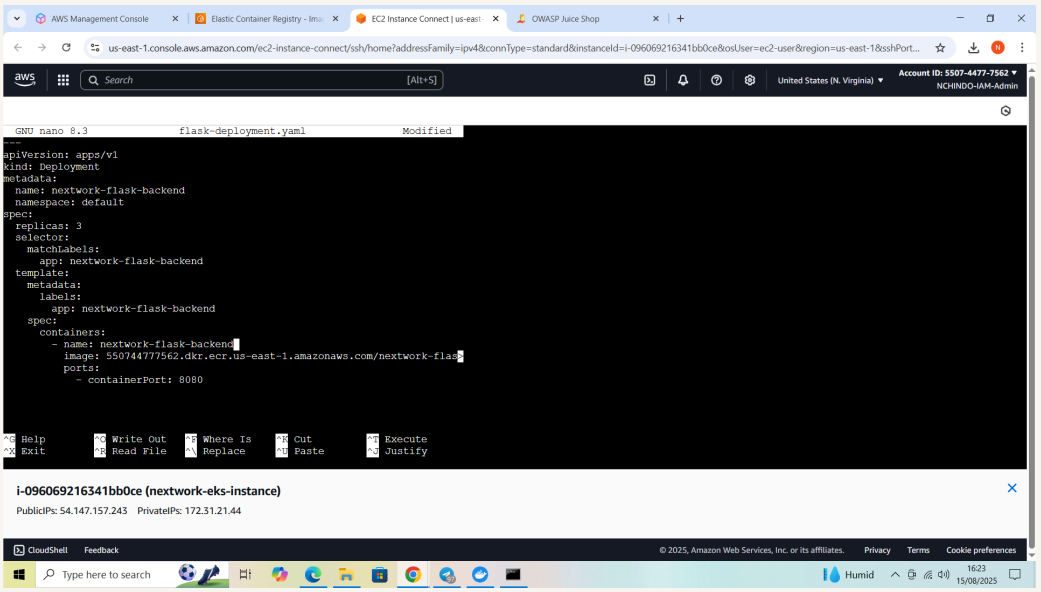
I also pushed the container image to a container registry, because ECR lets

To push the image to ECR, I created a new repository using the AWS CLI, authenticated Docker with ECR, tagged my image with the latest tag, and pushed it to the ECR repository. This setup ensures my containerised backend is stored in a reliable, AWS native registry that integrates seamlessly with EKS.

Manifest files

Kubernetes manifests are simple files that tell Kubernetes how to run your app. They're helpful because they make it easy to describe things like which container to use, how many copies of your app to run, and how to expose it to users. Instead of setting everything up by hand every time, Kubernetes just reads the manifest and does it for you, quickly and consistently.

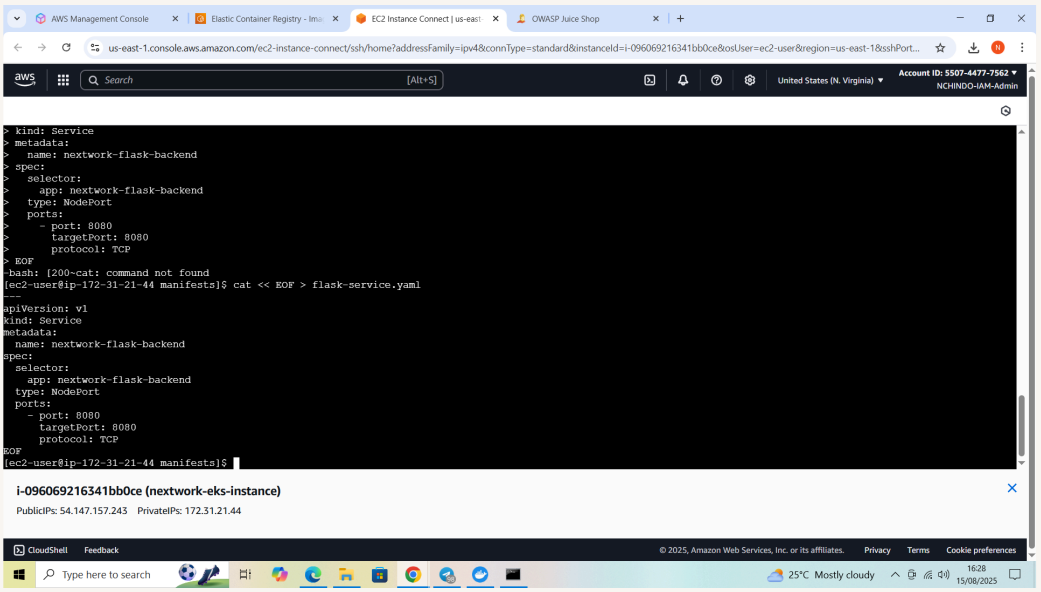
A Kubernetes deployment manages how many copies of my app should run, where they should run, and how to keep them running if something goes wrong. The container image URL in my Deployment manifest tells Kubernetes exactly which version of my app to pull from Amazon ECR, so it knows what to run inside each container across the cluster.


GNU nano 8.3 flask-deployment.yaml Modified
apiVersion: apps/v1
kind: Deployment
metadata:
name: nextwork-flask-backend
namespace: default
spec:
replicas: 3
selector:
matchLabels:
app: nextwork-flask-backend
template:
metadata:
labels:
app: nextwork-flask-backend
spec:
containers:
- name: nextwork-flask-backend
image: 550744777562.dkr.ecr.us-east-1.amazonaws.com/nextwork-flas
ports:
- containerPort: 8080
i-096069216341bb0ce (nextwork-eks-instance)
PublicIPs: 54.147.157.243 PrivateIPs: 172.31.21.44

Service Manifest

A Kubernetes Service exposes your app so it can receive traffic, either from inside the cluster or from the outside world. You need a Service manifest to tell Kubernetes which pods to route traffic to, which ports to use, and how that traffic should reach your app. Without a Service, your app would be running, but no one could access it!

My Service manifest sets up a NodePort service that exposes my backend app on port 8080. It tells Kubernetes to route traffic to any pods labelled app: nextwork-flask-backend, using TCP protocol. This allows external traffic to reach my app through a specific port on the cluster's nodes, making the backend accessible from outside the Kubernetes cluster.


AWS Management Console
Elastic Container Registry - Ima
EC2 Instance Connect | us-east
OWASP Juice Shop
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?addressFamily=ipv4&connType=standard&instanceId=i-096069216341bb0ce&osUser=ec2-user&region=us-east-1&sshPort...
Search
[Alt+S]
United States (N. Virginia)
Account ID: 5507-4477-7562
NCHINDO-IAM-Admin
> kind: Service
> metadata:
>   name: nextwork-flask-backend
> spec:
>   selector:
>   app: nextwork-flask-backend
>   type: NodePort
>   ports:
>   - port: 8080
>   targetPort: 8080
>   protocol: TCP
> EOF
-bash: [200~cat: command not found
[ec2-user@ip-172-31-21-44 manifests]$ cat << EOF > flask-service.yaml
---
apiVersion: v1
kind: Service
metadata:
  name: nextwork-flask-backend
spec:
  selector:
    app: nextwork-flask-backend
  type: NodePort
  ports:
    - port: 8080
      targetPort: 8080
      protocol: TCP
EOF
[ec2-user@ip-172-31-21-44 manifests]$
i-096069216341bb0ce (nextwork-eks-instance)
PublicIPs: 54.147.157.243 PrivateIPs: 172.31.21.44
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
25°C Mostly cloudy
16:28
15/08/2025

The place to learn & showcase your skills

Check out nextwork.org for more projects