THISARA KANDAGE

UNDERGRADUATE - SLIIT

E-mail LinkedIn GitHub Website

# Deploying the 2048 Game on Amazon EKS using Kubernetes

## Introduction

This project involved deploying the classic 2048 puzzle game on a containerized infrastructure using Amazon Elastic Kubernetes Service (EKS). The goal was to gain hands-on experience with container orchestration, AWS infrastructure management, and Kubernetes deployment practices in a cloud-native environment.

To begin, an EKS cluster was created using the AWS Console, along with a managed EC2 node group to serve as worker nodes. Access to the cluster was configured using the AWS CLI and kubectl, allowing interaction with the Kubernetes API server. A Kubernetes Pod definition (2048-pod.yml) was written using the public Docker image blackicebird/2048 and deployed via kubectl apply.

Upon deployment, the pod remained in Pending state due to the worker node being marked NotReady. Investigation using kubectl describe node revealed that the issue was caused by an uninitialized CNI plugin, which prevented the network from being ready. To resolve this, the AWS VPC CNI plugin was installed using the appropriate Kubernetes manifest. Once the aws-node DaemonSet started running, the node status changed to Ready, and the pod was scheduled successfully.

Finally, the application was verified through a browser by accessing the public IP or service URL, confirming that the game was running as expected. Throughout the process, detailed troubleshooting was performed using CLI tools and AWS services.

THISARA KANDAGE

UNDERGRADUATE - SLIIT

E-mail LinkedIn GitHub Website

- **Create IAM role for EKS cluster**

## Use case

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

**Service or use case**

EKS

Choose a use case for the specified service.

**Use case**

- ○ EKS - Service
  Allows EKS to manage clusters on your behalf.
- ◉ EKS - Cluster
  Allows the cluster Kubernetes control plane to manage AWS resources on your behalf.

## Permissions policies (1) Info

The type of role that you selected requires the following policy.

| Policy name | Type |
|---|---|
| AmazonEKSClusterPolicy | AWS managed |

## Role details

**Role name**

Enter a meaningful name to identify this role.

eks-cluster-role-101

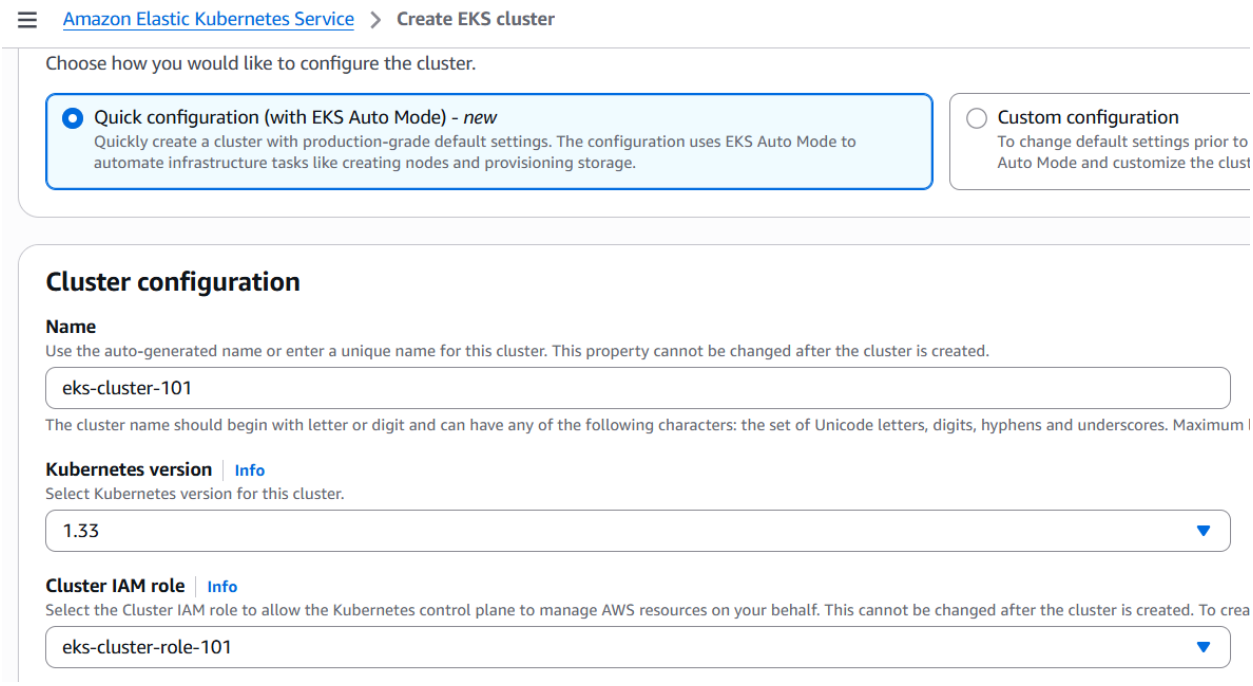
Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.

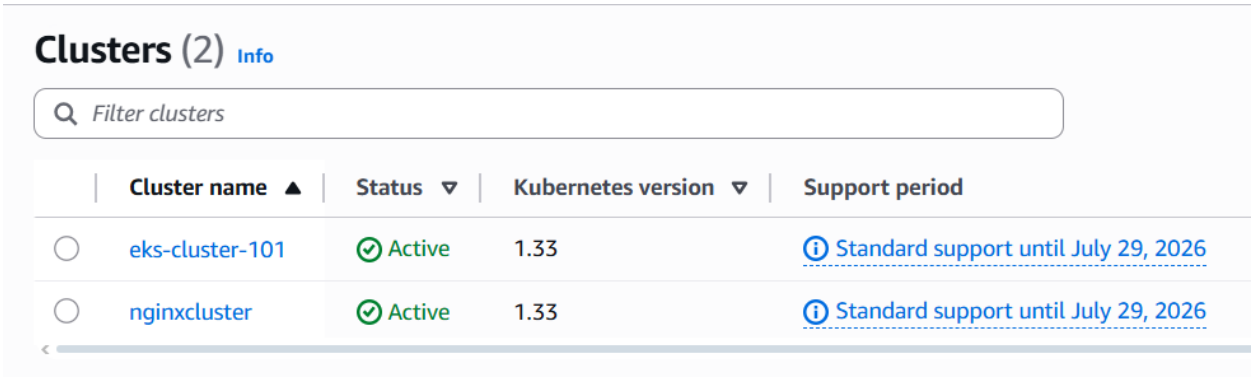
THISARA KANDAGE

UNDERGRADUATE - SLIIT

E-mail LinkedIn GitHub Website

- **Create EKS cluster**



THISARA KANDAGE

UNDERGRADUATE - SLIIT

E-mail LinkedIn GitHub Website

- **Create Security Groups(EC2)**

## Basic details

**Security group name** Info

eks-sec-grp-101

Name cannot be edited after creation.

**Description** Info

eks-sec-grp-101

**VPC** Info

vpc-0fad64e64b03d7441

**Inbound rules** Info

| Type Info | Protocol Info | Port range Info | Source Info | |
|---|---|---|---|---|
| SSH | TCP | 22 | Anyw… | 0.0.0.0/0 |
| HTTP | TCP | 80 | Anyw… | 0.0.0.0/0 |
| Custom TCP | TCP | 8080 | Anyw… | 0.0.0.0/0 |

Add rule

THISARA KANDAGE

UNDERGRADUATE - SLIIT

E-mail LinkedIn GitHub Website

## Setting up data plain Steps

- **Create new IAM role**

### Use case

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

**Service or use case**

EC2

### Role details

**Role name**

Enter a meaningful name to identify this role.

node-grp-role-101

Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.

**Permissions policy summary**

| Policy name | Type | Attached as |
|---|---|---|
| AmazonEC2ContainerRegistryReadOnly | AWS managed | Permissions policy |
| AmazonEKS_CNI_Policy | AWS managed | Permissions policy |
| AmazonEKSWorkerNodePolicy | AWS managed | Permissions policy |

THISARA KANDAGE

UNDERGRADUATE - SLIIT

E-mail LinkedIn GitHub Website

- **Add a node group**

## Node group configuration

These properties cannot be changed after the node group is created.

**Name**

Assign a unique name for this node group.

eks-node-grp-102

The node group name should begin with letter or digit and can have any of the following characters: the set of Unicode letters, digit

**Node IAM role** Info

Select the IAM role that will be used by the nodes. To create a new role, go to the IAM console.

node-grp-role-101

## Node group network configuration

These properties cannot be changed after the node group is created.

**Subnets** Info

Specify the subnets in your VPC where your nodes will run.To create a new subnet, go to the corresponding page in the VPC console.

*Select subnets*

subnet-002c66a4954d91454 us-east-1b 172.31.0.0/20

subnet-0c17a39f85eb6f160 us-east-1a 172.31.32.0/20

subnet-0ccba46079d1b1285 us-east-1c 172.31.80.0/20

**Configure remote access to nodes** Info

THISARA KANDAGE

UNDERGRADUATE - SLIIT

E-mail LinkedIn GitHub Website

## Clusters (3) Info

Filter clusters

| Cluster name | Status | Kubernetes version | Support period |
|---|---|---|---|
| eks-cluster-101 | Active | 1.33 | Standard support until July 29, 2026 |
| eks-node-grp-102 | Active | 1.33 | Standard support until July 29, 2026 |
| nginxcluster | Active | 1.33 | Standard support until July 29, 2026 |

**Authenticating the cluster by creating kubeconfig file**

- See my details

THISARA KANDAGE

UNDERGRADUATE - SLIIT

E-mail LinkedIn GitHub Website

- **Update the kubeconfig file**

```
~ $ rm.kube/config
-bash: rm.kube/config: No such file or directory
~ $
~ $ aws eks update-kubeconfig --region us-east-1 --name eks-cluster-101
Added new context arn:aws:eks:us-east-1:277707111634:cluster/eks-cluster-101 to /home/cloudshell-user/.kube/config
~ $
~ $
```

- **Create the config file for the port**

us-east-1 +

```
  GNU nano 8.3
apiVersion: v1
kind: Pod
metadata:
  name: 2048-pod
  labels:
    app: 2048-ws
spec:
  containers:
    - name: 2048-container
      image: blackicebird/2048
      ports:
        - containerPort: 80
```

- **Apply the config file**

```
~ $ kubectl apply -f 2048-pod.yml
pod/2048-pod created
~ $
~ $
```

THISARA KANDAGE
UNDERGRADUATE - SLIIT
E-mail LinkedIn GitHub Website

- **Create the Service witch will deploy**

```
us-east-1 +

~ $ kubectl get daemonsets -n kube-system
  GNU nano 8.3
apiVersion: v1
kind: Service
metadata:
  name: mygame-svc
spec:
  selector:
    app: 2048-ws
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
  type: LoadBalancer
```

- **Apply the changes**

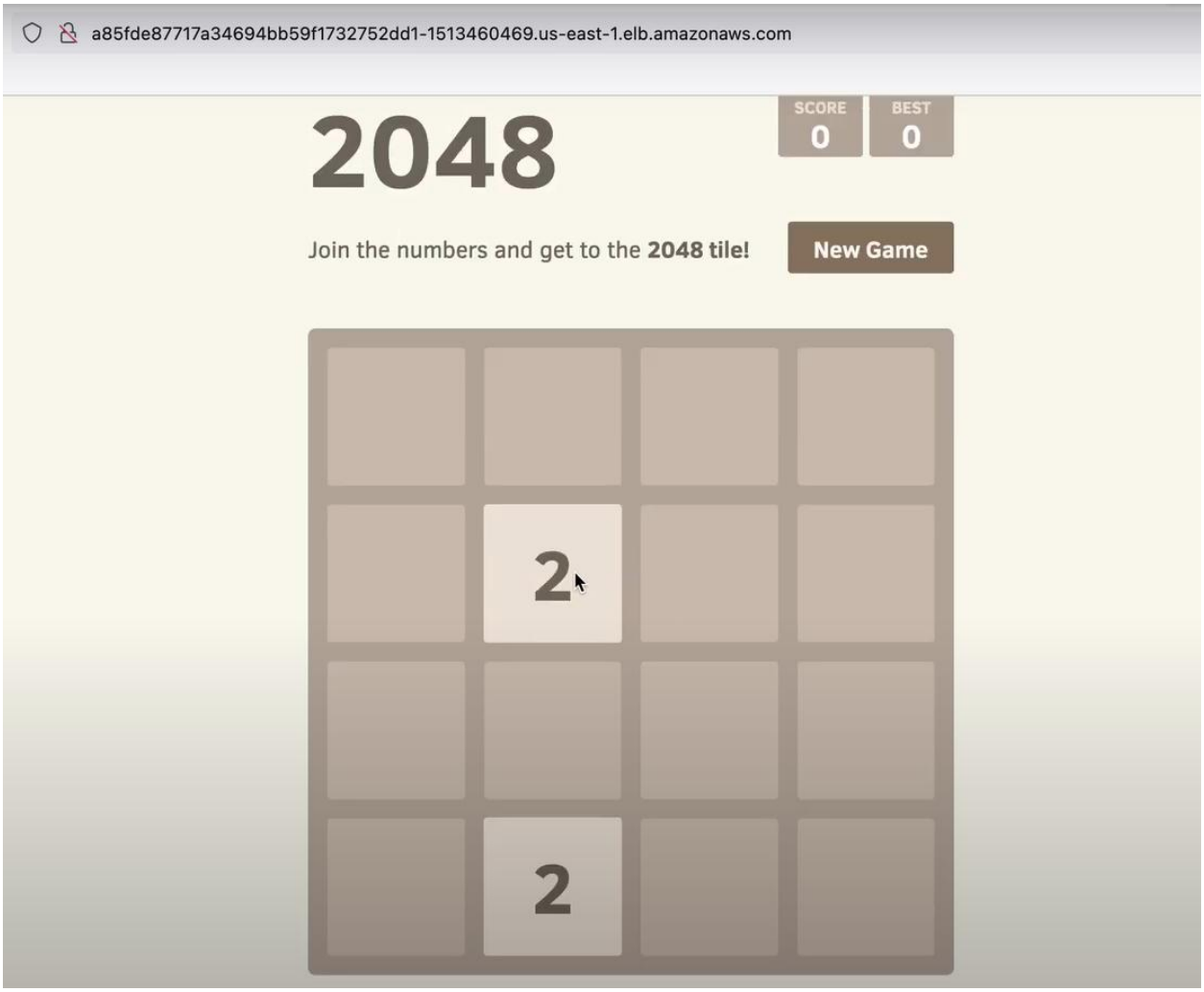
```
~ $ nano mygame-svc.yml
~ $ kubectl apply -f mygame-svc.yml
service/mygame-svc created
~ $
```

THISARA KANDAGE

UNDERGRADUATE - SLIIT

E-mail LinkedIn GitHub Website

- **Check the final output in web browser**



THISARA KANDAGE

UNDERGRADUATE - SLIIT

E-mail LinkedIn GitHub Website

# Summary

**This project improved my understanding of real-world DevOps and Kubernetes workflows. Key skills and concepts learned include:**

- **Setting up and configuring an Amazon EKS cluster using AWS Console and AWS CLI.**
- **Creating and managing EC2-based worker nodes as part of a node group.**
- **Using aws eks update-kubeconfig to securely access EKS from the CLI.**
- **Writing and applying Kubernetes YAML manifests to define application pods.**
- **Troubleshooting pod scheduling issues using kubectl describe.**
- **Understanding node states such as NotReady and how to resolve them.**
- **Installing and verifying the AWS VPC CNI plugin to restore Kubernetes networking.**
- **Monitoring and validating Kubernetes workloads using kubectl get and kubectl describe.**
- **Exposing a pod-based web application and verifying it via a browser.**
- **Strengthening foundational DevOps skills across cloud, infrastructure, and deployment layers.**