Creation of an infrastructure and deployment of a web application on the Elastic Kubernetes Service by AWS

By

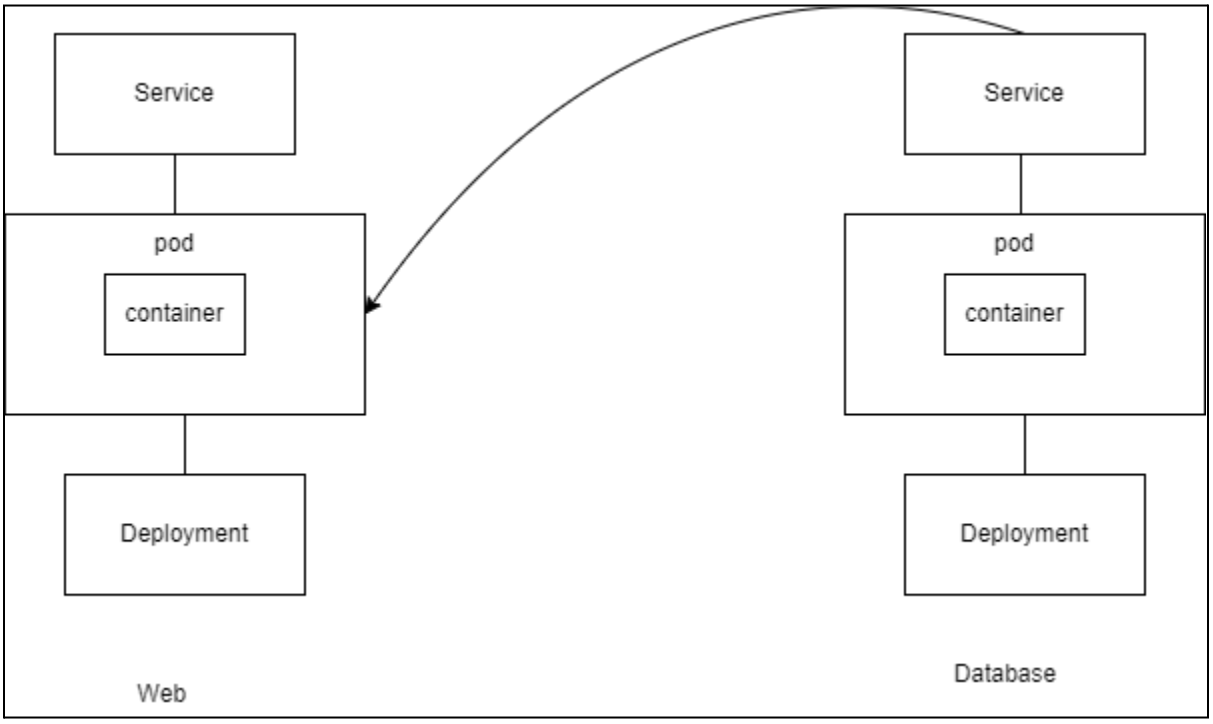
Akshat Parikh

Introduction

Amazon Elastic Kubernetes Service (Amazon EKS) is a managed Kubernetes service that makes it easy to run Kubernetes on AWS and on-premises. Kubernetes is an open-source system for automating the deployment, scaling, and management of containerized applications.

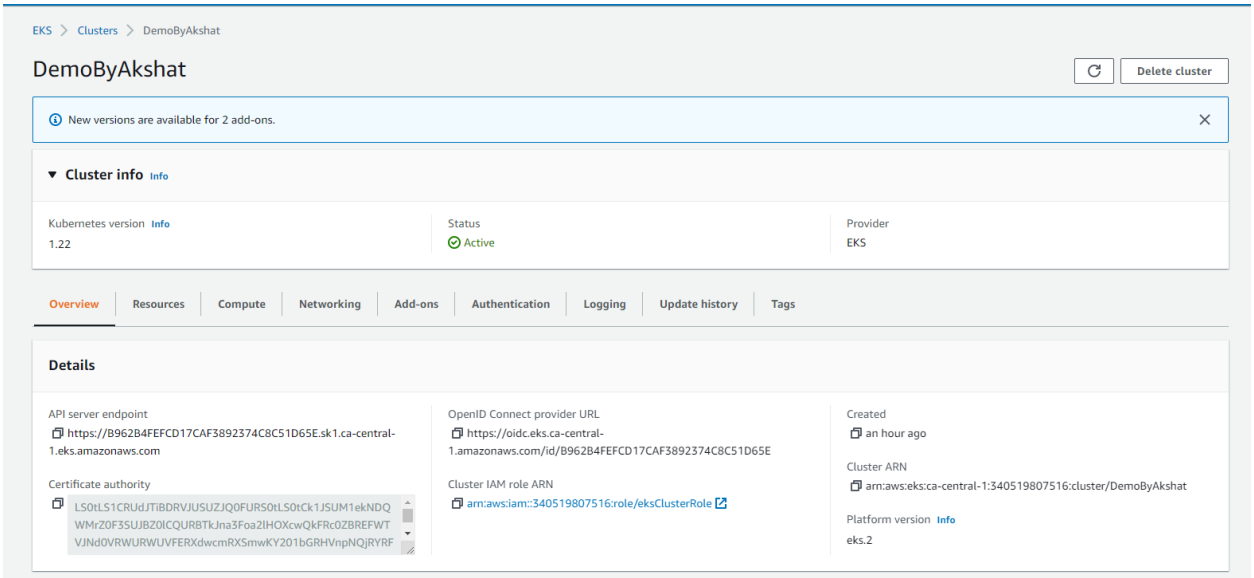
Application Infrastructure

- We have used a sample web application consisting of two services. One is a Web app, which will be our UI, and the second one is a database service.
- The database service is connected to a web service for data communication purposes.
- We have set up a public-facing load balancer to serve external requests.

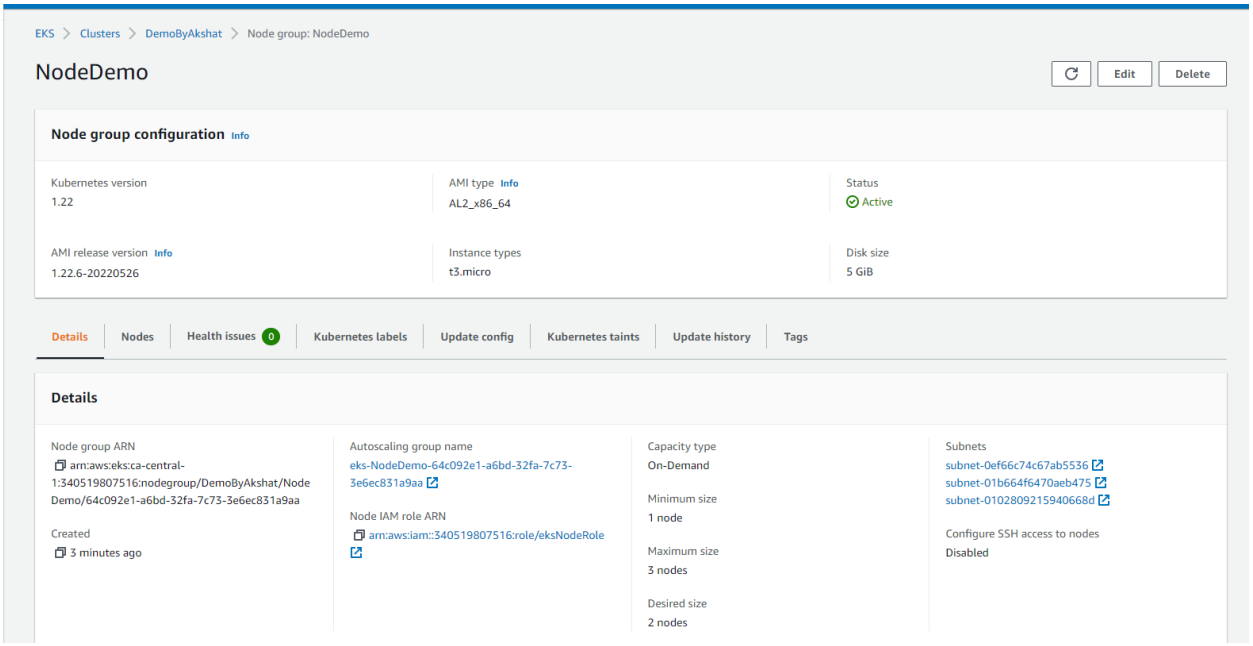


- The database part uses PostgresSQL and the web part uses Angular and .NET Core Web API.
- For each part, we have Kubernetes services and deployment.

Next, I created an EKS cluster for the application.



Afterwards, I created a node group having on-demand nodes with a minimum size of 1 node and a maximum size of 3 nodes. As shown below, the nodes are established and healthy.

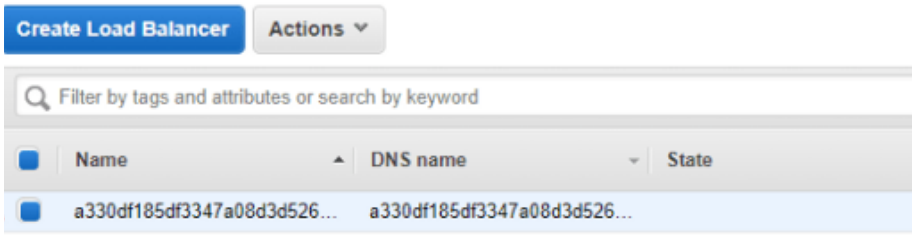


The deployment to EKS is handled by YAML files, which are the configuration files for K8s.

I have applied those files to the cluster created in the previous step.

We can see the information about deployments and other resources by simply running the command 'kubectl get all'.

As we are leveraging the AWS services, we can see the load balancer being created based on our service manifest file.



There are other AWS resources created, such as VPC, Security Groups, etc., to make all this happen.

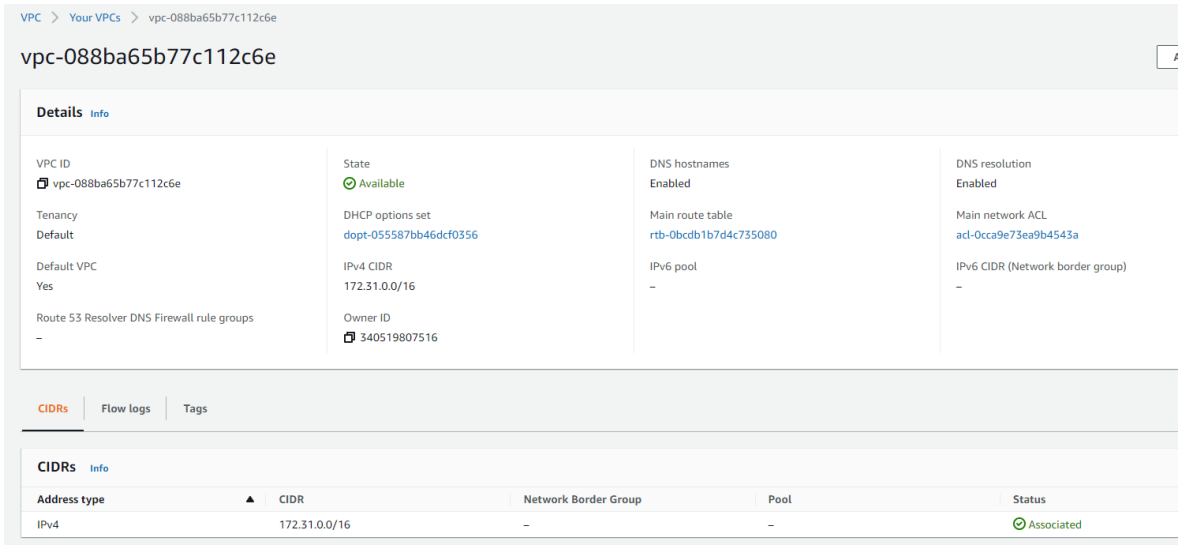
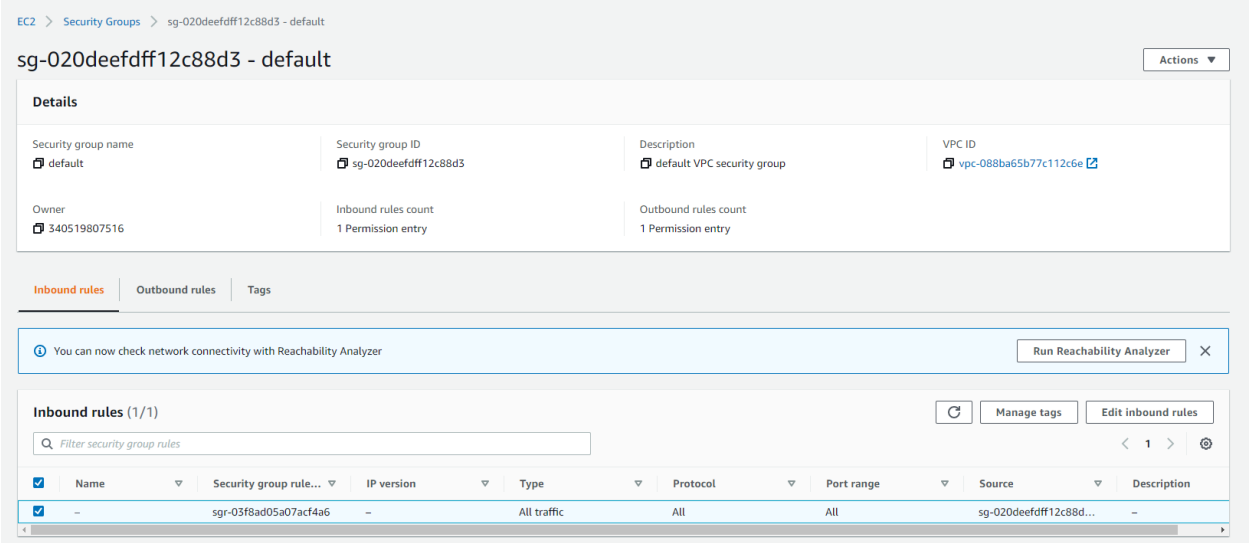


Image of VPC configuration



Security group configuration

Let's go to the public address of the load balancer, and our web application dashboard will show up.