

oooo

PRES

KnowQuest Team A

oooo



What's the project

The KnowQuest Platform and AWS Improvement project

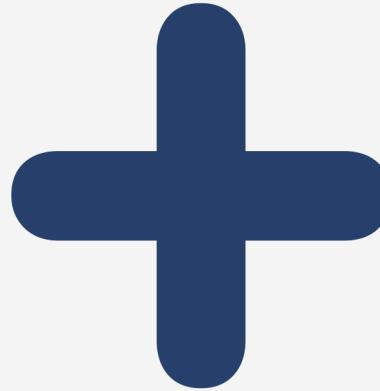
- Underwent a comprehensive transformation to enhance the efficiency and reliability of the KnowQuest operations.
- The design and implementation of a new cloud architecture, migrating the applications onto a new cloud PaaS model, and AWS infrastructure to the new setup.
- Deploying a demo application and initiating CloudWatch Data Visualization for KnowQuest.
- Performance testing was integral to the project's success.

Why?

- In addressing the multifaceted challenge at hand, aiming to provide innovative solutions that align with the stakeholders' evolving requirements.
- The project will improve the user experience of the KnowQuest website and make it more reliable and efficient.
- The dynamic interplay of stakeholder requirements, technical solutions, and experimental methodologies encapsulates the essence of our project, making it both interesting and relevant for professionals seeking to navigate the evolving landscape of Informatics.



AWS Migration for KnowQuest



EC2

To host the node clusters
of eks applications

CloudFormation

To manage the
provisioning and
deployment of AWS
resources automatically

OpenSearch

To provide a search and
analytics engine for the
website and etc.

S3

To store yaml configuration
and deployment files and
automatic backups

RDS

(Relational Database Service)
To manages the database with full
administration support, simplifying
setup, patching, and backups for
enhanced scalability and reliability

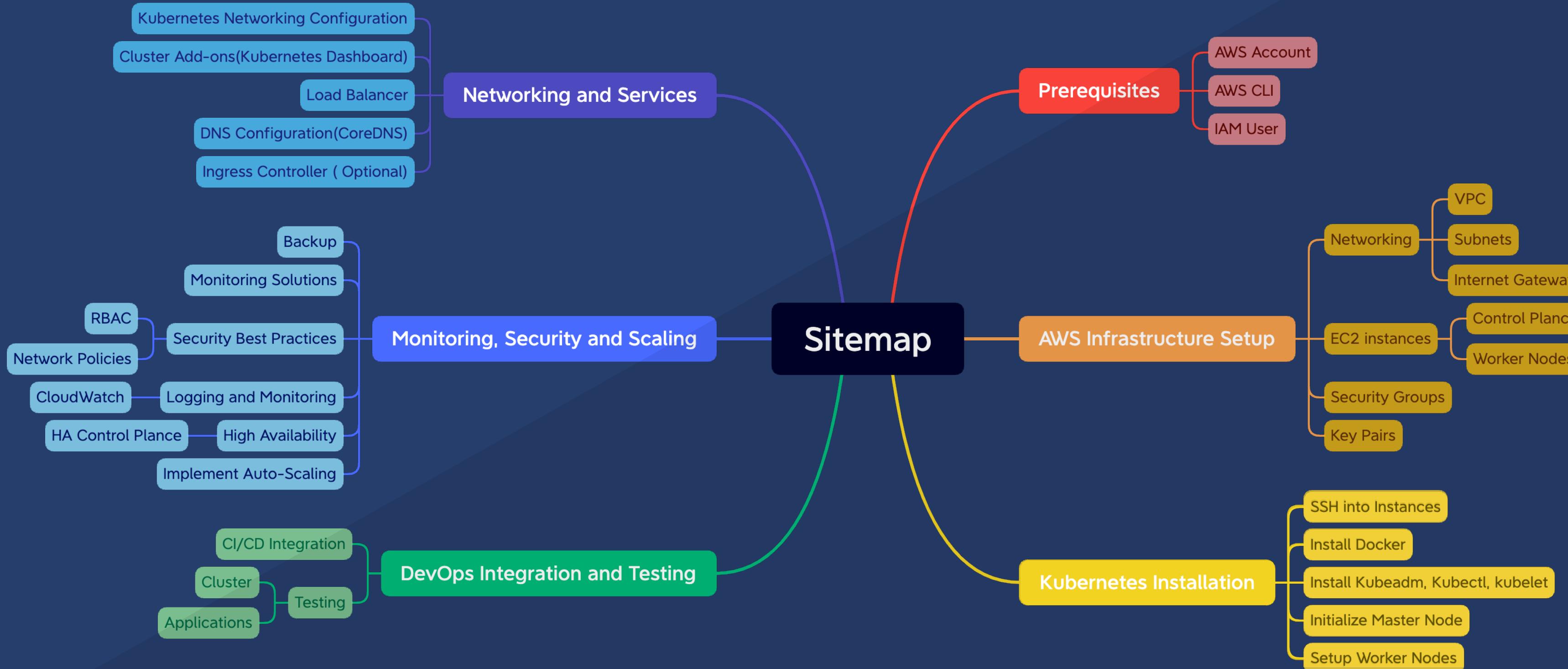
Lambda

Executes event-driven serverless code,
providing flexibility and efficiency
without the need for managing server
infrastructure, enabling dynamic and
cost-effective scaling based on demand

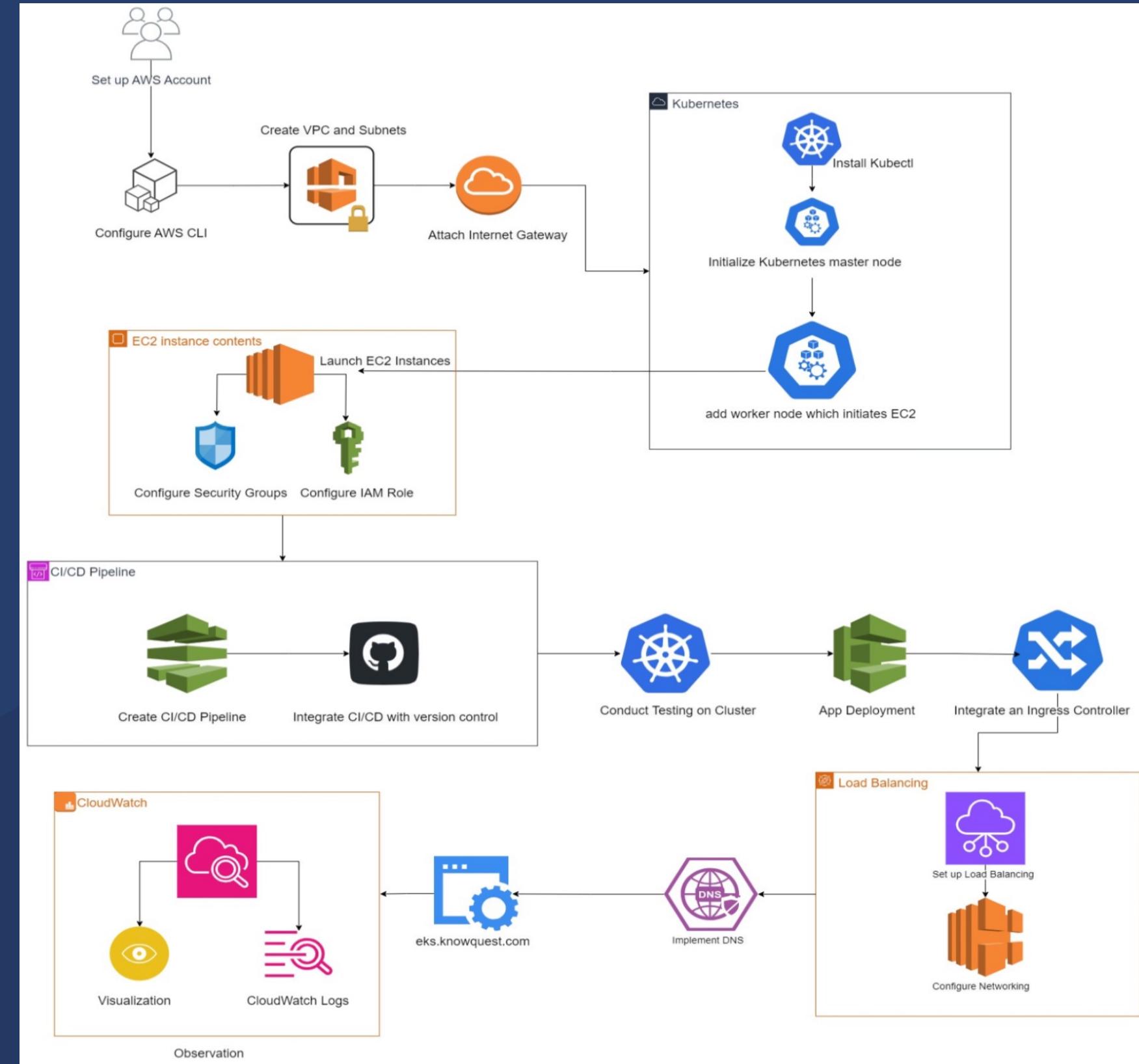


KnowQuest Sitemap

• based on AWS cloud service



Kubernetes Architecture



IAM Role for EKS Cluster

The sponsors have given an IAM role as a prerequisite for maintaining the permissions on EKS service. They have given us an IAM role which includes EKS cluster policies.

Create Worker Nodes

Create IAM Role for EKS Worker Nodes

Create Dedicated VPC for the EKS Cluster

Create EKS Cluster

Install & Setup IAM Authenticator and Kubectl Utility

Kubernetes Installation

VPC for the EKS Cluster

As master node should be in a network VPC. So, created a VPC and its subnets using cloud formation.

The screenshot shows the AWS VPC and Security Groups interfaces. The top part displays the 'Your VPCs' list with two entries: 'vpc-076ebd5cafd200059' and 'eks-vpc-stack-VPC'. The bottom part shows the 'Details' page for the 'eks-vpc-stack-VPC', which includes sections for Subnets, Route tables, Internet gateways, Egress-only internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, Endpoints, Endpoint services, NAT gateways, and Peering connections. The right side of the interface shows the 'sg-0de77392d6a290f26 - eks-cluster-sg-eks-cluster-365256518' security group details, including its description, owner, inbound rules count (4), and outbound rules count (1). The 'Inbound rules' table lists four entries, each allowing all traffic from specific source IP ranges.

The screenshot shows the EKS console under the 'Clusters' section. It displays the 'eks-cluster' cluster details. Key information includes:

- Status: Active
- Kubernetes version: 1.28
- Support type: Standard support until November 2024
- Provider: EKS

The 'Details' section provides API server endpoint (<https://BCDBA227F5EFFCEF268FF1C141D72692.gr7.us-east-1.eks.amazonaws.com>), OpenID Connect provider URL (<https://oidc.eks.us-east-1.amazonaws.com/id/BCDBA227F5EFFCEF268FF1C141D72692>), Cluster IAM role ARN (<arn:aws:iam::736679853012:role/eksClusterRole>), and Platform version (eks.4).

Create EKS Cluster

In this step created an EKS cluster which is a master node and connected it with the VPC created in the above step.

IAM Au. & Kubectl Utility

In this step, we installed and set up the IAM Authenticator and the Kubectl Utility.

```
(base) alina@AlinadeMacBook-Pro ~ % aws configure
AWS Access Key ID [*****]: AKIA2XBMKPKABT6KXNS
AWS Secret Access Key [*****]: Qi40SqBmAtqJS8GAZFUaCrkst5QqDxvzaUM5f/x
Default region name [us-east-1]:
Default output format [None]:
(base) alina@AlinadeMacBook-Pro ~ % aws eks update-kubeconfig --name eks-cluster --region us-east-1
● Added new context arn:aws:eks:us-east-1:736679853012:cluster/eks-cluster to /Users/alina/.kube/config
(base) alina@AlinadeMacBook-Pro ~ % aws s3 ls
2022-12-08 15:33:11 amplify-cryptocalcapp-dev-153302-deployment
2022-12-08 15:45:51 amplify-cryptocalcapp-master-204544-deployment
2022-11-19 12:57:53 cdk-hnb659fds-assets-736679853012-us-east-1
2023-11-10 15:47:53 cf-templates-wmlu3xbmqss-us-east-1
2023-11-10 15:32:48 cf-templates-wmlu3xbmqss-us-east-2
2023-11-02 14:18:33 elasticbeanstalk-us-east-1-736679853012
(base) alina@AlinadeMacBook-Pro ~ % more /Users/alina/.kube/config
apiVersion: v1
clusters:
- cluster:
  certificate-authority-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURCVENDQWUyZ0F3SUJBZ01JRHZxdTF0wdxaxd3RFFZSktvWklodmNOQVFTEJRQXdGVEVUTUFR0ExVUUKQXHS2EzVm1awEp1WlhSbGNQWVGdzB5TxFeE1UQX1NRFuYTkRsYUZ3MHPnEkV4TURje1UQXhORGxhTUJVeApFekFSQmdOVkJBTVRDxDQxW1WeWjVjBaWE13Z2dFa1BMEdDU3FHU0lím0RRRUJBUVBQTRJQKR3QXdnZ0VLckFvSUJBUnhYTJnOuhZNerVaZoLzAvTXBFUS9XaFZKZdhBWx1UZG02t2REHVpd1o30Cthb0JmcH1PTGRMNMGKQnJxay9ncTRPcerfVTJUU2NDQytjTmZCR1wTcFx1ThAZnZf1z8Yzh2pkC0txVHtK2adXVKn3JwNmRCMwpvVRRYUdreUVkyWdcC2YvRHacT1zXdcvXZsLzIakU0AhDaEPb0IxId1rJU62VkmVeE2MDBoZRIl3hXCh1ae00c1k5akRMVjFobWFMcWx5aDarZ0Rwy2V0mVGSE5GGQU01p4WhdQa2tLzJhBULSS255cThpaZMrnQKdEdvYktatUc5U2xScXBGc1R3eQ4Weh1L0oyZnBWK3JkU2Jgk1Nxr3BStkZwYVV6elZUcHv2REs0dVR3MkFGaQpYKzN0cWo4djAvVzdNSUd1bHRoMu5tVdk2QnExQwdNQkFBR2pXVEJYtUE0R0ExVWREd0VCL3dRRUF3SUnwREFQCKJnT1ZIUk1CQWY4RUJUQURBUUgvTUwR0ExVWRZ1FX0kJRTGQR0dT1U2YWRZRVphQmd4Nv3UURsMjVqQVYKQmd0VkhSRUVEakFNZ2dwcmRXSmxjbTVsZEdwek1BMedDU3FHu0liM0RRRUJDd1VBQTRQkFRQm9E0E1cjBWSgpDVGxpVpxeW1XUmROZFM2dmUvYkhUNjNwWEvtUmxxraytCMzJqWXZsUC8zdkx5SFdr0E9TdXlyR1VrNJaUmlVCnFmS2FKMFdZc01vRj10djBORVFeElls3pNUnkx0WgrL0ZDzhuTkJGSjdvUWFBCzJHUxPUsk1JeUxu1RDXV1KaVNyTrhEnJVUnswSTRKMEtGSDZMWFm2b3FCRG1LvdhISX14dwtyt1hGyjgxNuNiRwhIuZhxU0p0U2oxaj1fSgpQQnY0QWNaRGJDE91c2t0TyZZ2PM0tuRH0dVJ1V1PQkx0YKrVmwvVOGRRMUJrdTqrVHJEN2h2WUN2R2V6CjJuODNzLzFDVDJ6Q1VBagnxVRjQ0M5UEtGwmN2cWRWFJpk1ZMajRBV3V1W1B1cUU5WTJKV0xLK29YN1FZwjYKMFRZTVg0bjNvUghaCi0tLS0tRU5Ef1NEU1RjRk1D0VRF1S0tLS0K
server: https://BCDBA227F5EFFCEF268FF1C141D72692.gr7.us-east-1.eks.amazonaws.com
name: arn:aws:eks:us-east-1:736679853012:cluster/eks-cluster
contexts:
- context:
  cluster: arn:aws:eks:us-east-1:736679853012:cluster/eks-cluster
  user: arn:aws:eks:us-east-1:736679853012:cluster/eks-cluster
  name: arn:aws:eks:us-east-1:736679853012:cluster/eks-cluster
current-context: arn:aws:eks:us-east-1:736679853012:cluster/eks-cluster
kind: Config
```

The screenshot shows the AWS IAM Roles page for the 'eksworkernoderole'. The role is an Amazon EKS Role. The 'Summary' section displays basic information: Creation date (November 10, 2023, 16:57 UTC-05:00), ARN (arn:aws:iam:736679853012:role/eksworkernoderole), Instance profile ARN (arn:aws:iam:736679853012:instance-profile/eks-6ec5e581-dbc9-a597-19f7-efa24abd0ec4), Last activity (40 minutes ago), and Maximum session duration (1 hour). The 'Permissions' tab is selected, showing three managed policies attached: 'AmazonEC2ContainerRegistryReadOnlyPolicy', 'AmazonEKS_CNI_Policy', and 'AmazonEKSWorkerNodePolicy'. Each policy is listed with its type (AWS managed) and the number of attached entities (1).

IAM Role for EKS Nodes

In this step, we created an IAM role for worker nodes which gives permissions and EC2 container policy to access permissions to each instance in it.

Create Worker Nodes

In this step, we created a worker node by creating a node group.

Nodes (4) Info

Node name	Instance type	Node group	Created	Status
ip-192-168-104-143.ec2.internal	t2.small	eks-worker-node-group	Created November 13, 2023, 18:32 (UTC-05:00)	Ready
ip-192-168-2-108.ec2.internal	t2.small	eks-worker-node-group	Created November 13, 2023, 18:29 (UTC-05:00)	Ready
ip-192-168-53-32.ec2.internal	t2.small	eks-worker-node-group	Created November 13, 2023, 18:32 (UTC-05:00)	Ready
ip-192-168-97-42.ec2.internal	t2.small	eks-worker-node-group	Created November 13, 2023, 18:29 (UTC-05:00)	Ready

Node groups (1) Info

Group name	Desired size	AMI release version	Launch template	Status
eks-worker-node-group	4	1.28.3-20231106	-	Active

Deploying The Demo Application

•
•
•
•
•
•
•
•

knote

A simple note-taking app.

Upload an image

選擇檔案 未選擇任何檔案

Write your content here

Publish

Notes

test

Networking & Testing

Implement Code by Creating Docker Image

Create yaml files and make sure the application is in the pods and also running. Here we uploaded the sample files, the first one is “deployment.yaml” and the other one is “service.yaml”.

Then, we can use “kubectl get pods” command to confirm the application is working.

```
(base) alina@AlinadeMacBook-Pro ~ % kubectl apply -f /Users/alina/Desktop/Me/USA/INT6940\ KnowQuest\ Project/deployment.yaml
deployment.apps/test-app created
%
(base) alina@AlinadeMacBook-Pro ~ % kubectl apply -f /Users/alina/Desktop/Me/USA/INT6940\ KnowQuest\ Project/service.yaml
service/test-app-service created
.
.
.
(base) alina@AlinadeMacBook-Pro ~ % kubectl get pods
NAME                                     READY   STATUS    RESTARTS   AGE
knote-7c4c557759-2snd5                   1/1     Running   0          6h1m
knote-7c4c557759-n9qdd                   1/1     Running   0          6h1m
mongo-f4568867c-4b594                    1/1     Running   0          6h1m
mongo-f4568867c-b2fq2                   1/1     Running   0          6h1m
test-app-77b66f4c9f-62q5x                1/1     Running   0          29s
backend-deployment-79ff848469-6krv9      0/1     CrashLoopBackOff
                                         webdevproject-
                                         CrashLoopBackOff
75 (3m45s ago) 6h3m
webdevproject-backend-deployment-79ff848469-v4rcb 0/1
                                         webdevproject-
                                         CrashLoopBackOff
CrashLoopBackOff 75 (79s ago) 6h
```

```
(base) alina@AlinadeMacBook-Pro ~ % kubectl get svc
NAME           CLUSTER-IP      EXTERNAL-IP      PORT(S)        AGE   TYPE
LoadBalancer   10.100.148.206  ae02e45dbe6ca430cb1198be49608712-1191761865.us-east-1.elb.amazonaws.com  80:30000/TCP      22d  knote
kubernetes     10.100.0.1      <none>          443/TCP       26d   ClusterIP
mongo          10.100.102.224  <none>          27017/TCP     26d   ClusterIP
nginx-ingress-ingress-nginx-controller  10.100.223.65  a62fda5640b4648669db885c836ac0ef-1082317029.us-east-1.elb.amazonaws.com  80:31909/TCP,443:31768/TCP  2d23h  LoadBalancer
nginx-ingress-ingress-nginx-controller-admission  10.100.77.199  <none>          443/TCP       2d23h ClusterIP
sample-app-service  10.100.3.3      <none>          8080/TCP     3d3h   ClusterIP
test-app-service  10.100.62.12    a68e2d5b4684e4f948b9bf1842e8c569-2123278139.us-east-1.elb.amazonaws.com  80:32603/TCP      4m7s  LoadBalancer
```

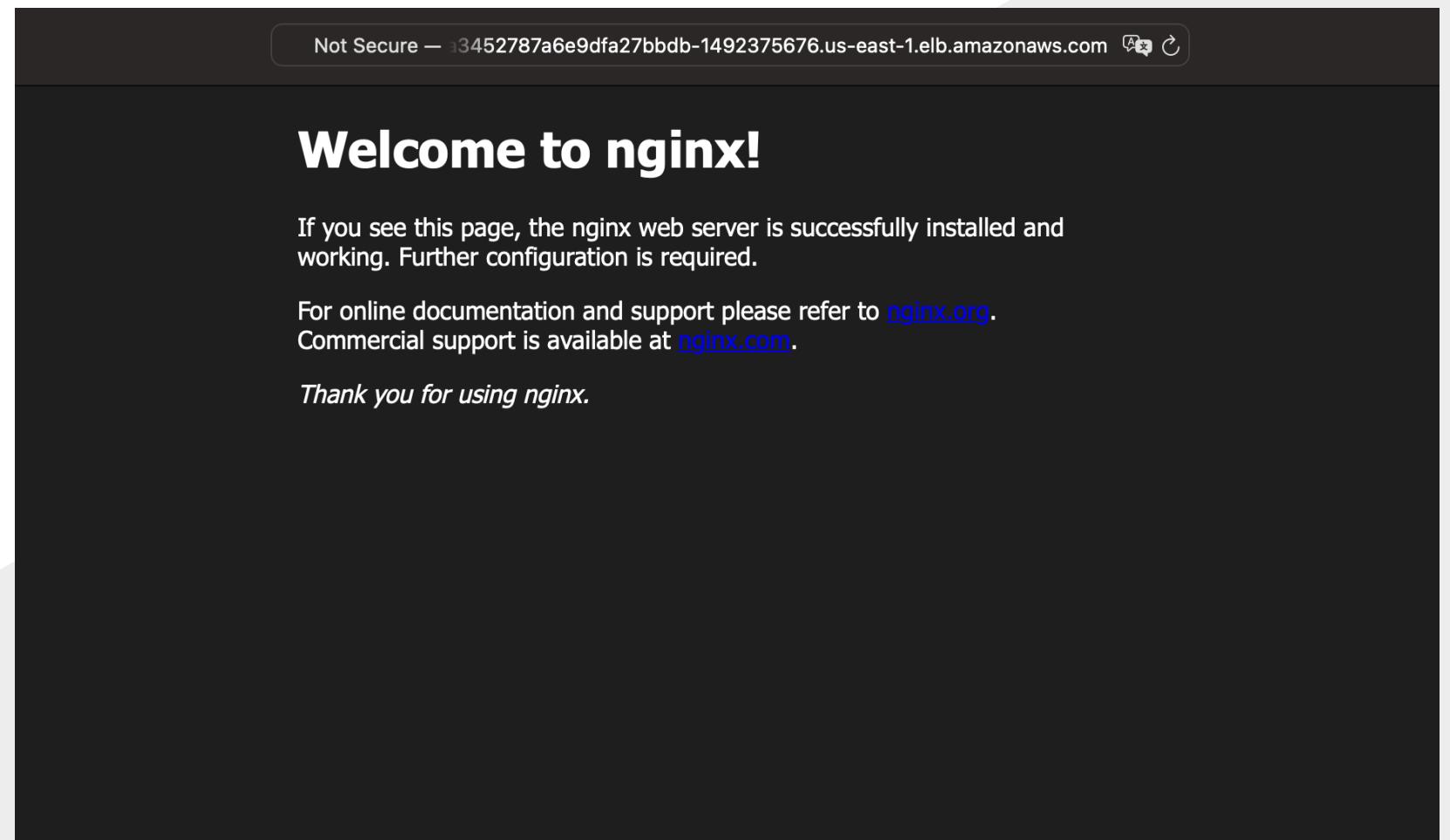
After deploying the sample service application, we can use the “kubectl get svc” command to check the external link that been generated by Cluster. As the application shows, the external link will be “ a3aef796bd8a3452787a6e9dfa27bbdb-1492375676.us-east-1.elb.amazonaws.com”

Confirm the External Link

Query DNS

This step is to obtain domain name mapping or other DNS records

```
(base) alina@AlinadeMacBook-Pro ~ % nslookup  
a68e2d5b4684e4f948b9bf1842e8c569-2123278139.us-  
east-1.elb.amazonaws.com  
Server: 2001:558:feed::1  
Address: 2001:558:feed::1#53  
  
Non-authoritative answer:  
  
Name: a68e2d5b4684e4f948b9bf1842e8c569-2123278139.us-  
east-1.elb.amazonaws.com Address: 54.221.138.35  
Name: a68e2d5b4684e4f948b9bf1842e8c569-2123278139.us-  
east-1.elb.amazonaws.com Address: 54.144.186.68  
  
  
(base) alina@AlinadeMacBook-Pro ~ % curl  
a68e2d5b4684e4f948b9bf1842e8c569-2123278139.us-  
east-1.elb.amazonaws.com  
<!DOCTYPE html>  
 <html>  
   <head>  
     <title>Welcome to nginx!</title>  
   <style>  
     html { color-scheme: light dark; }  
     body { width: 35em; margin: 0 auto;  
            font-family: Tahoma, Verdana, Arial, sans-serif; }  
   </style>  
 </head>  
  
<body>  
  <h1>Welcome to nginx!</h1>  
  <p>If you see this page, the nginx web server is successfully  
  installed and  
          working. Further configuration is  
  required.</p>  
  <p>For online documentation and support please refer to  
  <a href="http://nginx.org/">nginx.org</a>,<br/>  
          Commercial support is available  
  at  
  <a href="http://nginx.com/">nginx.com</a>.</p>  
  <p><em>Thank you for using nginx.</em></p>  
</body>  
</html>
```



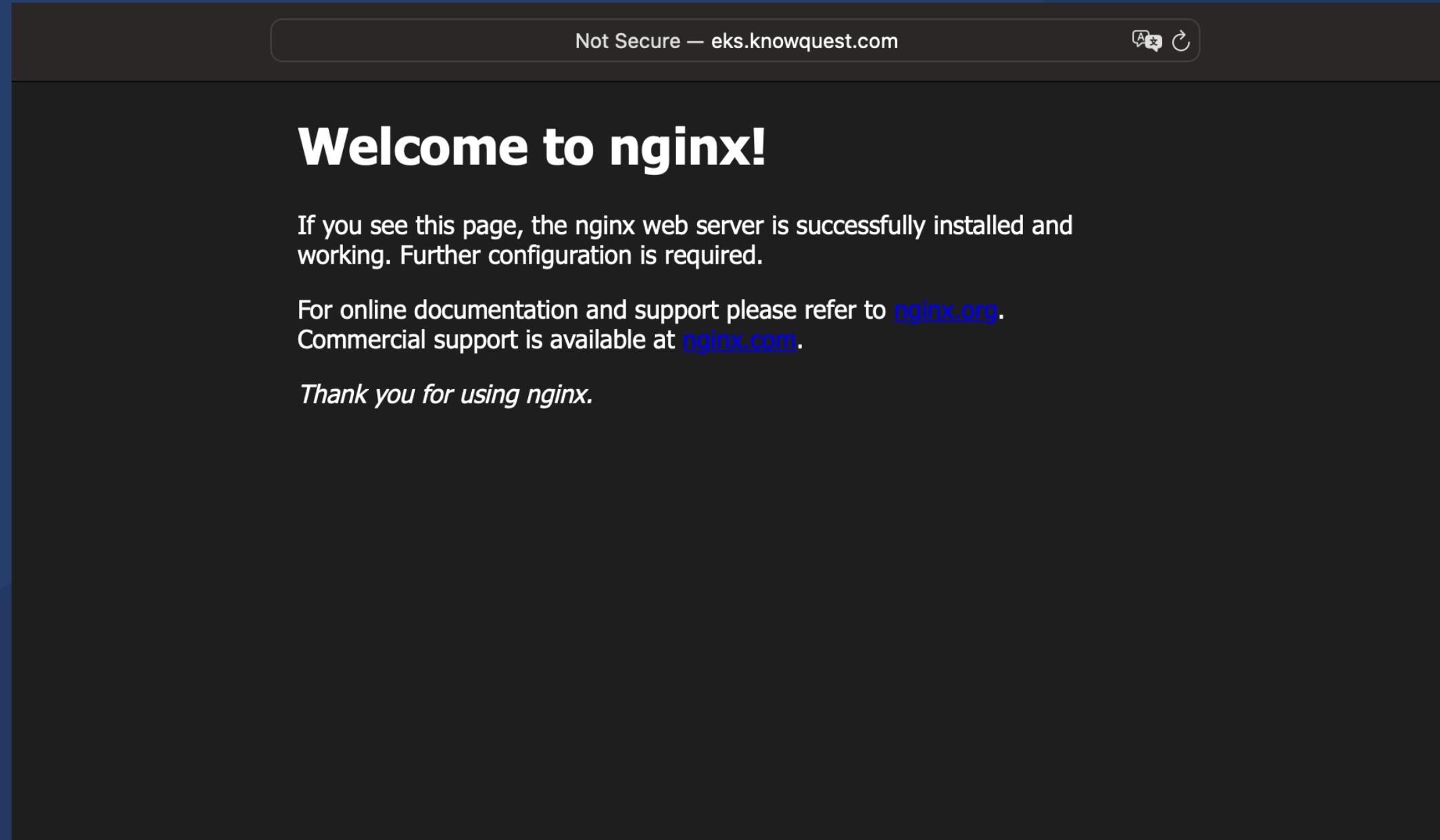
After we successfully configure the application, we can directly test the domain URL by open a new browser..

Test the Domain

Update DNS Route

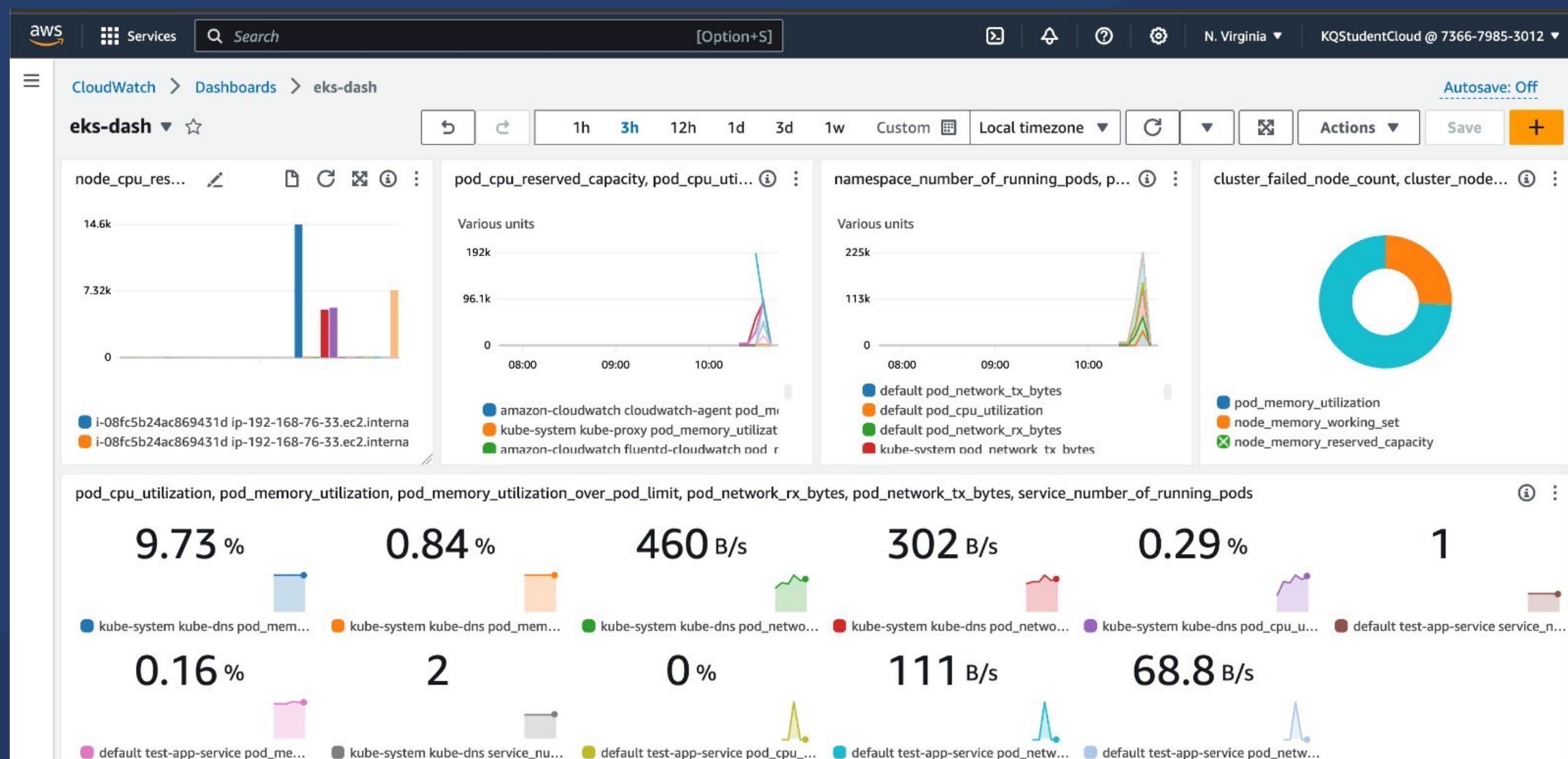
As our sample application is running and able to access by the external URL, we would like to change the URL into KnowQuest's domain by update in DNS.

http://eks.knowquest.com



Monitoring – AWS CloudWatch

Monitoring AWS workloads and resources is made possible by Amazon CloudWatch, which offers real-time insights via metrics and logs. It's very helpful for monitoring how resources are used, how well applications perform, and how well operations are running. It also enables automated reactions to preset scenarios.



What worked with the sponsor

- We commonly communicated with our sponsor using Discord App for project update and updated the status of report by email.
- We have updated the CEO on the project's progress, including any potential risks or challenges.
- We asked for any resources or URL from sponsor regularly to meet the needs of our project, including the AWS account, permission to access AWS services, and other sponsor's documents.



What are areas for improvement on engagement with sponsor

- During collaboration with the sponsor, it became evident that KnowQuest had a single individual responsible for both backend and cloud aspects without specific cloud expertise. This constraint limited our access to cloud resources and expert guidance.
- Following the completion of the deployment process, we recognized the importance of scheduling additional meetings with the sponsor to facilitate open communication, share insights, and validate expectations for ongoing alignment.



further improvement | DATA VISUALIZATION SUGGESTION

Create a Dashboard

- Go to the “Dashboard” tab in Kibana
- Click on the “+”. Button to add a new panel to the dashboard.
- Choose the type of visualization you want to add to the dashboard and select the saved visualization.

The Kibana logo is displayed diagonally across the slide. It consists of the word "kibana" in a lowercase, sans-serif font. The letters are white with a slight shadow, set against a dark blue background that has diagonal stripes.

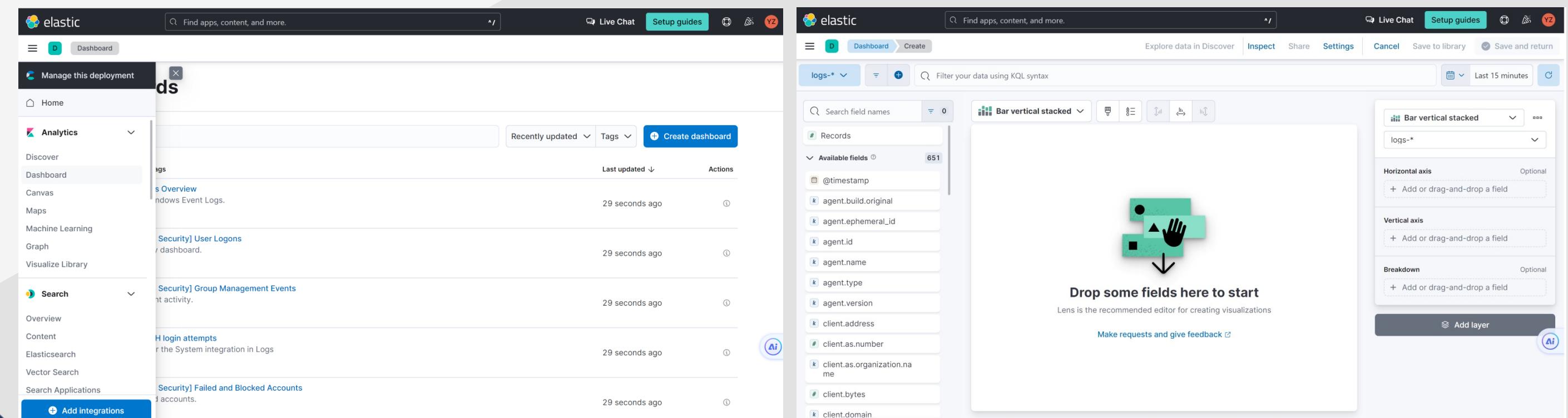
Create Visualization

- In the “Visualize” tab, create visualization for the data. Examples include bar charts, line charts, pie charts, etc.
- Choose the appropriate fields and aggregations for your visualizations.

Log-File Processing

- Kibana relies on Elasticsearch as the backend data store. So, first, you need to index your log data into Elasticsearch. Ensure that Elasticsearch is set up and running.
- Configure Logstash or File beat to point to your log files or data sources. Update the configuration files to specify the location of your logs and the Elasticsearch endpoint.
- After indexing, check that your log data is present in the Elasticsearch indices. You can use the Elasticsearch REST API or Kibana's Dev Tools to query the indices.
- In Kibana, go to the "Management" tab and then select "Index Patterns."
- Create an index pattern that matches your log data indices. This step helps Kibana understand how to explore and visualize your data.

Kibana



THANK YOU



Sandeep Samsani

AWS Principal
Editor



Jui Ying Lee

Lead
AWS & Editor



Jie Zhu

Data Support
Editor & Slides



Vivek Vaghasiya

AWS Assistant
Editor



Ananya Kakani

AWS Assistant
Editor



Yi Zou

Data Principal
Kibana Intro