Dang Quoc Khoa

SD4591

PI-SHAPED

Just inform that in this section, we have some lines of codes that this document does not provide in detail because we have done it on the first program called "*Practical DevOps for Devs*". This document will focus on the latest changes.

1. Manage K8s Application

DevOps for Devs:

You would need to have a repository for the MSA application. You can clone the source code from https://github.com/nashtech-garage/kubernetes/tree/master/src

App repository pattern: https://github.com/<your-github-name>/<staffcode_msa>

Example: https://github.com/hoanglecao/sd0660_msa

You also need another repository for infrastructure. Provision codes and manifest files will be stored in this repository.

App repository pattern: https://github.com/<your-github-name>/<staffcode_aws_infrastructure>

Example:
https://github.com/hoanglecao/sd0660_aws_infrastructure

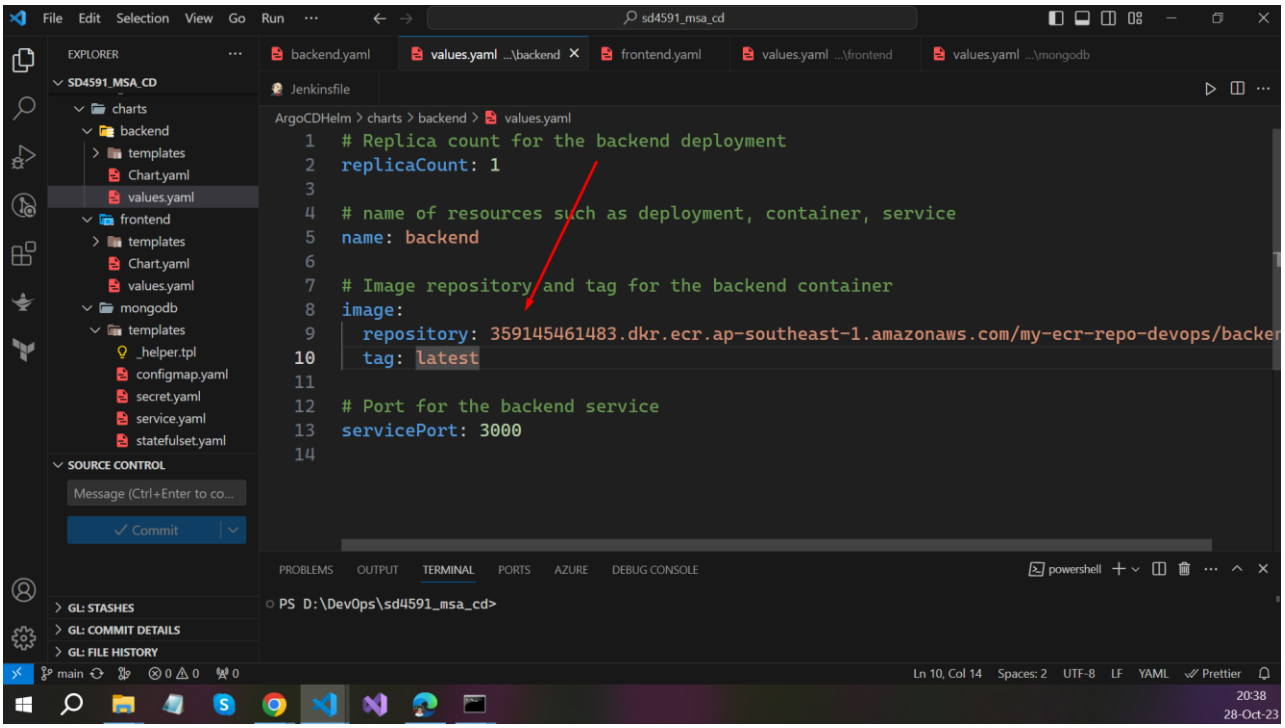
DevOps for Devs:

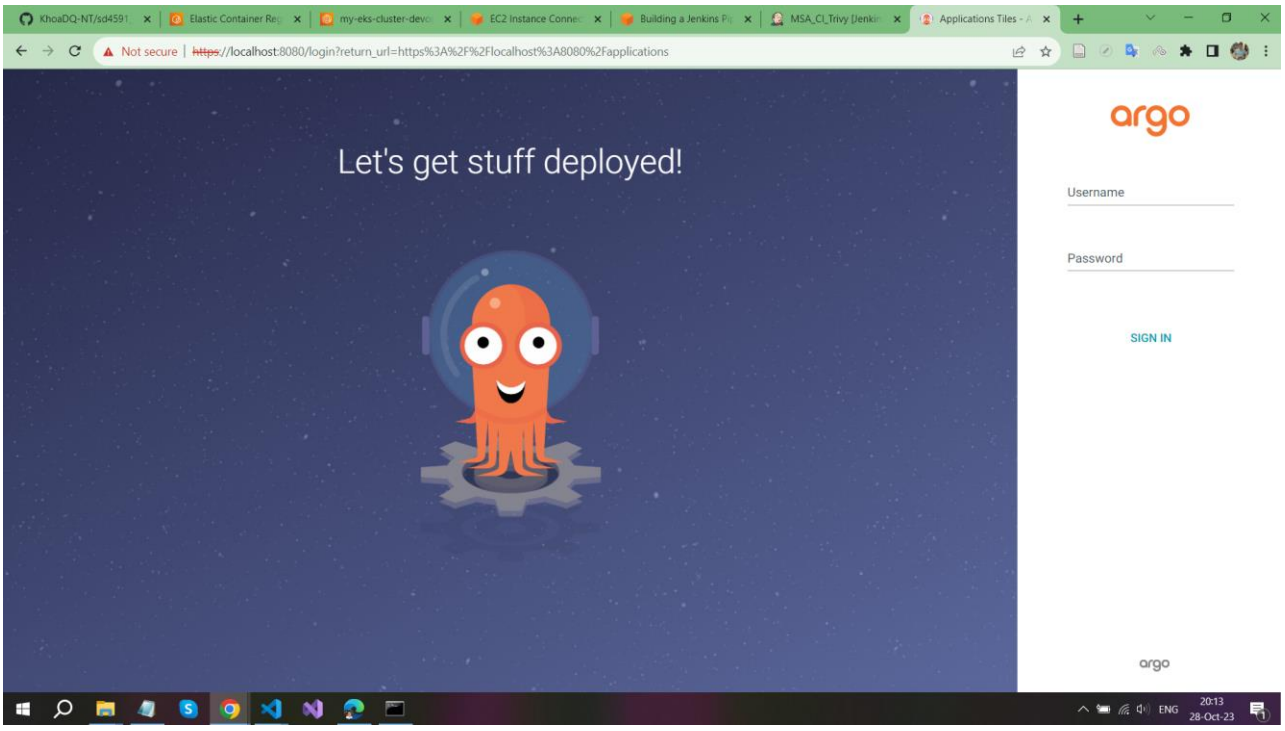
Create K8s manifest.

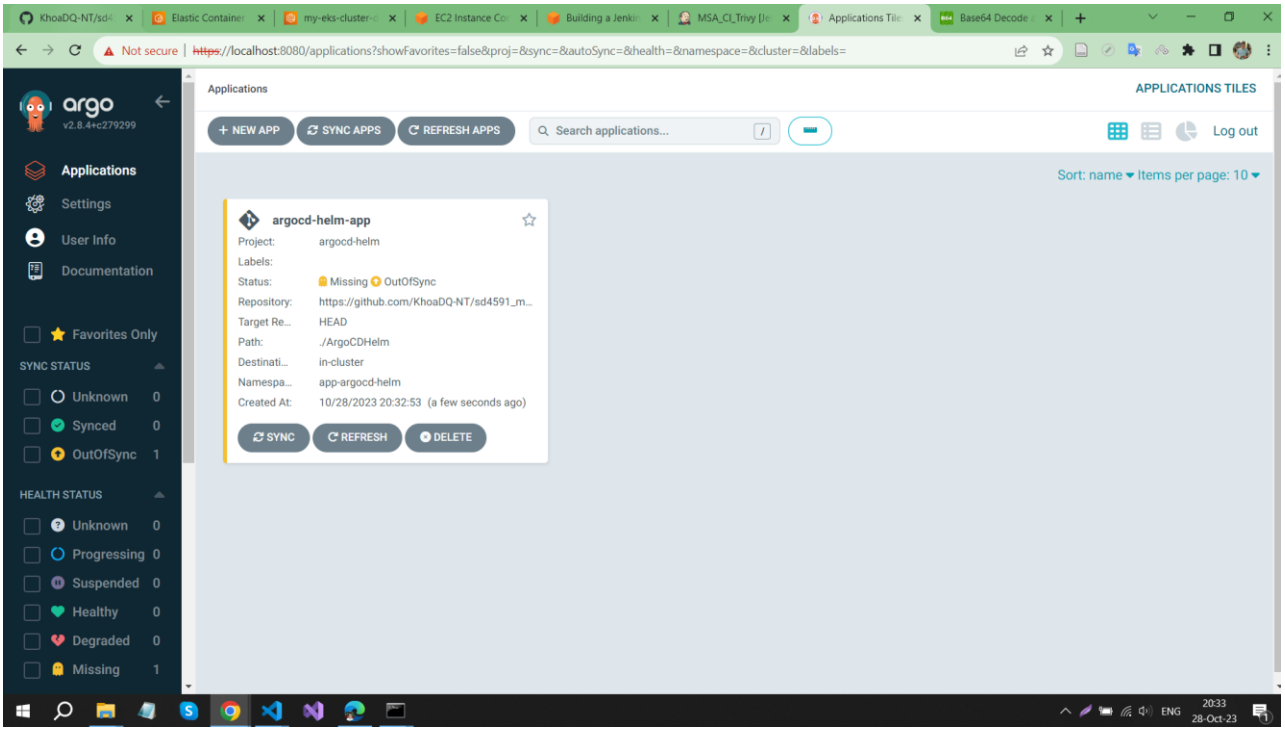
Pi-sharp:

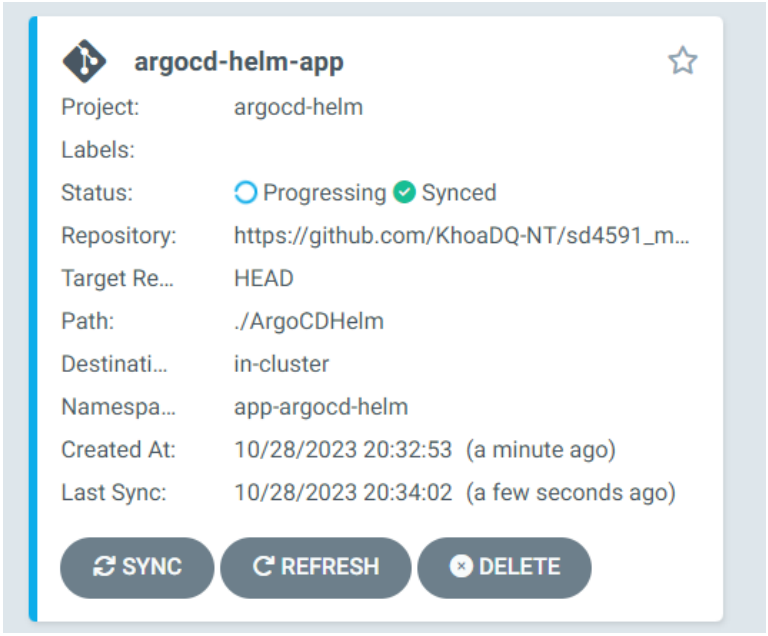
Use helm-charts to manage K8s manifest.

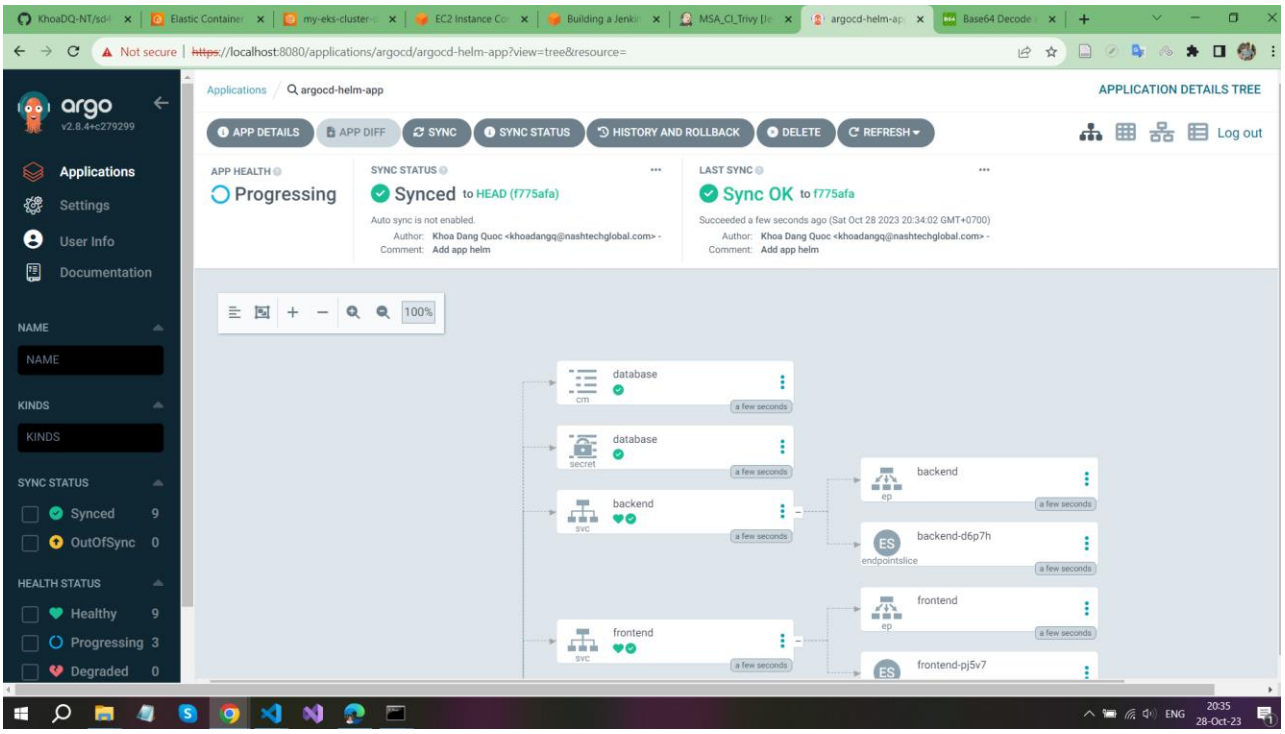
After building and pushing the image to ECR (I have done requirements of 2-3-4-5 and then back to this requirement), we apply helm to project and update accordingly image information for each service.



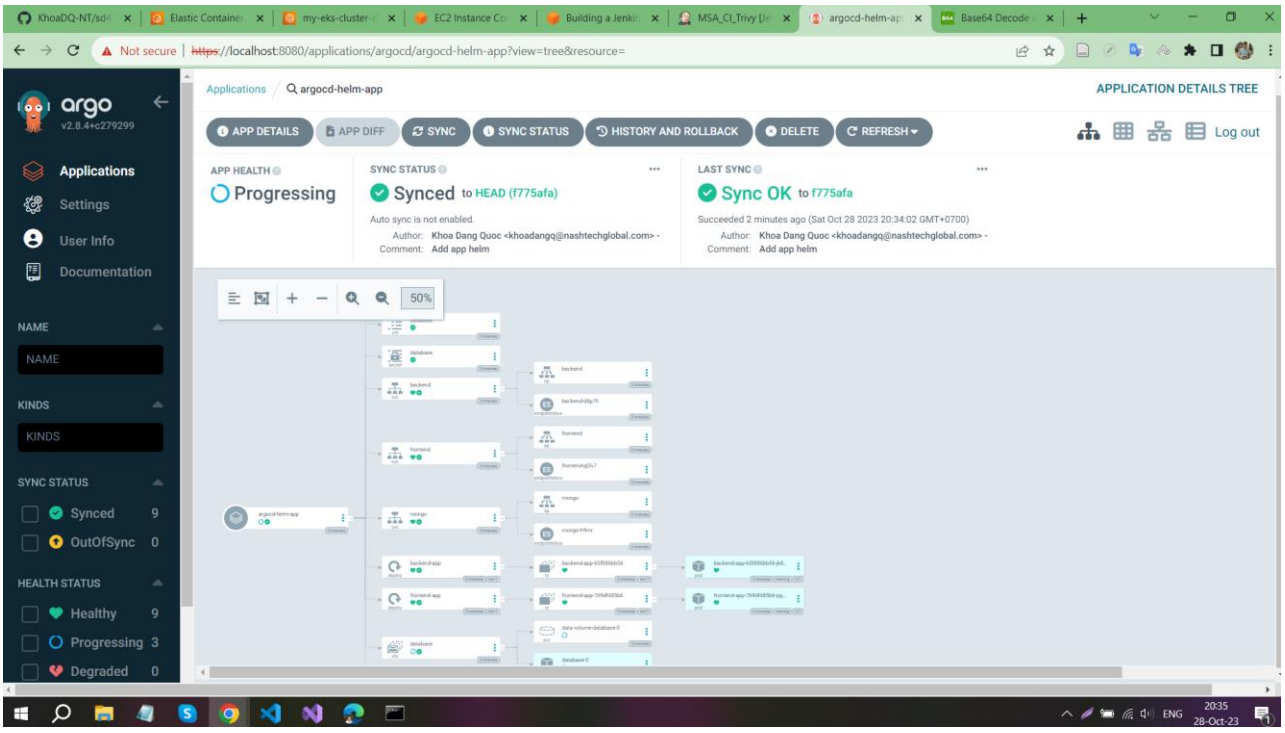


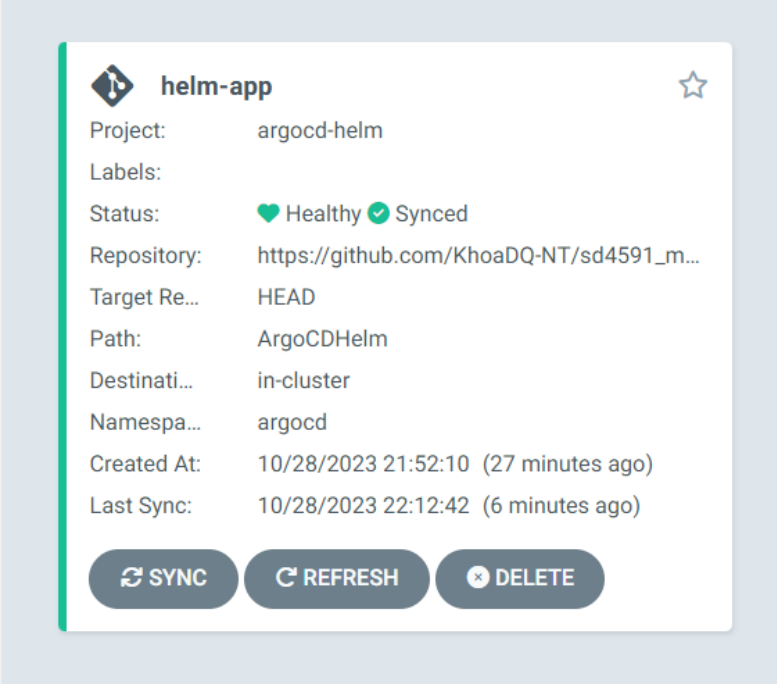

Not secure | https://localhost:8080/applications?showFavorites=false&proj=&sync=&autoSync=&health=&namespace=&cluster=&labels=
argo
v2.8.4+c279299
Applications
Settings
User Info
Documentation
Favorites Only
SYNC STATUS
Unknown 0
Synced 0
OutOfSync 1
HEALTH STATUS
Unknown 0
Progressing 0
Suspended 0
Healthy 0
Degraded 0
Missing 1
Applications
APPLICATIONS TILES
+ NEW APP
SYNC APPS
REFRESH APPS
Search applications...
Log out
Sort: name
Items per page: 10
argocd-helm-app
Project: argocd-helm
Labels:
Status: Missing OutOfSync
Repository: https://github.com/KhoaDQ-NT/sd4591_m...
Target Re... HEAD
Path: ./ArgoCDHelm
Destinati... in-cluster
Namespa... app-argocd-helm
Created At: 10/28/2023 20:32:53 (a few seconds ago)
SYNC
REFRESH
DELETE
ENG
20:33
28-Oct-23

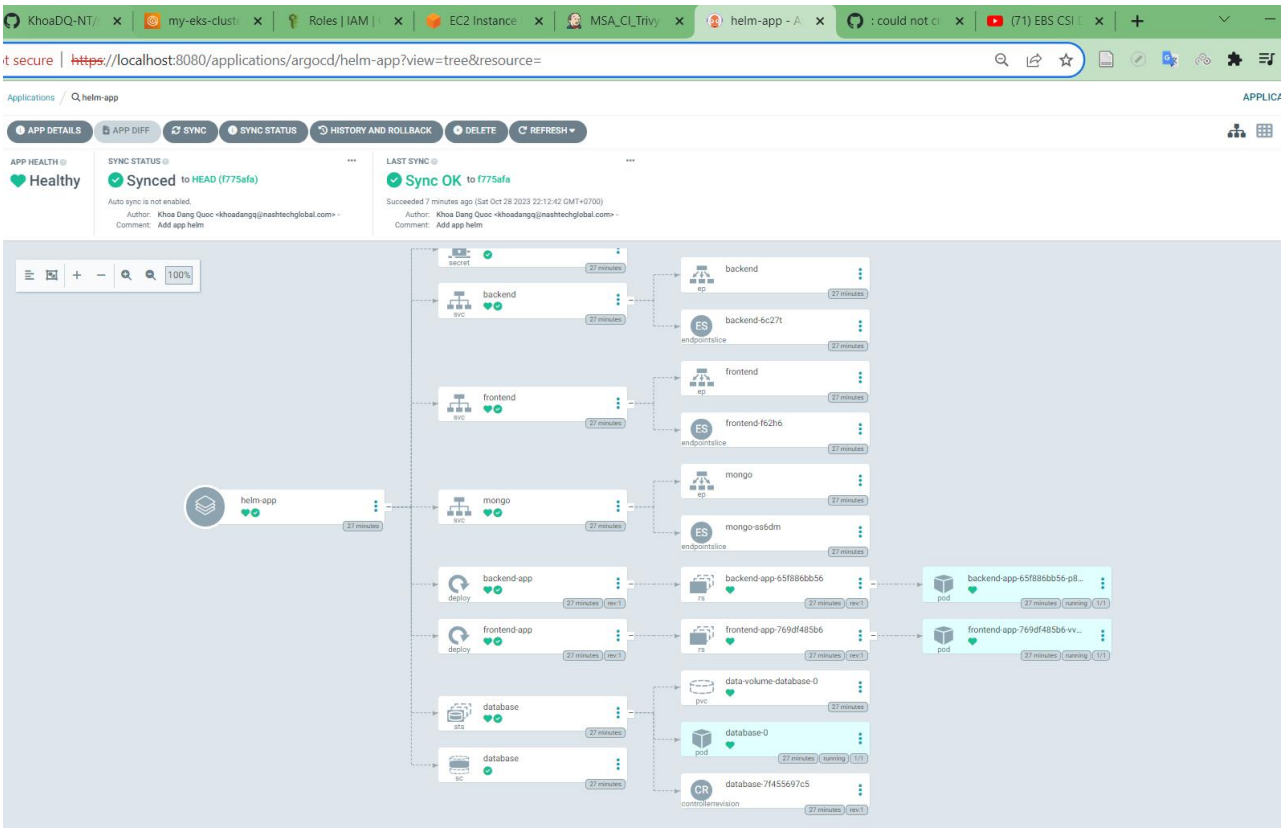

argocd-helm-app
Project: argocd-helm
Labels:
Status: Progressing Synced
Repository: https://github.com/KhoaDQ-NT/sd4591_m...
Target Re... HEAD
Path: ./ArgoCDHelm
Destinati... in-cluster
Namespa... app-argocd-helm
Created At: 10/28/2023 20:32:53 (a minute ago)
Last Sync: 10/28/2023 20:34:02 (a few seconds ago)
SYNC
REFRESH
DELETE



My application is pending for a long time because the EBS does not have permission on volume, after investigating more on EBS driver and configuring exactly policy, it works OK



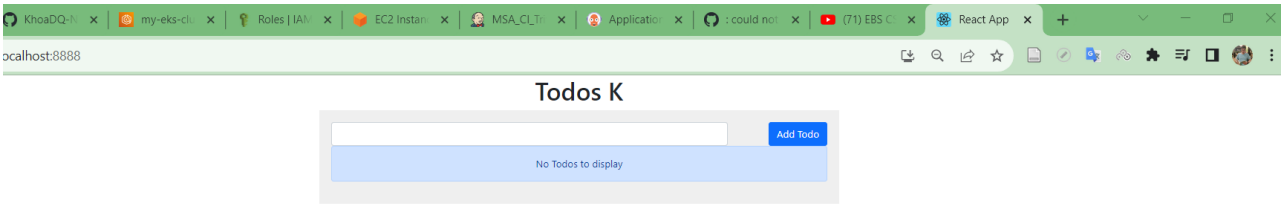

helm-app
Project: argocd-helm
Labels:
Status: Healthy Synced
Repository: https://github.com/KhoaDQ-NT/sd4591_m...
Target Re... HEAD
Path: ArgoCDHelm
Destinati... in-cluster
Namespa... argocd
Created At: 10/28/2023 21:52:10 (27 minutes ago)
Last Sync: 10/28/2023 22:12:42 (6 minutes ago)
SYNC
REFRESH
DELETE


https://localhost:8080/applications/argocd/helm-app?view=tree&resource=
Applications
helm-app
APP DETAILS
APP DIFF
SYNC
SYNC STATUS
HISTORY AND ROLLBACK
DELETE
REFRESH
APP HEALTH
Healthy
SYNC STATUS
Synced to HEAD (f775afa)
LAST SYNC
Sync OK to f775afa
helm-app
backend
backend-6c27t
frontend
frontend-f62h6
mongo
mongo-ss6dm
backend-app
backend-app-65f886bb56
frontend-app
frontend-769df485b6
database
data-volume-database-0
database-0
database-7f455697c5

Command Prompt

```
data-volume-database-0   Pending                                      database       3m29s

C:\Users\khoadangq>kubectl get services -n argocd
NAME                                       TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)                      AGE
argocd-applicationset-controller           ClusterIP   172.20.132.147   <none>        7000/TCP,8080/TCP            131m
argocd-dex-server                          ClusterIP   172.20.19.37     <none>        5556/TCP,5557/TCP,5558/TCP   131m
argocd-metrics                             ClusterIP   172.20.86.187    <none>        8082/TCP                     131m
argocd-notifications-controller-metrics    ClusterIP   172.20.201.16    <none>        9001/TCP                     131m
argocd-redis                               ClusterIP   172.20.207.214   <none>        6379/TCP                     131m
argocd-repo-server                         ClusterIP   172.20.234.23    <none>        8081/TCP,8084/TCP            131m
argocd-server                              ClusterIP   172.20.1.208     <none>        80/TCP,443/TCP               131m
argocd-server-metrics                      ClusterIP   172.20.82.210    <none>        8083/TCP                     131m
backend                                    ClusterIP   172.20.218.205   <none>        3000/TCP                     27m
frontend                                   ClusterIP   172.20.27.35     <none>        3000/TCP                     27m
mongo                                      ClusterIP   172.20.203.48    <none>        27017/TCP                    27m

C:\Users\khoadangq>
```

After all services were “running” we need to port forward to using application on local



Command Prompt - kubectl port-forward svc/frontend -n argocd 8888:3000

```
error: Service frontend does not have a service port 8080

C:\Users\khoadangq>kubectl port-forward svc/frontend -n argocd 8888:3000
Forwarding from 127.0.0.1:8888 -> 3000
Forwarding from [::1]:8888 -> 3000
Handling connection for 8888
Handling connection for 8888
Handling connection for 8888
Handling connection for 8888
Handling connection for 8888
Handling connection for 8888
Handling connection for 8888
Handling connection for 8888
Handling connection for 8888
Handling connection for 8888
Handling connection for 8888
```

2. Provision AWS resources

Use https://github.com/nashtech-garage/terraform-demo as a reference.

Use Terraform to provision VPC, EC2, ECR and EKS on AWS

DevOps for Devs:

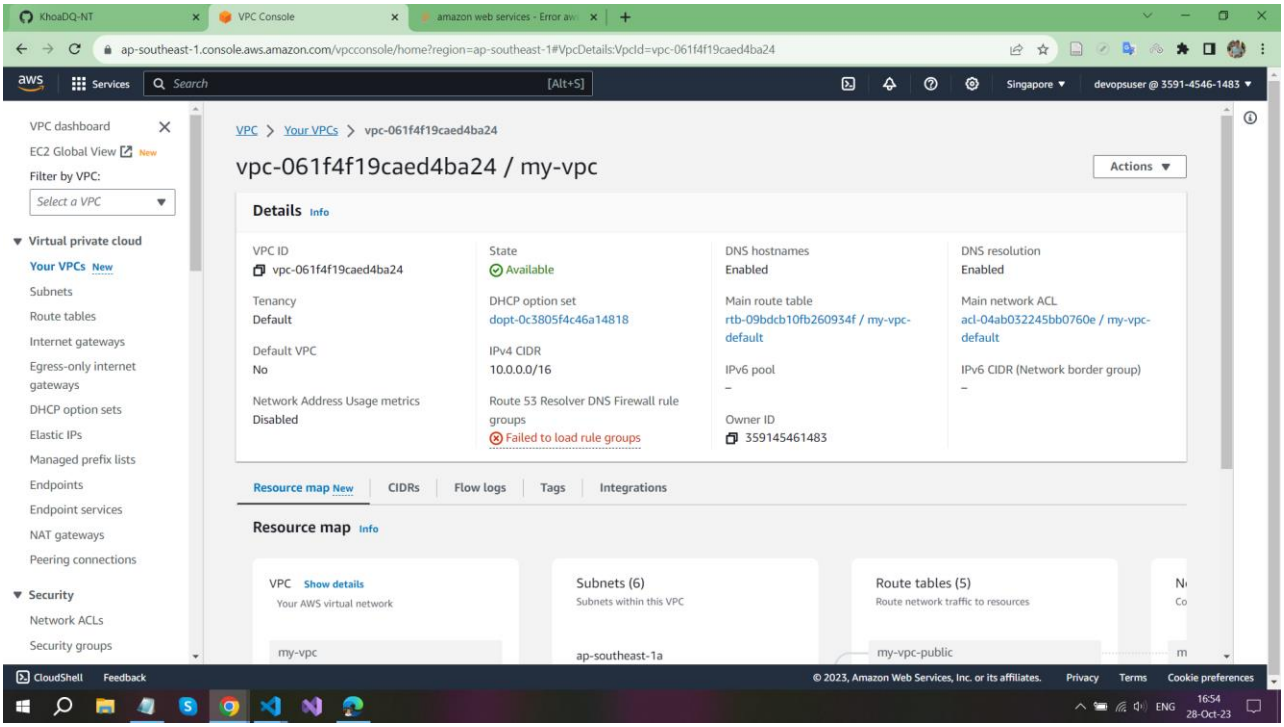
- EKS can created in one zone and
- Can use default VPC for worker nodes

Pi-sharp:

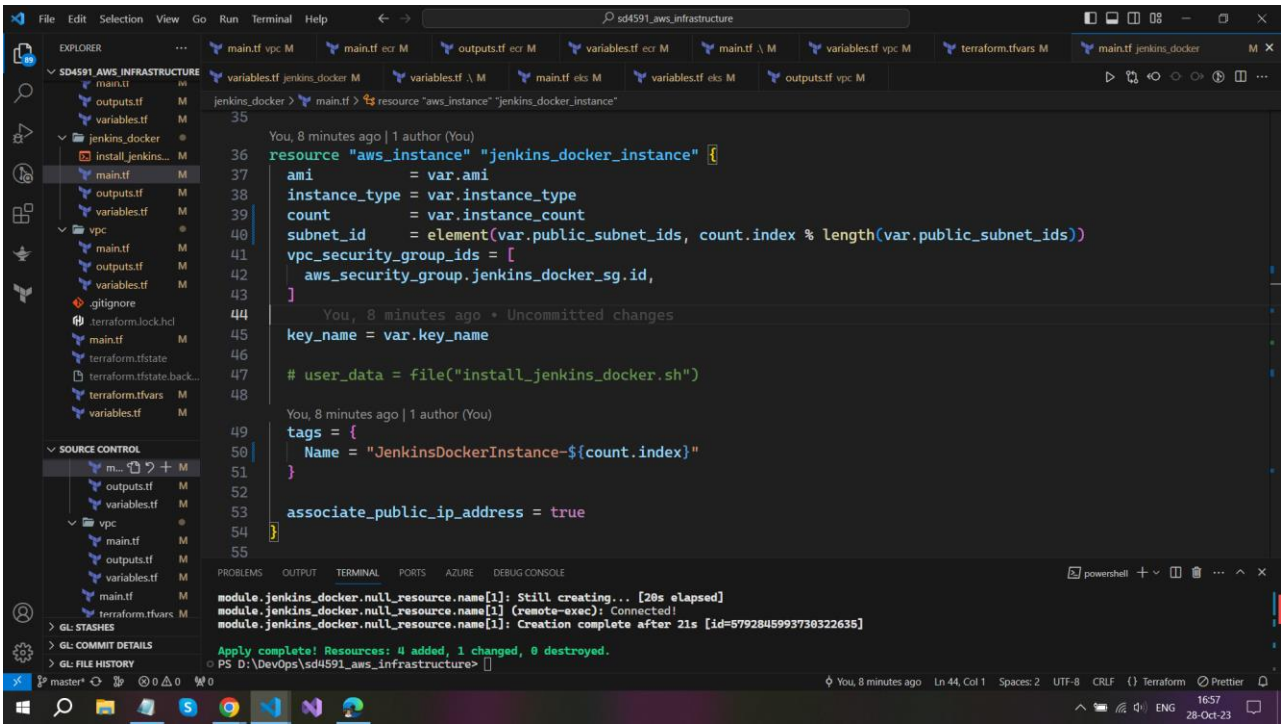
- EKS must be created in High Availability (Use Multi-AZs)
- Create your VPC and use it for worker nodes

I need to create a custom VPC as requirement, so that I have made some changes on my Infrastructure source code for using Terraform to deploy all necessary AWS services

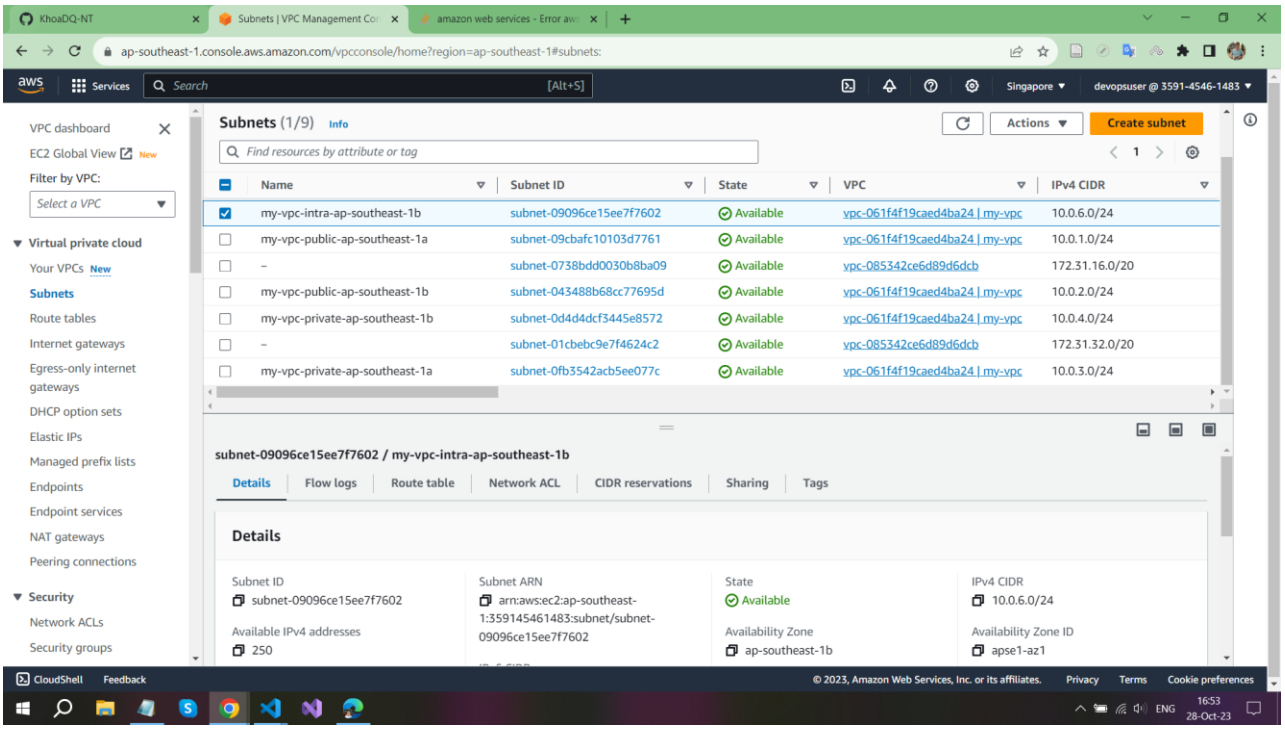
The I mage below shows you my custom VPC

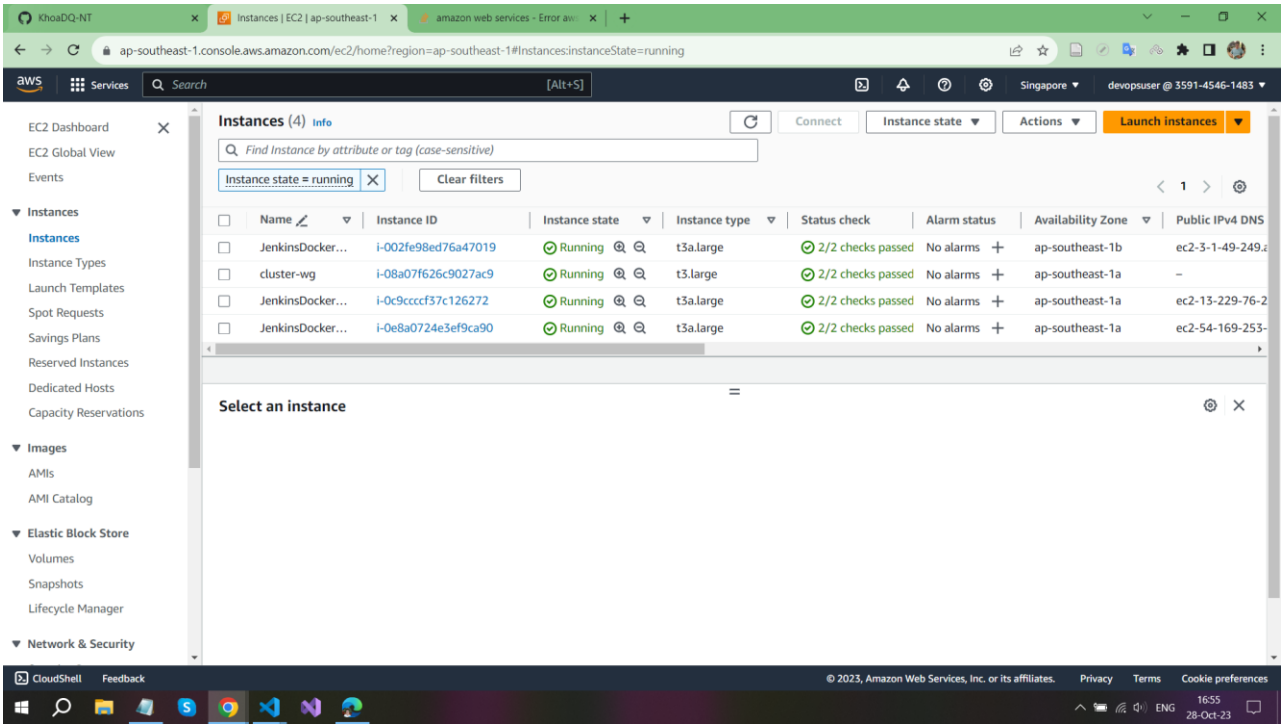


The EC2 instances will be deployed in multiple zones within a region, that will make our instances more “High availability”, the ability to operate continuously without failing for a designated period of time. The main code changes on creating EC2 is showed below



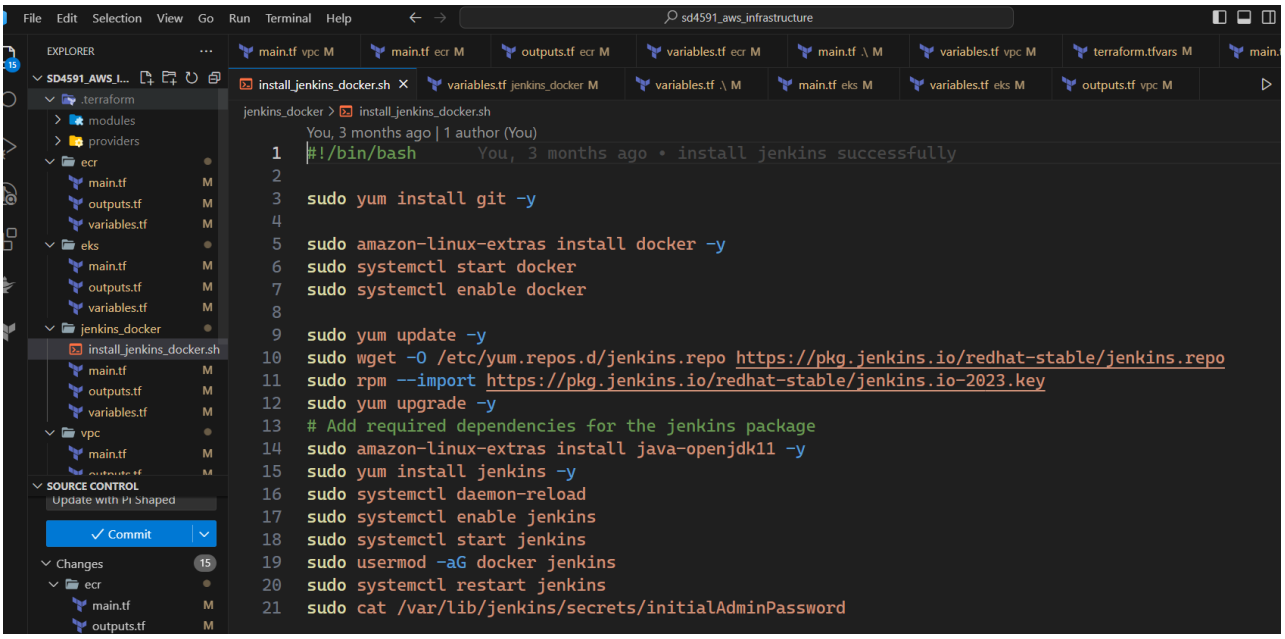
My subnets on multiple zones

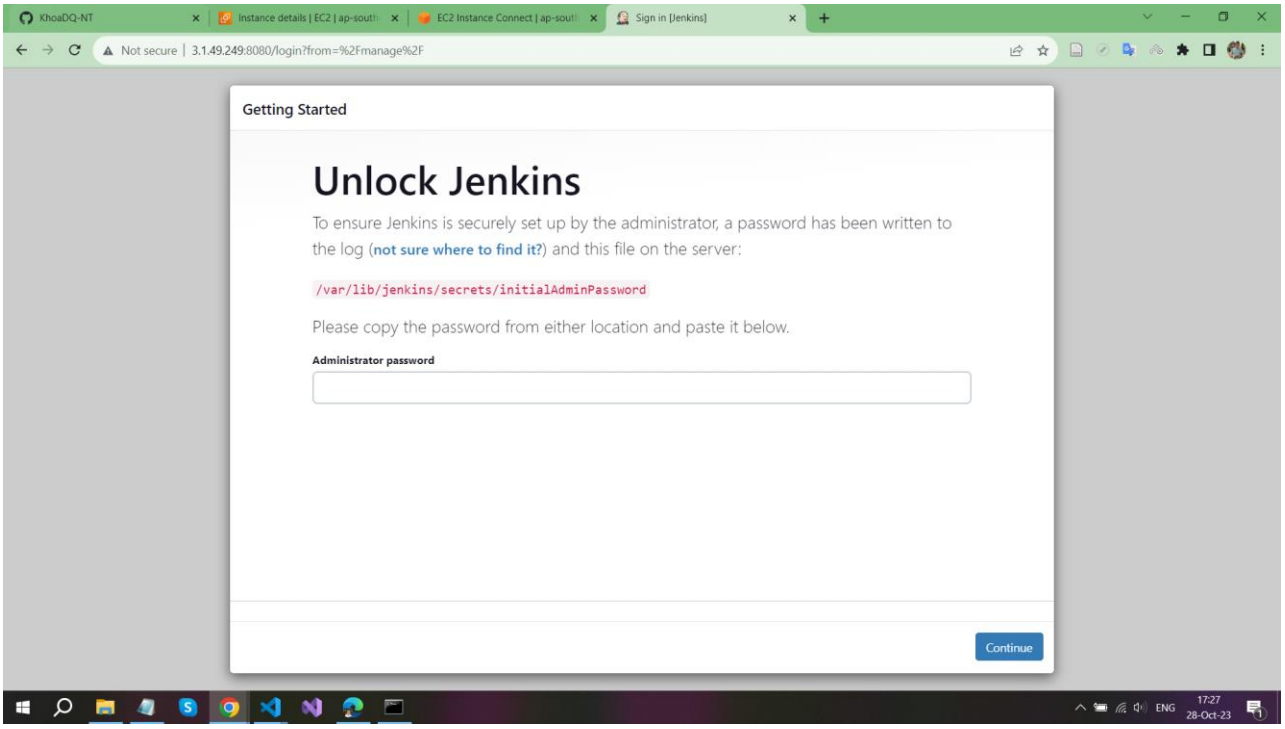


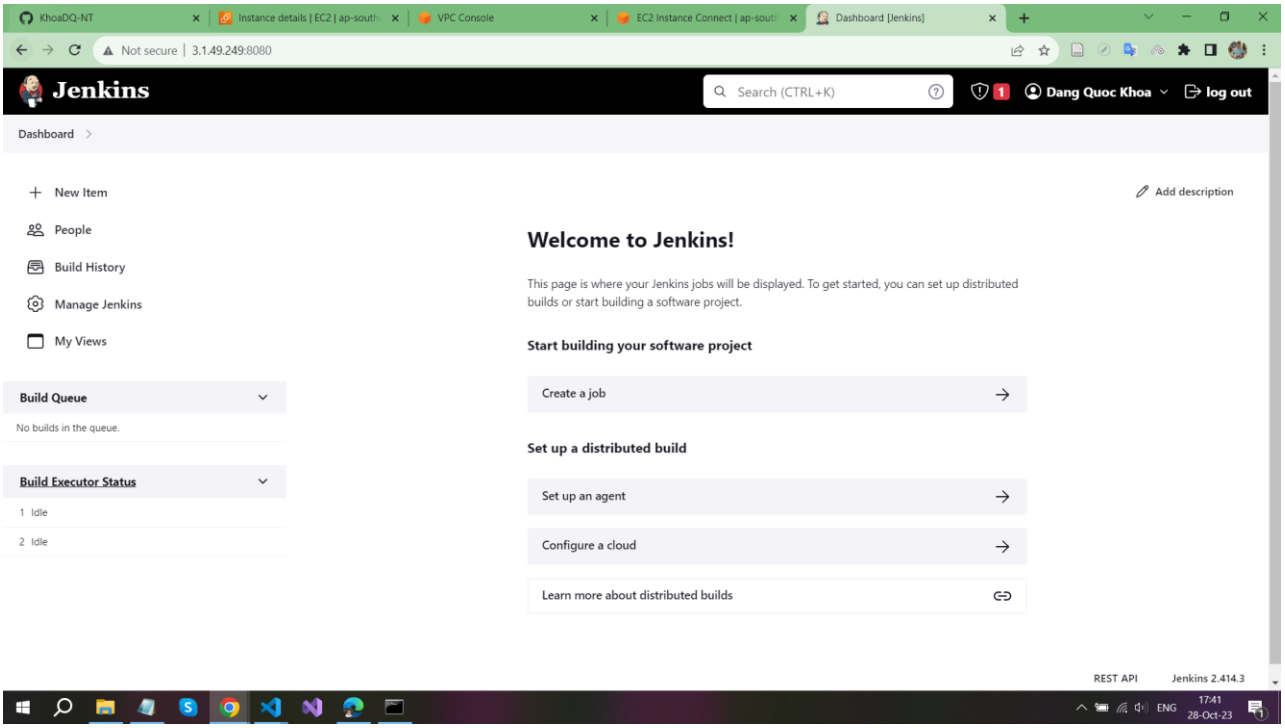

KhoaDQ-NT
Instances | EC2 | ap-southeast-1
amazon web services - Error aw
ap-southeast-1.console.aws.amazon.com/ec2/home?region=ap-southeast-1#Instances:instanceState=running
aws
Services
Search
[Alt+S]
Singapore
devopsuser @ 3591-4546-1483
EC2 Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Instances (4) Info
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
Instance state = running
Clear filters
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IPv4 DNS
JenkinsDocker... i-002fe98ed76a47019 Running t3a.large 2/2 checks passed No alarms ap-southeast-1b ec2-3-1-49-249.
cluster-wg i-08a07f626c9027ac9 Running t3.large 2/2 checks passed No alarms ap-southeast-1a –
JenkinsDocker... i-0c9ccccf37c126272 Running t3a.large 2/2 checks passed No alarms ap-southeast-1a ec2-13-229-76-2
JenkinsDocker... i-0e8a0724e3ef9ca90 Running t3a.large 2/2 checks passed No alarms ap-southeast-1a ec2-54-169-253-
Select an instance
CloudShell
Feedback
© 2023, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
ENG
16:55
28-Oct-23

3. Installation

You need to have Docker and Jenkins servers installed on EC2.







After Jenkins have been installed, we need to add some needed blugins and create some necessary credentials.

4. Setup Jenkins pipeline for CI

Use https://github.com/nashtech-garage/devops-ci-cd as a reference.

DevOps for Devs:

- You can use normal Jenkins pipeline or create Jenkins pipeline using Groovy.

Pi-sharp:

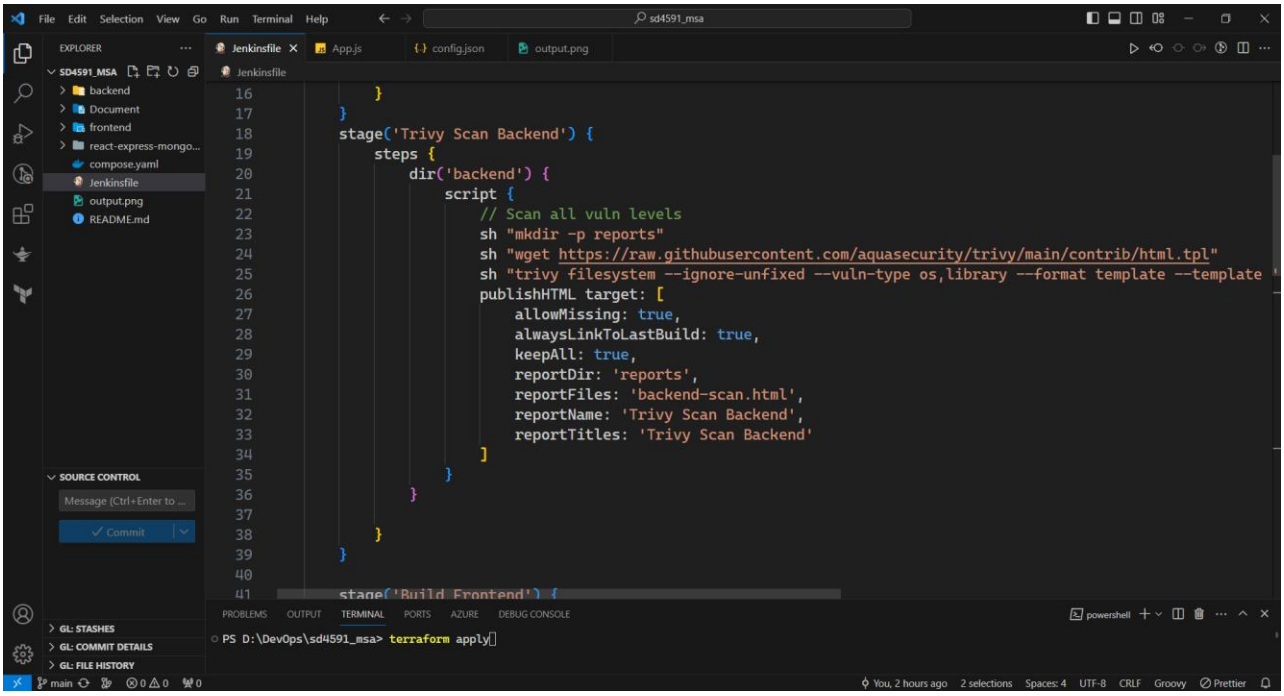
- Use Trivy in Jenkins CI pipeline.

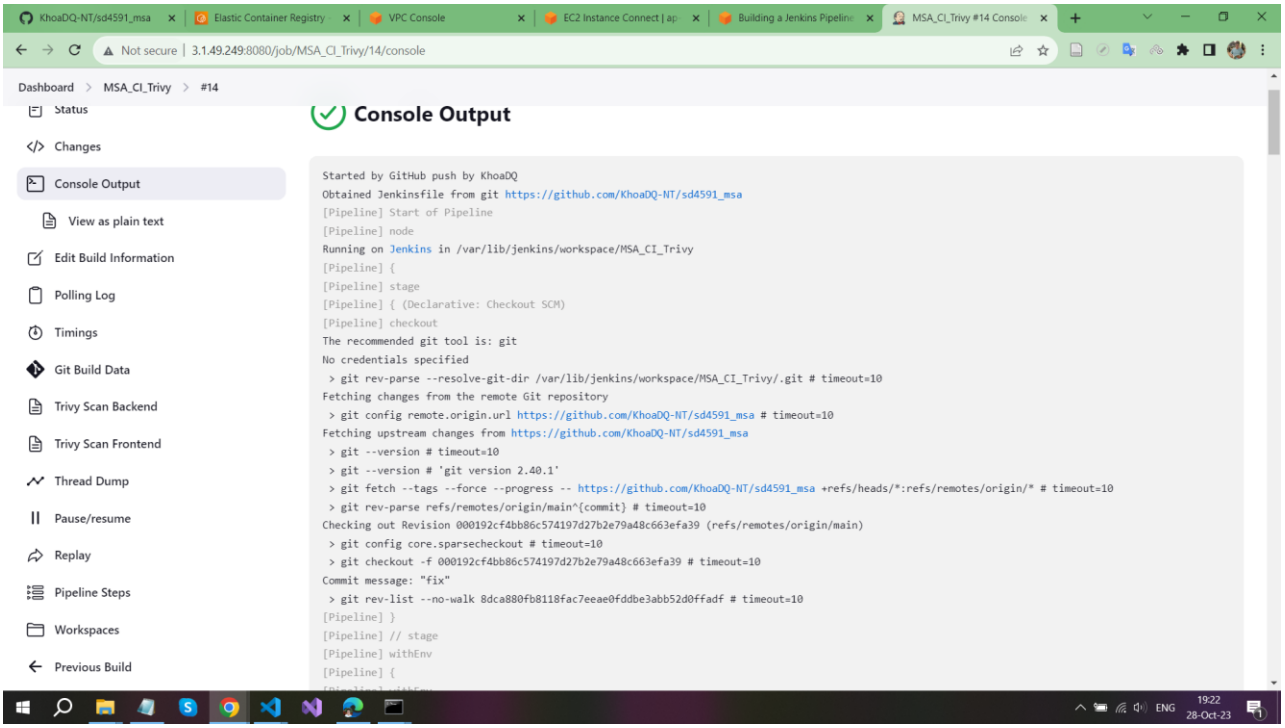
Trivy should be installed before running the Jenkins CI pipeline

curl -sfL https://raw.githubusercontent.com/aquasecurity/trivy/main/contrib/install.sh | sudo sh -s -- -b /usr/local/bin v0.18.3

curl -sfL https://raw.githubusercontent.com/aquasecurity/trivy/main/contrib/html.tpl > html.tpl

Besides, I have changed MSA source code to add more stages related to Trivy and use it on Jenkins CI

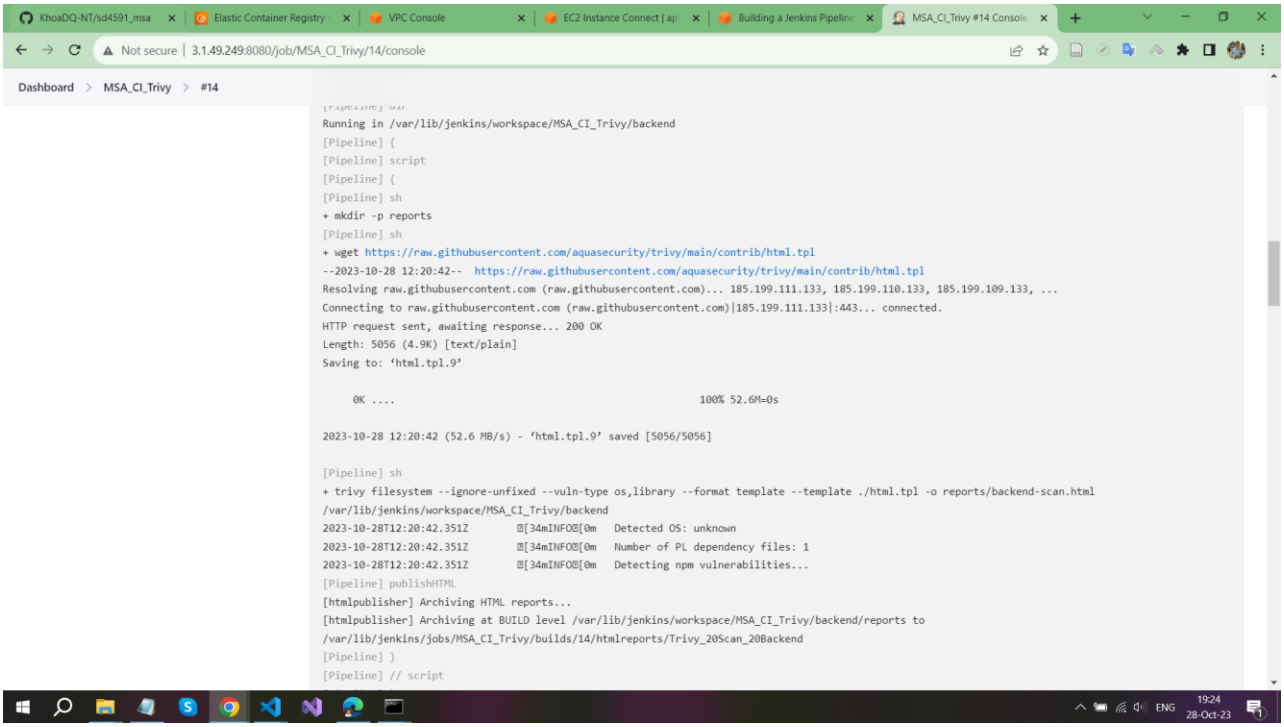



KhoaDQ-NT/sd4591_msa
Elastic Container Registry
VPC Console
EC2 Instance Connect | ap
Building a Jenkins Pipeline
MSA_CI_Trivy #14 Console
Not secure | 3.1.49.249:8080/job/MSA_CI_Trivy/14/console
Dashboard > MSA_CI_Trivy > #14
Status
Changes
Console Output
View as plain text
Edit Build Information
Polling Log
Timings
Git Build Data
Trivy Scan Backend
Trivy Scan Frontend
Thread Dump
Pause/resume
Replay
Pipeline Steps
Workspaces
Previous Build
Console Output
Started by GitHub push by KhoaDQ
Obtained Jenkinsfile from git https://github.com/KhoaDQ-NT/sd4591_msa
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/MSA_CI_Trivy
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
The recommended git tool is: git
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/MSA_CI_Trivy/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/KhoaDQ-NT/sd4591_msa # timeout=10
Fetching upstream changes from https://github.com/KhoaDQ-NT/sd4591_msa
> git --version # timeout=10
> git --version # 'git version 2.40.1'
> git fetch --tags --force --progress -- https://github.com/KhoaDQ-NT/sd4591_msa +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 000192cf4bb86c574197d27b2e79a48c663efa39 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 000192cf4bb86c574197d27b2e79a48c663efa39 # timeout=10
Commit message: "fix"
> git rev-list --no-walk 8dca880fb8118fac7eeae0fddbe3abb52d0ffadf # timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
ENG
19:22
28-Oct-23

Build backend

```
Running in /var/lib/jenkins/workspace/MSA_CI_Trivy/backend
[Pipeline] {
[Pipeline] script
[Pipeline] {
[Pipeline] sh
+ docker build -t 359145461483.dkr.ecr.ap-southeast-1.amazonaws.com/my-ecr-repo-devops/backend:latest .
Sending build context to Docker daemon  142.8kB
```

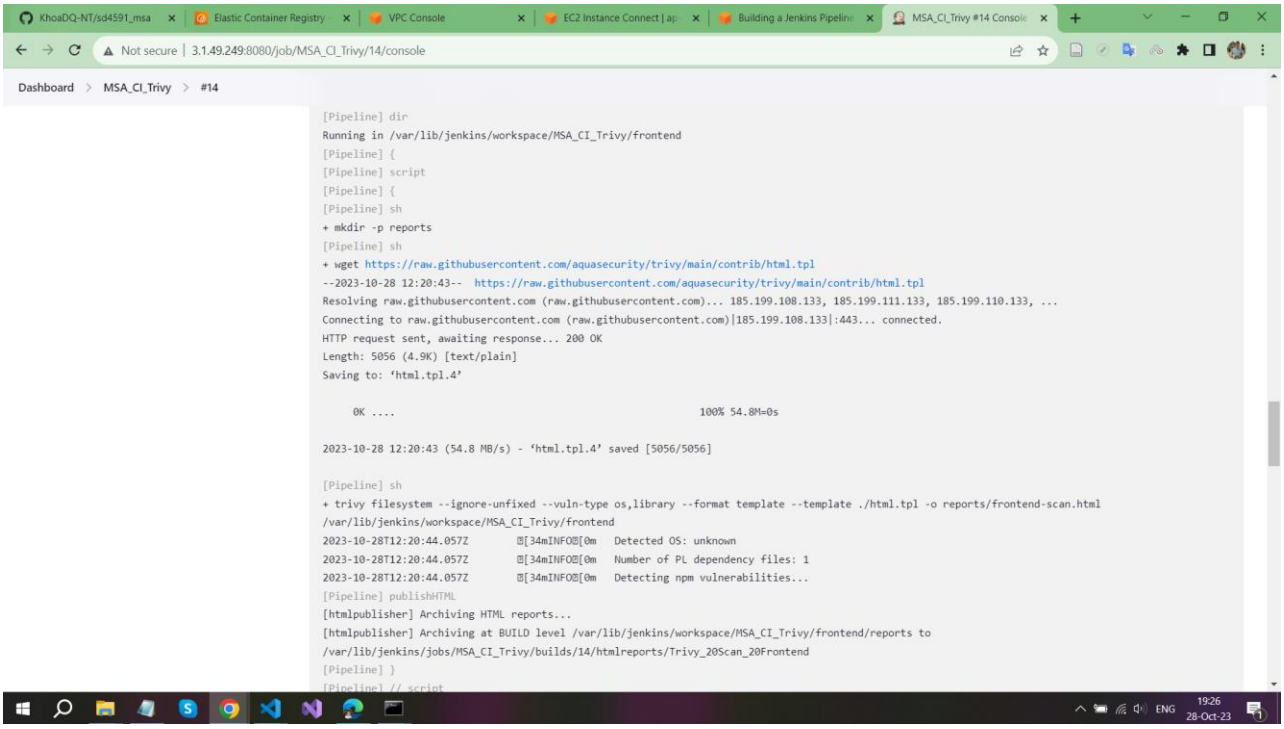
Use Trivy scanner on Backend



Build frontend

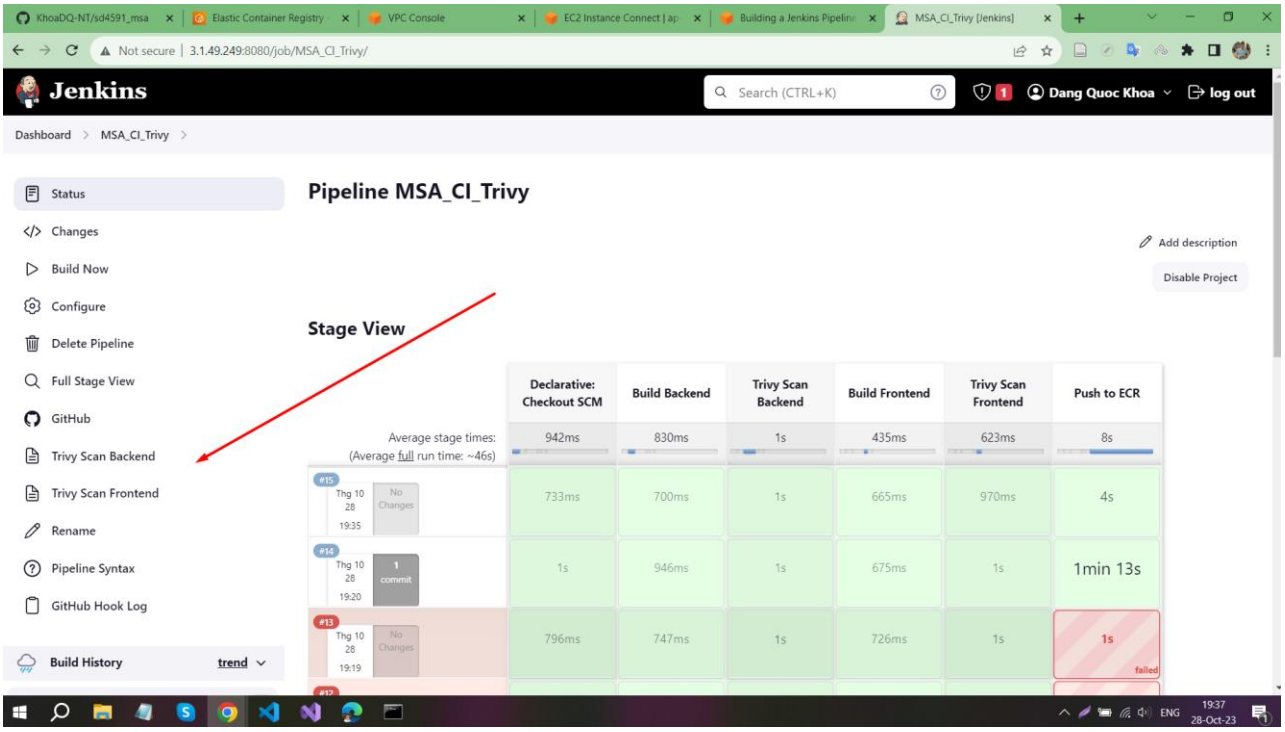
```
Running in /var/lib/jenkins/workspace/MSA_CI_Trivy/frontend
[Pipeline] {
[Pipeline] script
[Pipeline] {
[Pipeline] sh
+ docker build -t 359145461483.dkr.ecr.ap-southeast-1.amazonaws.com/my-ecr-repo-devops/frontend:latest .
Sending build context to Docker daemon  600.6kB
```

User Trivy scanner on frontend



Push image into ECR and complete CI pipeline

```
Constructing AWS CredentialsSetting AWS region ap-southeast-1
 [Pipeline] {
[Pipeline] sh
+ aws ecr get-login-password --region ap-southeast-1
+ docker login --username AWS --password-stdin 359145461483.dkr.ecr.ap-southeast-1.amazonaws.com/my-ecr-repo-devops/backend
WARNING! Your password will be stored unencrypted in /var/lib/jenkins/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[Pipeline] sh
+ docker push 359145461483.dkr.ecr.ap-southeast-1.amazonaws.com/my-ecr-repo-devops/backend:latest
The push refers to repository [359145461483.dkr.ecr.ap-southeast-1.amazonaws.com/my-ecr-repo-devops/backend]
20c1f73a02c5: Preparing
1c96f4860b84: Pushed
3142bf4692e1: Pushed
latest: digest: sha256:660b924c9a3dd462ea5b3df1d61a235ab0bda5674acfc55160c31516fb66e12d size: 2411
[Pipeline] sh
+ docker login --username AWS --password-stdin 359145461483.dkr.ecr.ap-southeast-1.amazonaws.com/my-ecr-repo-devops/frontend
+ aws ecr get-login-password --region ap-southeast-1
WARNING! Your password will be stored unencrypted in /var/lib/jenkins/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[Pipeline] sh
+ docker push 359145461483.dkr.ecr.ap-southeast-1.amazonaws.com/my-ecr-repo-devops/frontend:latest
The push refers to repository [359145461483.dkr.ecr.ap-southeast-1.amazonaws.com/my-ecr-repo-devops/frontend]
84a4c41623c5: Preparing
a59aff4fa371: Pushed
6a24eb7db86f: Pushed
latest: digest: sha256:6a985261d54223c8b8309e06bcaa83f56d0a7b61fa489993a71f77c7a17272fc size: 3051
[Pipeline] }
[Pipeline] // withAWS
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```


Jenkins
Search (CTRL+K)
Dang Quoc Khoa
log out
Dashboard
MSA_CI_Trivy
Status
Changes
Build Now
Configure
Delete Pipeline
Full Stage View
GitHub
Trivy Scan Backend
Trivy Scan Frontend
Rename
Pipeline Syntax
GitHub Hook Log
Build History
trend
Pipeline MSA_CI_Trivy
Add description
Disable Project
Stage View
Declarative: Checkout SCM
Build Backend
Trivy Scan Backend
Build Frontend
Trivy Scan Frontend
Push to ECR
Average stage times:
(Average full run time: ~46s)
942ms
830ms
1s
435ms
623ms
8s
#15
Thg 10 28
19:35
No Changes
733ms
700ms
1s
665ms
970ms
4s
#14
Thg 10 28
19:20
1 commit
1s
946ms
1s
675ms
1s
1min 13s
#13
Thg 10 28
19:19
No Changes
796ms
747ms
1s
726ms
1s
1s
failed
ENG
19:37
28-Oct-23

5. Monitoring

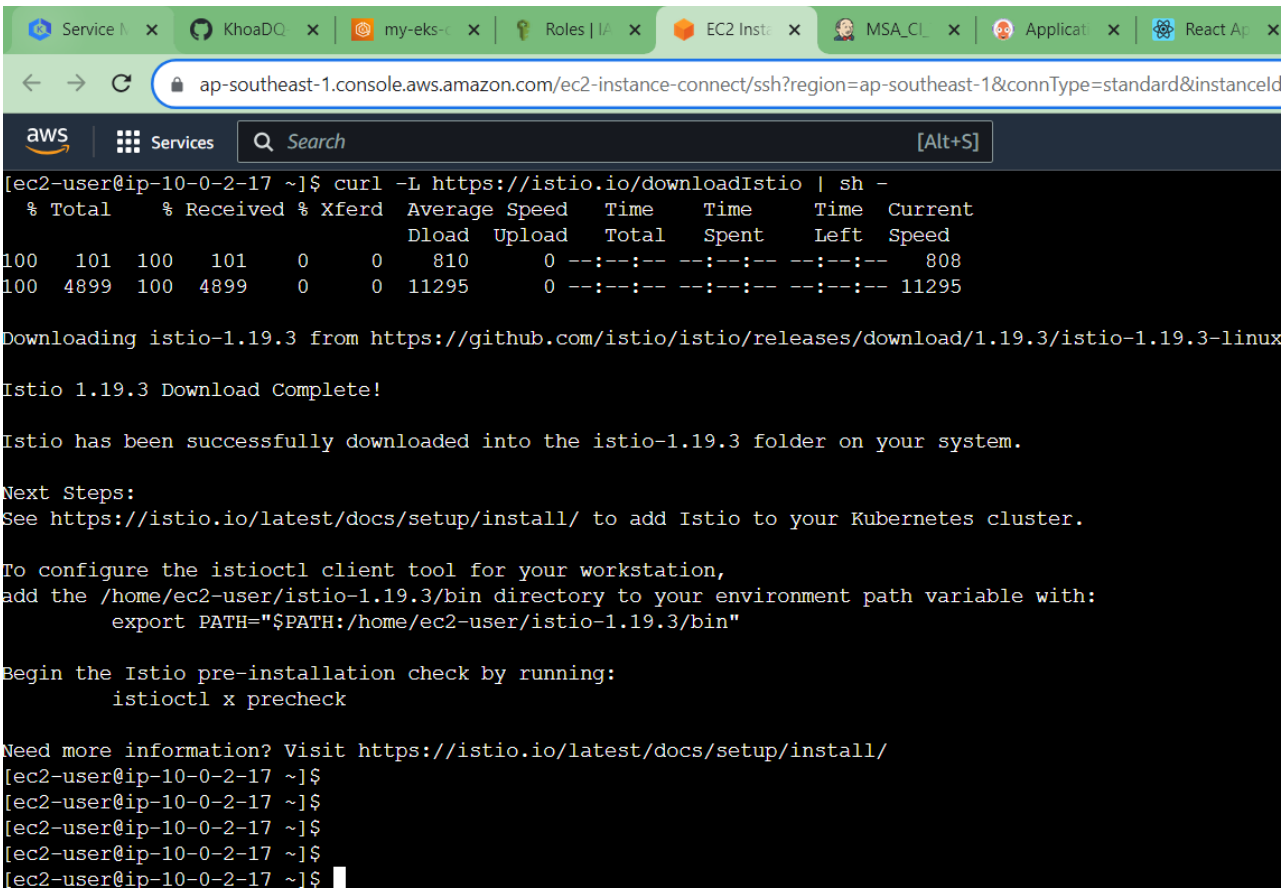
DevOps for Devs:

Set up Prometheus and Grafana to monitor EKS resources and default EKS resource metric.

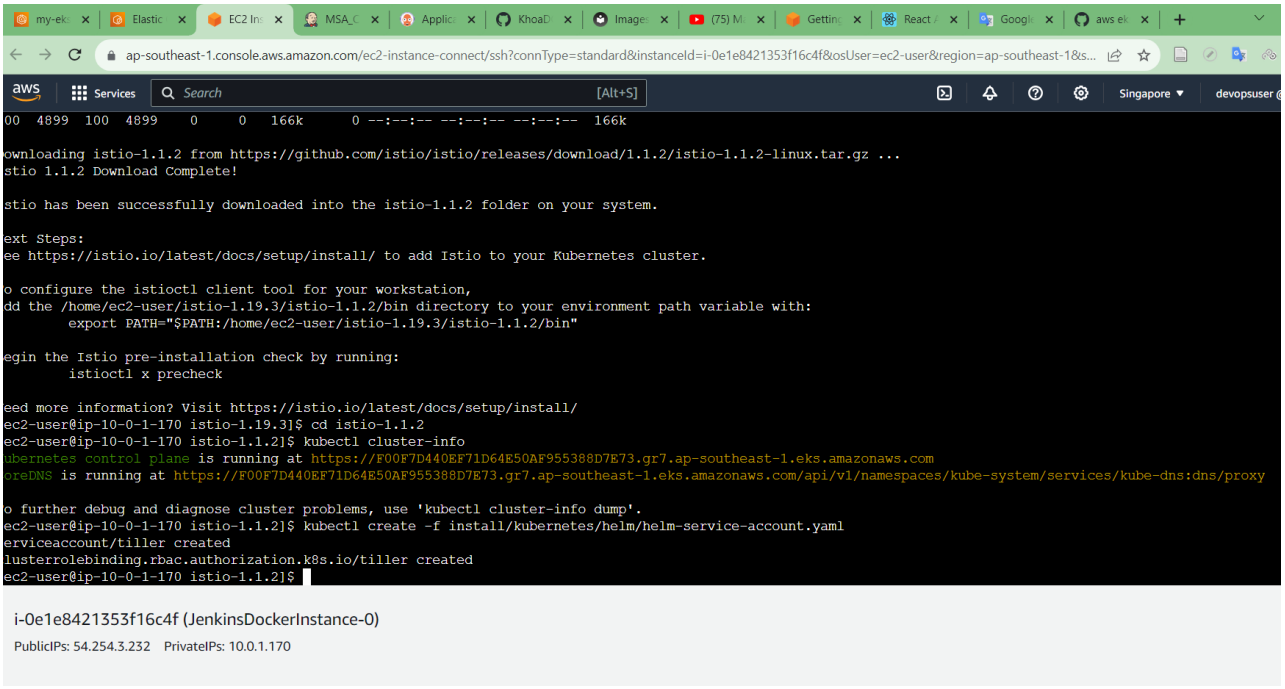
Pi-sharp:

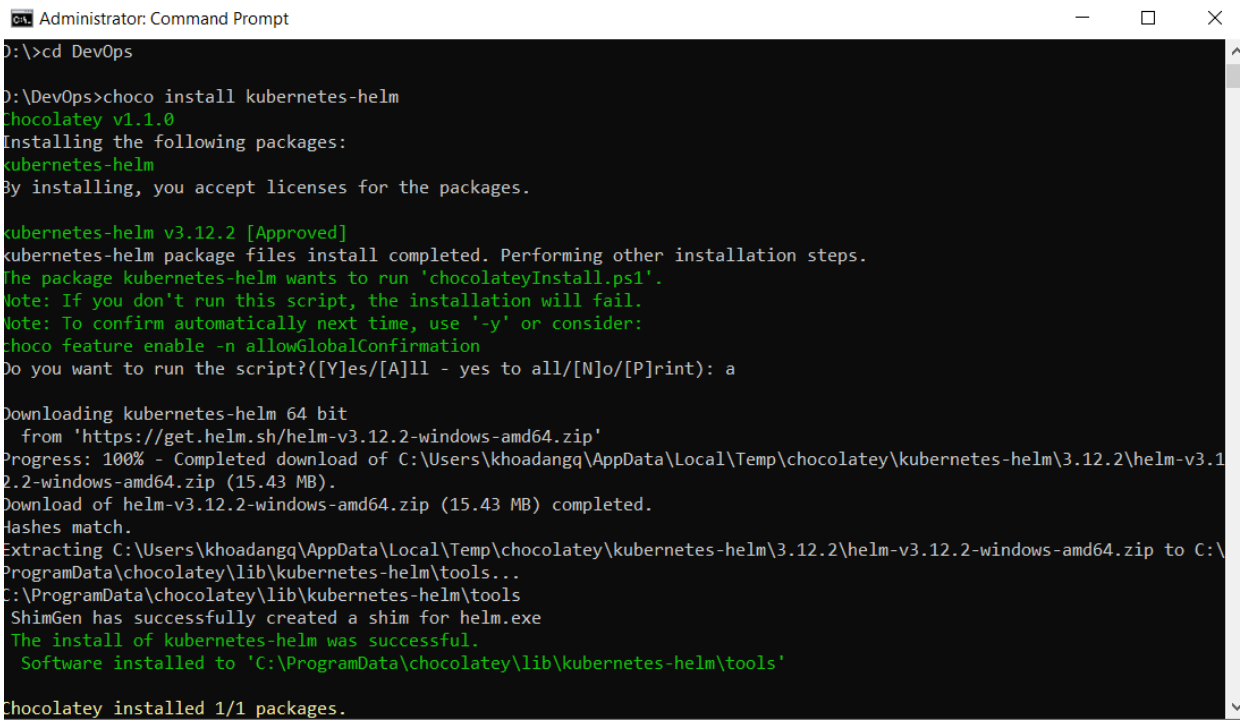
Use Istio to provide observability metrics, which can be visualized and collected using Grafana and Prometheus.

Download Istio Deployment Files



Configure Helm

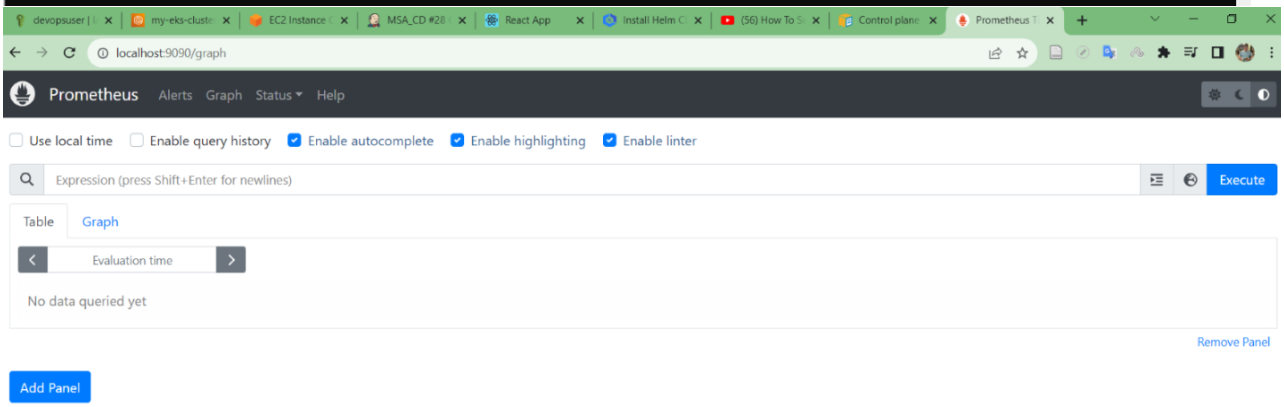


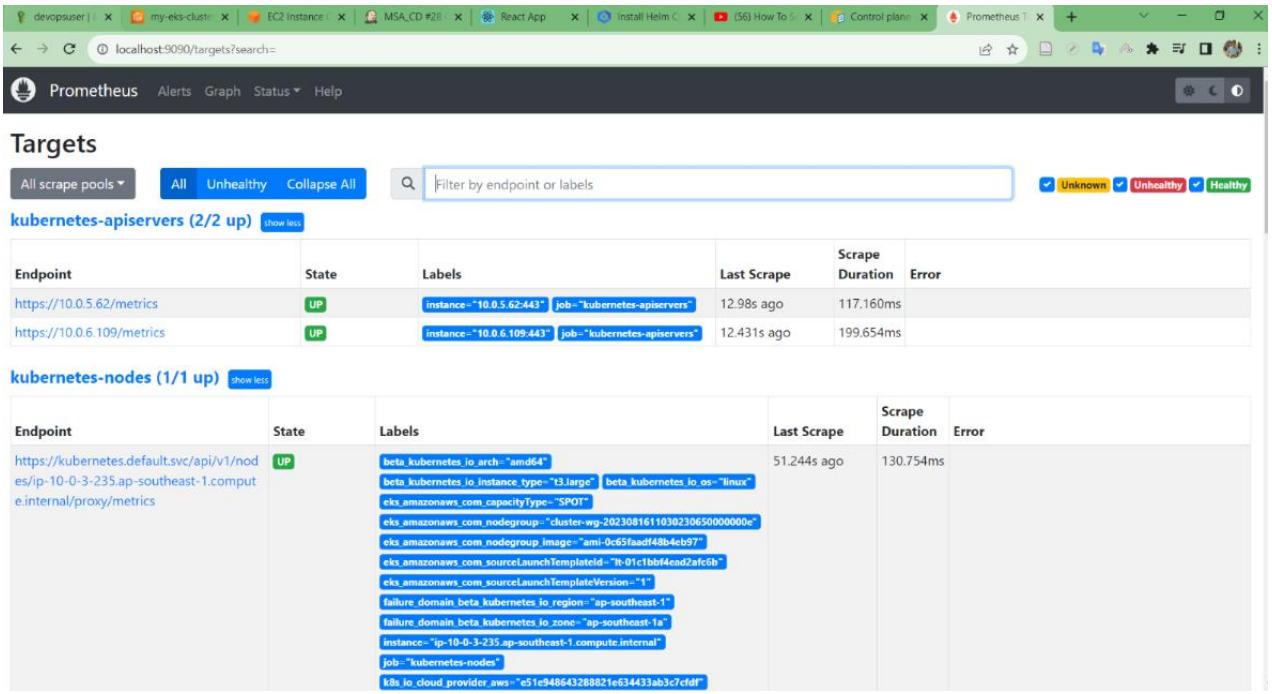


Prometheus

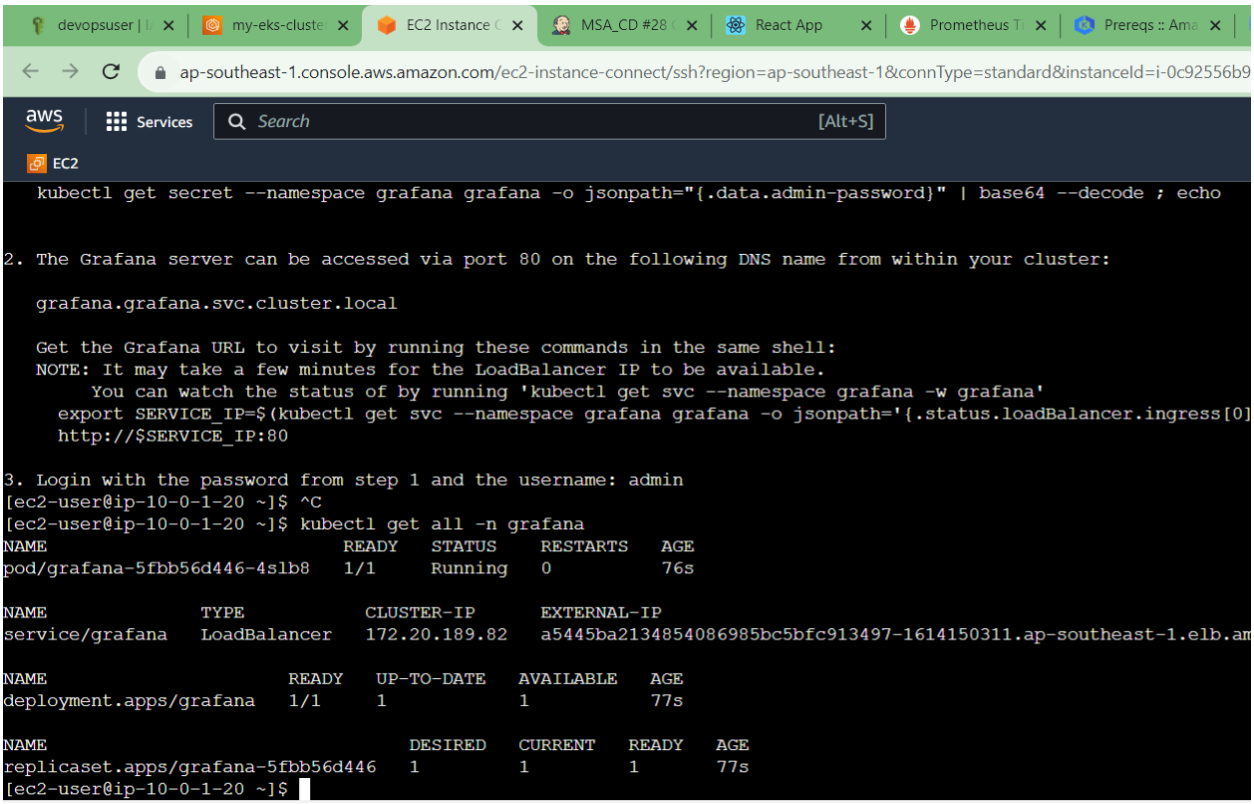
Administrator: Command Prompt - kubectl --namespace=prometheus port-forward deploy/prometheus-server 9090

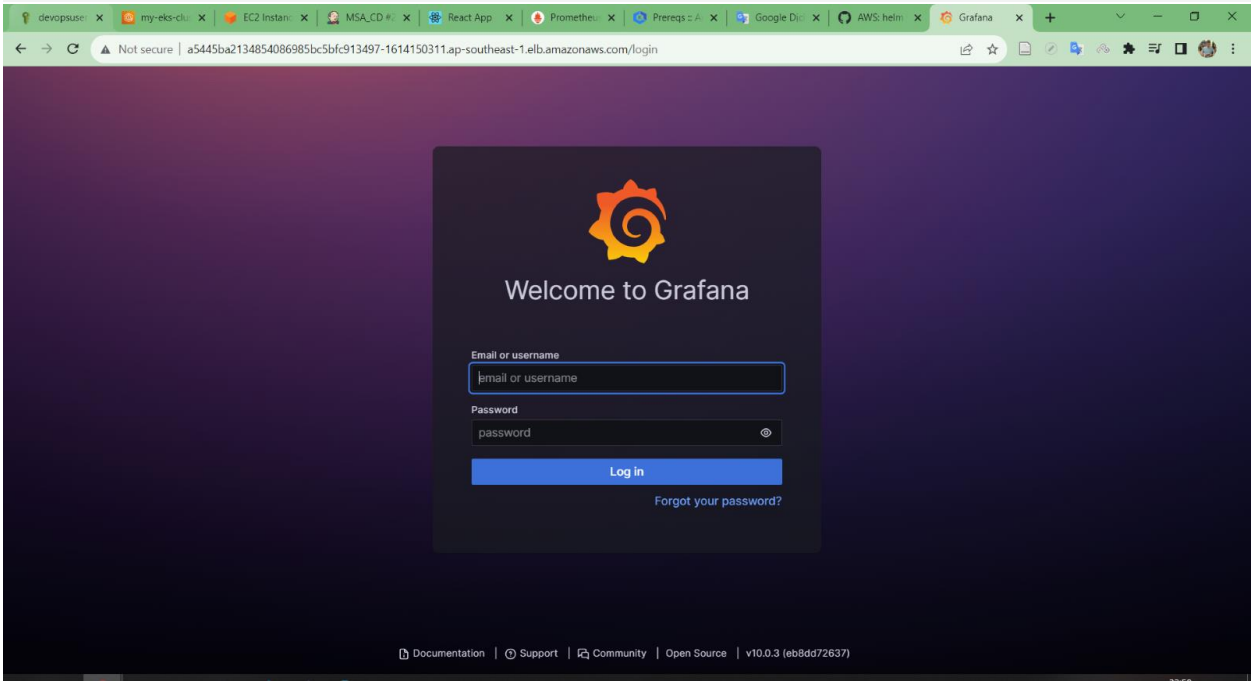
```
NAME                                            READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/prometheus-kube-state-metrics   1/1     1            1           104s
deployment.apps/prometheus-prometheus-pushgateway   1/1     1            1           104s
deployment.apps/prometheus-server               1/1     1            1           104s

NAME                                                  DESIRED   CURRENT   READY   AGE
replicaset.apps/prometheus-kube-state-metrics-69c8887cfd       1         1         1       104s
replicaset.apps/prometheus-prometheus-pushgateway-79ff799669   1         1         1       104s
replicaset.apps/prometheus-server-855f6c967f                   1         1         1       104s

NAME                                       READY   AGE
statefulset.apps/prometheus-alertmanager   0/1     104s

D:\DevOps>kubectl get pods -n prometheus
NAME                                             READY   STATUS    RESTARTS   AGE
prometheus-alertmanager-0                        1/1     Running   0          5m50s
prometheus-kube-state-metrics-69c8887cfd-zrcnn   1/1     Running   0          5m50s
prometheus-prometheus-node-exporter-n5xc4        1/1     Running   0          5m51s
prometheus-prometheus-pushgateway-79ff799669-6h6n2   1/1     Running   0          5m50s
prometheus-server-855f6c967f-xmb6t               2/2     Running   0          5m50s

D:\DevOps>kubectl --namespace=prometheus port-forward deploy/prometheus-server 9090
Forwarding from 127.0.0.1:9090 -> 9090
Forwarding from [::1]:9090 -> 9090
```

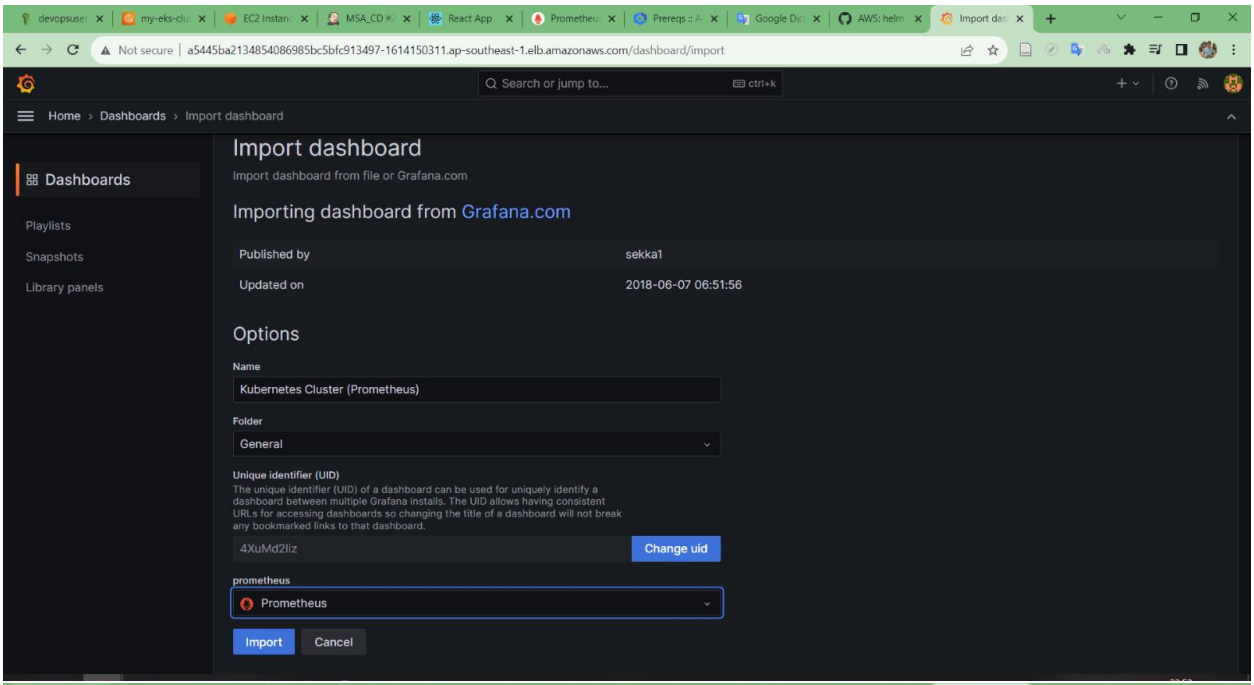


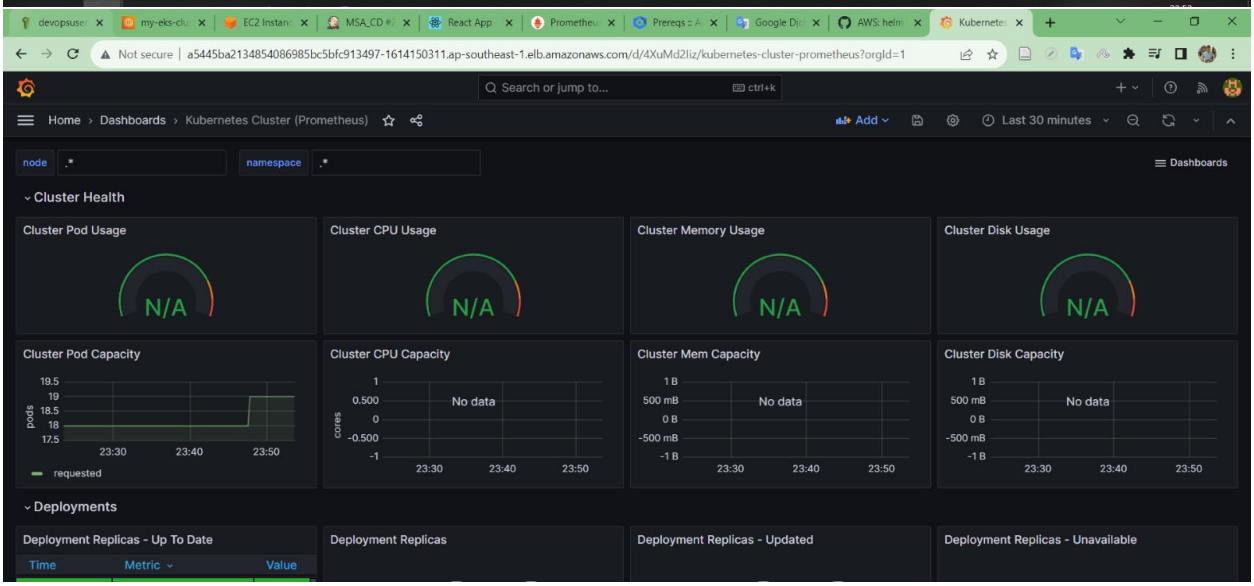

devopsuser
my-eks-cluste
EC2 Instance
MSA_CD #28
React App
Install Helm
(56) How To
Control plane
Prometheus
localhost:9090/targets?search=
Prometheus
Alerts
Graph
Status
Help
Targets
All scrape pools
All
Unhealthy
Collapse All
Filter by endpoint or labels
Unknown
Unhealthy
Healthy
kubernetes-apiservers (2/2 up)
show less
Endpoint
State
Labels
Last Scrape
Scrape Duration
Error
https://10.0.5.62/metrics
UP
instance="10.0.5.62:443"
job="kubernetes-apiservers"
12.98s ago
117.160ms
https://10.0.6.109/metrics
UP
instance="10.0.6.109:443"
job="kubernetes-apiservers"
12.431s ago
199.654ms
kubernetes-nodes (1/1 up)
show less
Endpoint
State
Labels
Last Scrape
Scrape Duration
Error
https://kubernetes.default.svc/api/v1/nodes/ip-10-0-3-235.ap-southeast-1.compute.internal/proxy/metrics
UP
beta_kubernetes_io_arch="amd64"
beta_kubernetes_io_instance_type="t3.large"
beta_kubernetes_io_os="linux"
eks_amazonaws_com_capacityType="SPOT"
eks_amazonaws_com_nodegroup="cluster-wg-20230816110302306500000000e"
eks_amazonaws_com_nodegroup_image="ami-0c65faadf48b4eb97"
eks_amazonaws_com_sourceLaunchTemplateId="lt-01c1bbf4ead2afc6b"
eks_amazonaws_com_sourceLaunchTemplateVersion="1"
failure_domain_beta_kubernetes_io_region="ap-southeast-1"
failure_domain_beta_kubernetes_io_zone="ap-southeast-1a"
instance="ip-10-0-3-235.ap-southeast-1.compute.internal"
job="kubernetes-nodes"
k8s_io_cloud_provider_aws="e51e948643288821e634433ab3c7cfdf"
51.244s ago
130.754ms

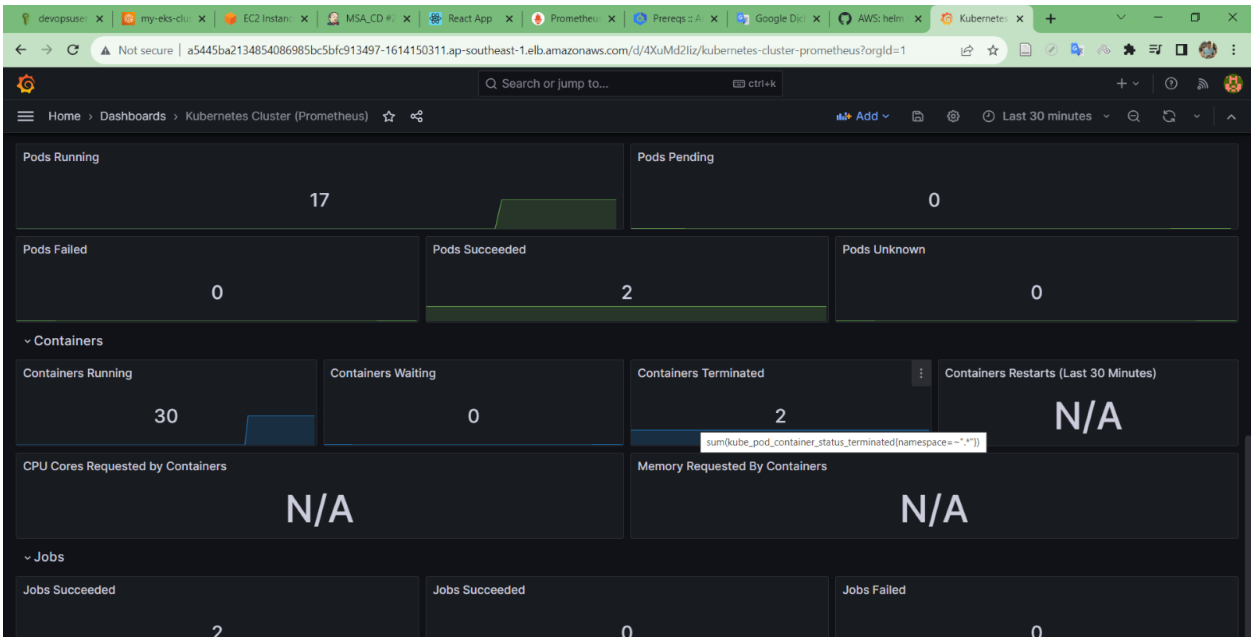
Grapfana


devopsuser
my-eks-cluste
EC2 Instance
MSA_CD #28
React App
Prometheus
Prereqs :: Ama
ap-southeast-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=ap-southeast-1&connType=standard&instanceId=i-0c92556b9
aws
Services
Search
[Alt+S]
EC2
kubectl get secret --namespace grafana grafana -o jsonpath="{.data.admin-password}" | base64 --decode ; echo
2. The Grafana server can be accessed via port 80 on the following DNS name from within your cluster:
grafana.grafana.svc.cluster.local
Get the Grafana URL to visit by running these commands in the same shell:
NOTE: It may take a few minutes for the LoadBalancer IP to be available.
You can watch the status of by running 'kubectl get svc --namespace grafana -w grafana'
export SERVICE_IP=$(kubectl get svc --namespace grafana grafana -o jsonpath='{.status.loadBalancer.ingress[0]
http://$SERVICE_IP:80
3. Login with the password from step 1 and the username: admin
[ec2-user@ip-10-0-1-20 ~]$ ^C
[ec2-user@ip-10-0-1-20 ~]$ kubectl get all -n grafana
NAME READY STATUS RESTARTS AGE
pod/grafana-5fbb56d446-4slb8 1/1 Running 0 76s
NAME TYPE CLUSTER-IP EXTERNAL-IP
service/grafana LoadBalancer 172.20.189.82 a5445ba2134854086985bc5bfc913497-1614150311.ap-southeast-1.elb.am
NAME READY UP-TO-DATE AVAILABLE AGE
deployment.apps/grafana 1/1 1 1 77s
NAME DESIRED CURRENT READY AGE
replicaset.apps/grafana-5fbb56d446 1 1 1 77s
[ec2-user@ip-10-0-1-20 ~]$


Not secure
a5445ba2134854086985bc5bfc913497-1614150311.ap-southeast-1.elb.amazonaws.com/login
Grafana
Welcome to Grafana
Email or username
email or username
Password
password
Log in
Forgot your password?
Documentation
Support
Community
Open Source
v10.0.3 (eb8dd72637)


Not secure | a5445ba2134854086985bc5bfc913497-1614150311.ap-southeast-1.elb.amazonaws.com/dashboard/import
Search or jump to... ctrl+k
Home › Dashboards › Import dashboard
Dashboards
Playlists
Snapshots
Library panels
Import dashboard
Import dashboard from file or Grafana.com
Importing dashboard from Grafana.com
Published by
sekka1
Updated on
2018-06-07 06:51:56
Options
Name
Kubernetes Cluster (Prometheus)
Folder
General
Unique identifier (UID)
The unique identifier (UID) of a dashboard can be used for uniquely identify a dashboard between multiple Grafana installs. The UID allows having consistent URLs for accessing dashboards so changing the title of a dashboard will not break any bookmarked links to that dashboard.
4XuMd2liz
Change uid
prometheus
Prometheus
Import
Cancel


Not secure | a5445ba2134854086985bc5bfc913497-1614150311.ap-southeast-1.elb.amazonaws.com/d/4XuMd2liz/kubernetes-cluster-prometheus?orgId=1
Search or jump to... ctrl+k
Home › Dashboards › Kubernetes Cluster (Prometheus)
Add
Last 30 minutes
node .*
namespace .*
Dashboards
Cluster Health
Cluster Pod Usage
N/A
Cluster CPU Usage
N/A
Cluster Memory Usage
N/A
Cluster Disk Usage
N/A
Cluster Pod Capacity
19.5
19
18.5
18
17.5
pods
23:30
23:40
23:50
requested
Cluster CPU Capacity
1
0.500
0
-0.500
-1
cores
No data
23:30
23:40
23:50
Cluster Mem Capacity
1 B
500 mB
0 B
-500 mB
-1 B
No data
23:30
23:40
23:50
Cluster Disk Capacity
1 B
500 mB
0 B
-500 mB
-1 B
No data
23:30
23:40
23:50
Deployments
Deployment Replicas - Up To Date
Time
Metric
Value
Deployment Replicas
Deployment Replicas - Updated
Deployment Replicas - Unavailable


Not secure
a5445ba2134854086985bc5bfc913497-1614150311.ap-southeast-1.elb.amazonaws.com/d/4XuMd2liz/kubernetes-cluster-prometheus?orgId=1
Search or jump to...
ctrl+k
Home › Dashboards › Kubernetes Cluster (Prometheus)
Add
Last 30 minutes
Pods Running
17
Pods Pending
0
Pods Failed
0
Pods Succeeded
2
Pods Unknown
0
Containers
Containers Running
30
Containers Waiting
0
Containers Terminated
2
sum(kube_pod_container_status_terminated{namespace=~".*"})
Containers Restarts (Last 30 Minutes)
N/A
CPU Cores Requested by Containers
N/A
Memory Requested By Containers
N/A
Jobs
Jobs Succeeded
2
Jobs Succeeded
0
Jobs Failed
0

6. Use GitOps for the CD pipeline.

DevOps for Devs:

Install Argo CD on your EKS Cluster

Create an application in Argo CD (you can use GUI or CLI)

Pi-sharp:

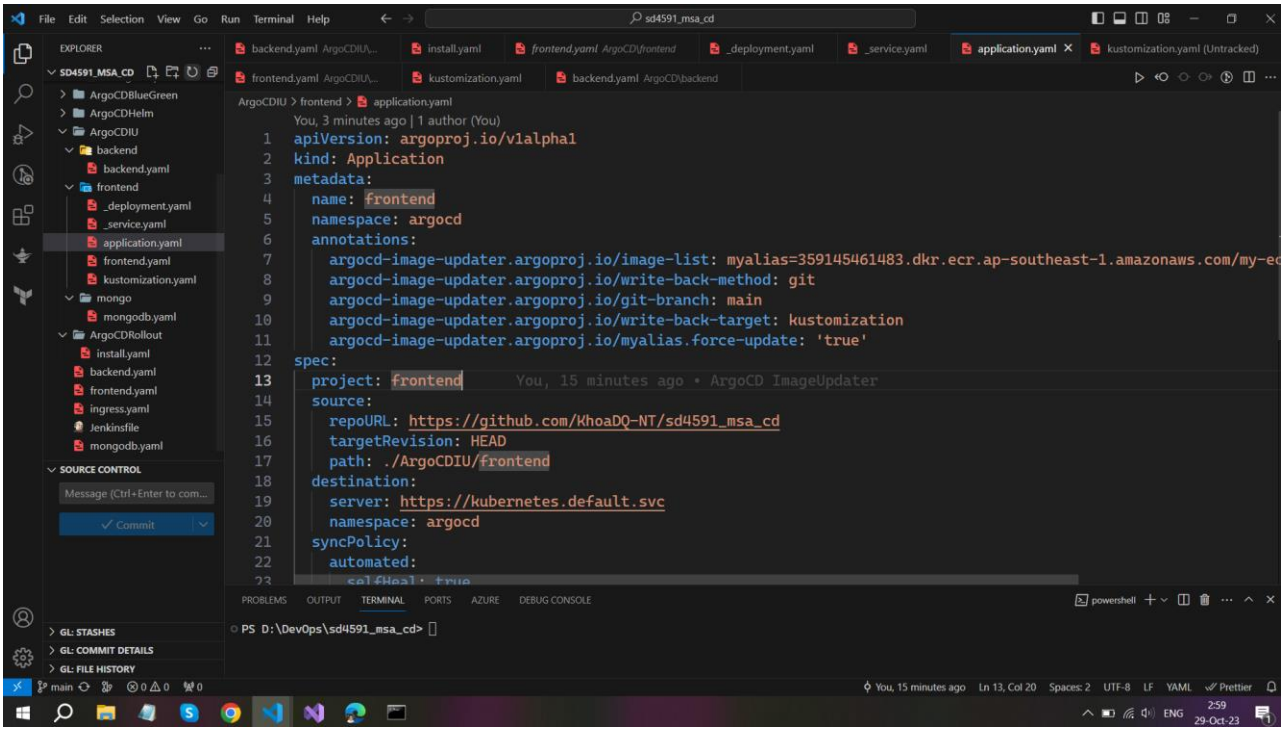
- Install Argo CD Image Update (https://argocd-image-updater.readthedocs.io/en/stable/install/installation/). This will help you listen to changes the image on ECR and update the image tag in Ops repo.
- Use Blue/Green strategy for deployment.

Using ArgoCD Image Updater

Install ArgoCD Image Updater and make some necessary configurations

```
C:\Users\khoadangq>kubectl apply -n argocd -f https://raw.githubusercontent.com/argoproj-labs/argocd-image-updater/stabl
e/manifests/install.yaml
serviceaccount/argocd-image-updater created
role.rbac.authorization.k8s.io/argocd-image-updater created
rolebinding.rbac.authorization.k8s.io/argocd-image-updater created
configmap/argocd-image-updater-config created
configmap/argocd-image-updater-ssh-config created
secret/argocd-image-updater-secret created
deployment.apps/argocd-image-updater created

C:\Users\khoadangq>
```

```
C:\Users\khoadangq>kubectl --namespace argocd logs --selector app.kubernetes.io/name=argocd-image-updater
time="2023-10-28T19:05:23Z" level=info msg="Starting image update cycle, considering 0 annotated application(s) for upda
te"
time="2023-10-28T19:05:23Z" level=info msg="Processing results: applications=0 images_considered=0 images_skipped=0 imag
es_updated=0 errors=0"
time="2023-10-28T19:07:24Z" level=info msg="Starting image update cycle, considering 0 annotated application(s) for upda
te"
time="2023-10-28T19:07:24Z" level=info msg="Processing results: applications=0 images_considered=0 images_skipped=0 imag
es_updated=0 errors=0"
time="2023-10-28T19:09:24Z" level=info msg="Starting image update cycle, considering 0 annotated application(s) for upda
te"
time="2023-10-28T19:09:24Z" level=info msg="Processing results: applications=0 images_considered=0 images_skipped=0 imag
es_updated=0 errors=0"
time="2023-10-28T19:11:24Z" level=info msg="Starting image update cycle, considering 0 annotated application(s) for upda
te"
time="2023-10-28T19:11:24Z" level=info msg="Processing results: applications=0 images_considered=0 images_skipped=0 imag
es_updated=0 errors=0"
time="2023-10-28T19:13:24Z" level=info msg="Starting image update cycle, considering 0 annotated application(s) for upda
te"
time="2023-10-28T19:13:24Z" level=info msg="Processing results: applications=0 images_considered=0 images_skipped=0 imag
es_updated=0 errors=0"

C:\Users\khoadangq>
```



Sort: name Items per page: 10

backend

Project:	argocd
Labels:	
Status:	Healthy Synced
Repository:	https://github.com/KhoaDQ-NT/sd4591_m...
Target Re...	HEAD
Path:	./ArgoCDIU/backend
Destinati...	in-cluster
Namespa...	argocd
Created At:	10/29/2023 15:59:59 (2 hours ago)
Last Sync:	10/29/2023 16:00:00 (2 hours ago)

SYNC REFRESH DELETE

frontend

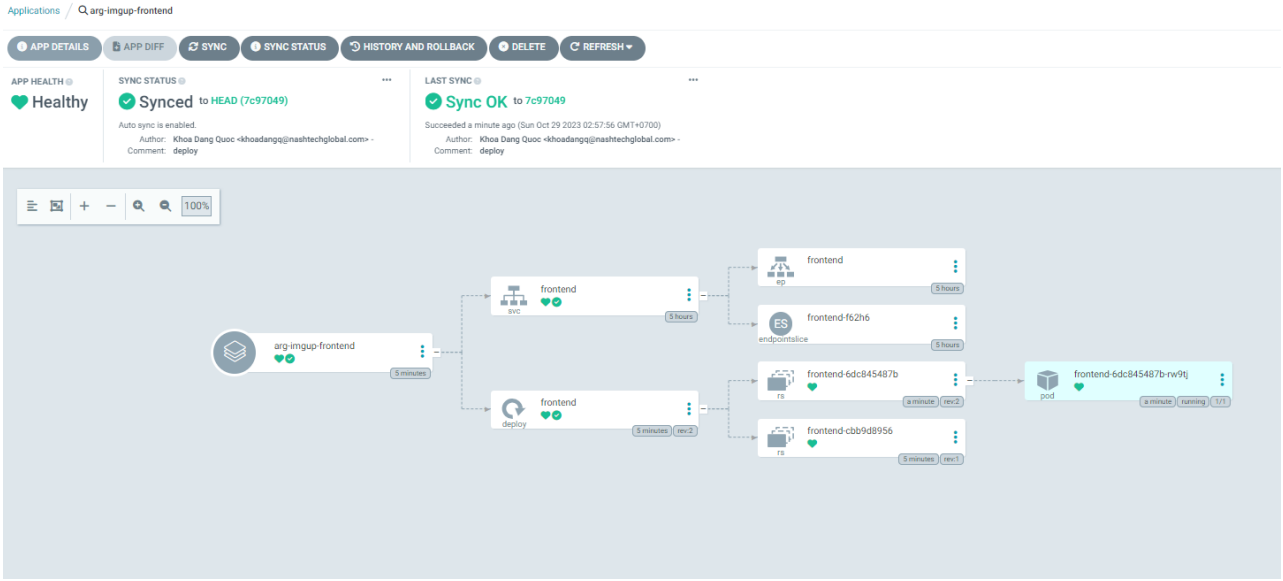
Project:	argocd
Labels:	
Status:	Healthy Synced
Repository:	https://github.com/KhoaDQ-NT/sd4591_m...
Target Re...	HEAD
Path:	./ArgoCDIU/frontend
Destinati...	in-cluster
Namespa...	argocd
Created At:	10/29/2023 16:00:49 (2 hours ago)
Last Sync:	10/29/2023 18:00:25 (3 minutes ago)

SYNC REFRESH DELETE

mongo

Project:	argocd
Labels:	
Status:	Healthy Synced
Repository:	https://github.com/KhoaDQ-NT/sd4591_m...
Target Re...	HEAD
Path:	./ArgoCDIU/mongo
Destinati...	in-cluster
Namespa...	argocd
Created At:	10/29/2023 15:58:37 (2 hours ago)
Last Sync:	10/29/2023 15:58:39 (2 hours ago)

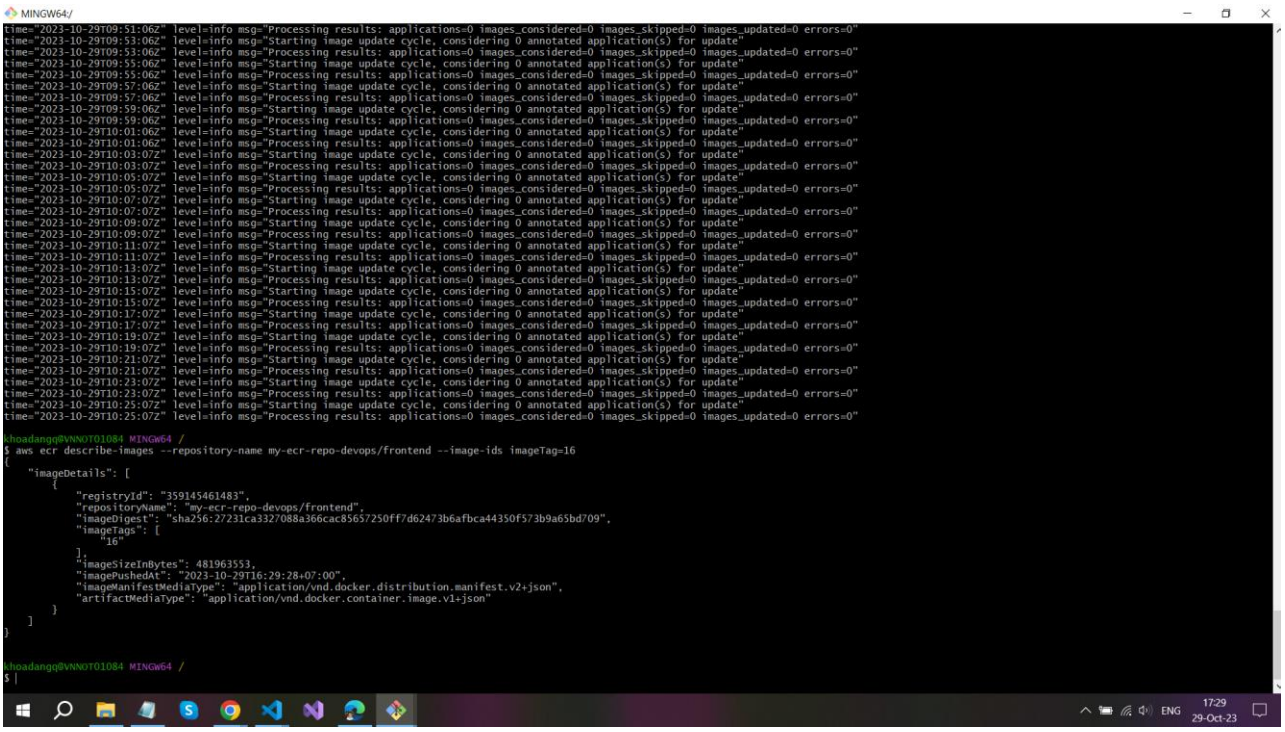
SYNC REFRESH DELETE



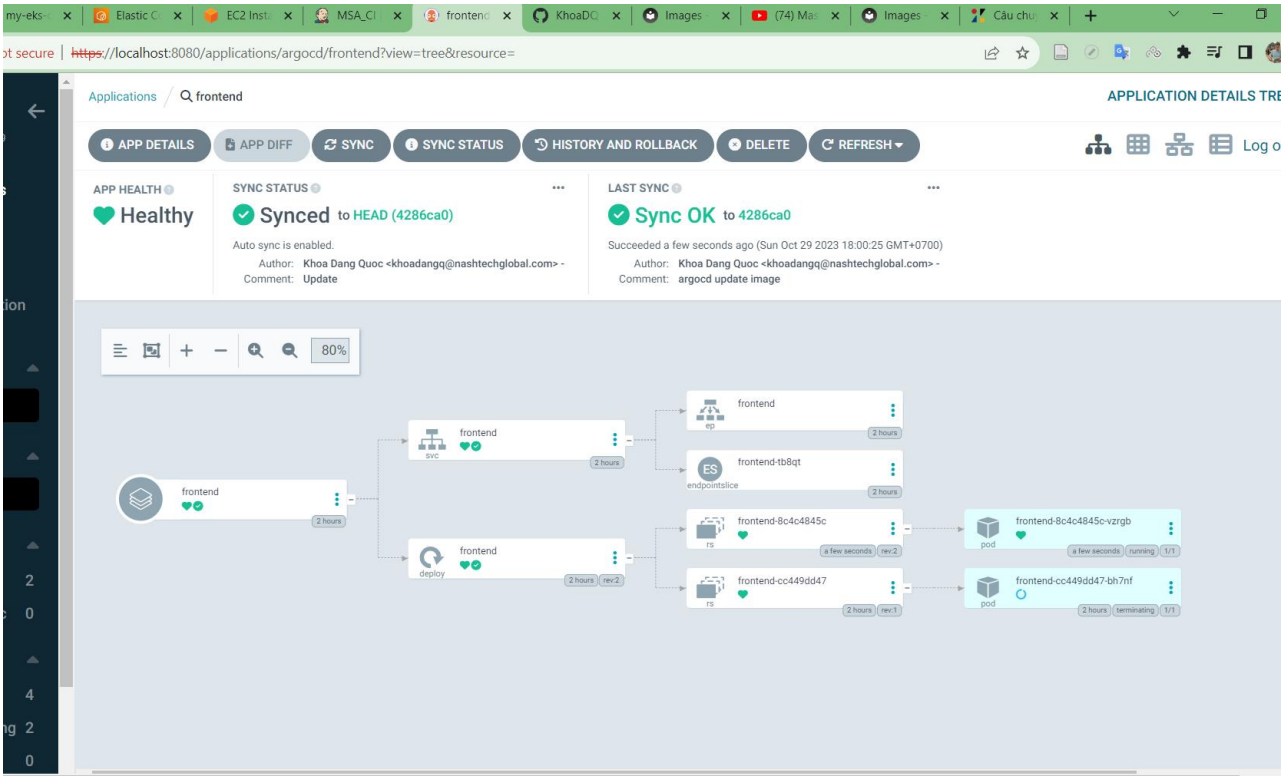
I did try to find any problem, but it still did not work.

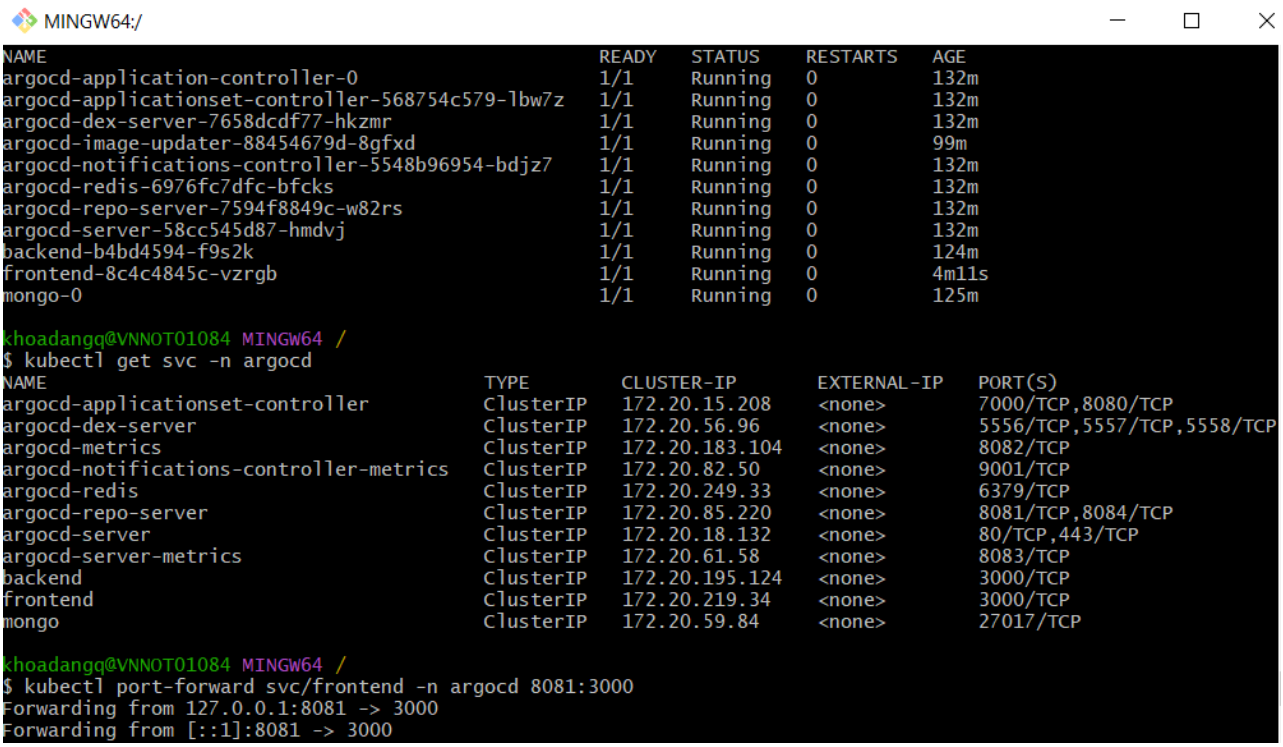
MINGW64:/

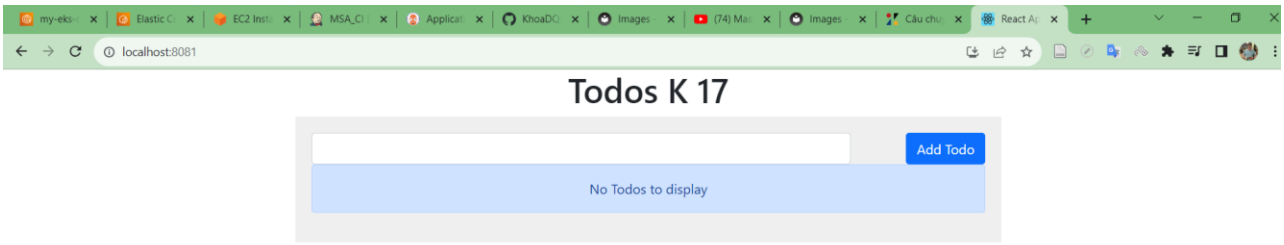
```
time="2023-10-29T10:13:07Z" level=info msg="Starting image update cycle, considering 0 annotated application(s) for update"
time="2023-10-29T10:13:07Z" level=info msg="Processing results: applications=0 images_considered=0 images_skipped=0 images_updated=0 errors=0"

khoadangq@VNNOT01084 MINGW64 /
$ aws ecr describe-images --repository-name my-ecr-repo-devops/frontend --image-ids imageTag=16 --query 'images[0].imageDigest' --output text
None


khoadangq@VNNOT01084 MINGW64 /
$ aws ecr describe-images --repository-name my-ecr-repo-devops/frontend --image-ids imageTag=15 --query 'images[0].imageDigest' --output text
None


khoadangq@VNNOT01084 MINGW64 /
$ aws ecr describe-images --repository-name my-ecr-repo-devops/frontend --image-ids imageTag=17 --query 'images[0].imageDigest' --output text

An error occurred (ImageNotFoundException) when calling the DescribeImages operation: The image with imageId {imageDigest:'null', imageTag:'17'} does not exist w
s/frontend' in the registry with id '359145461483'

khoadangq@VNNOT01084 MINGW64 /
$ aws ecr describe-images --repository-name my-ecr-repo-devops/frontend --output table
-------------------------------------------------------------------------------------------------
|                                         DescribeImages                                        |
+-----------------------------------------------------------------------------------------------+
||                                        imageDetails                                         ||
|+--------------------------+--------------------------------------------------------------------+|
||  artifactMediaType       |  application/vnd.docker.container.image.v1+json                    ||
||  imageDigest             |  sha256:27231ca3327088a366cac85657250ff7d62473b6afbca44350f573b9a65bd709  ||
||  imageManifestMediaType  |  application/vnd.docker.distribution.manifest.v2+json              ||
||  imagePushedAt           |  2023-10-29T16:29:28+07:00                                         ||
||  imageSizeInBytes        |  481963553                                                         ||
||  lastRecordedPullTime    |                                                                    ||
||  registryId              |  359145461483                                                      ||
||  repositoryName          |  my-ecr-repo-devops/frontend                                       ||
|+--------------------------+--------------------------------------------------------------------+|
|||                                        imageTags                                          |||
||+-----------------------------------------------------------------------------------------+||
|||  16                                                                                     |||
||+-----------------------------------------------------------------------------------------+||
||                                        imageDetails                                         ||
|+--------------------------+--------------------------------------------------------------------+|
||  artifactMediaType       |  application/vnd.docker.container.image.v1+json                    ||
||  imageDigest             |  sha256:667985d1aadaa92bd3985ff7a6b7bf183a093d433d314662365c28c9e48e5fd8  ||
||  imageManifestMediaType  |  application/vnd.docker.distribution.manifest.v2+json              ||
||  imagePushedAt           |  2023-10-29T15:54:35+07:00                                         ||
||  imageSizeInBytes        |  481962019                                                         ||
||  lastRecordedPullTime    |  2023-10-29T16:00:51.843000+07:00                                  ||
||  registryId              |  359145461483                                                      ||
||  repositoryName          |  my-ecr-repo-devops/frontend                                       ||
|+--------------------------+--------------------------------------------------------------------+|
|||                                        imageTags                                          |||
||+-----------------------------------------------------------------------------------------+||
|||  15                                                                                     |||
||+-----------------------------------------------------------------------------------------+||
```



ArgoCD Image Updater cannot detect new images, so they do not perform any operations in GitHub Actions. Synchronization still must be done manually. I will take more time to learn later.

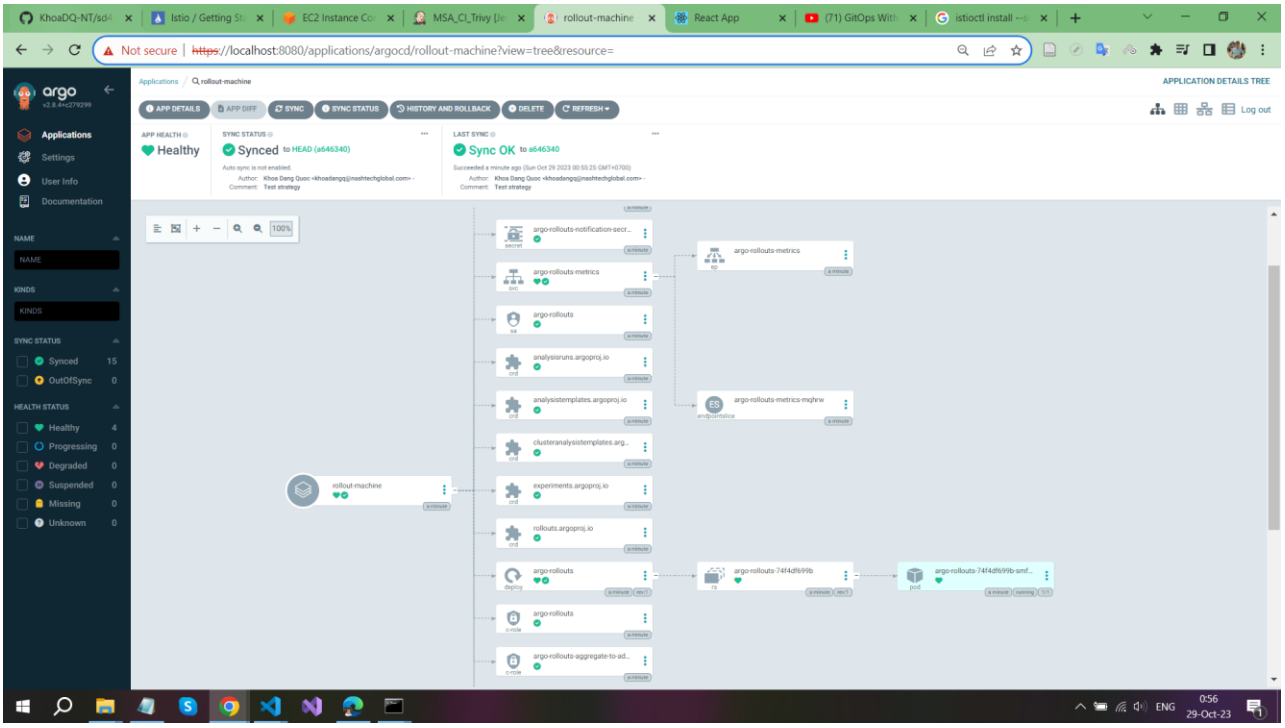




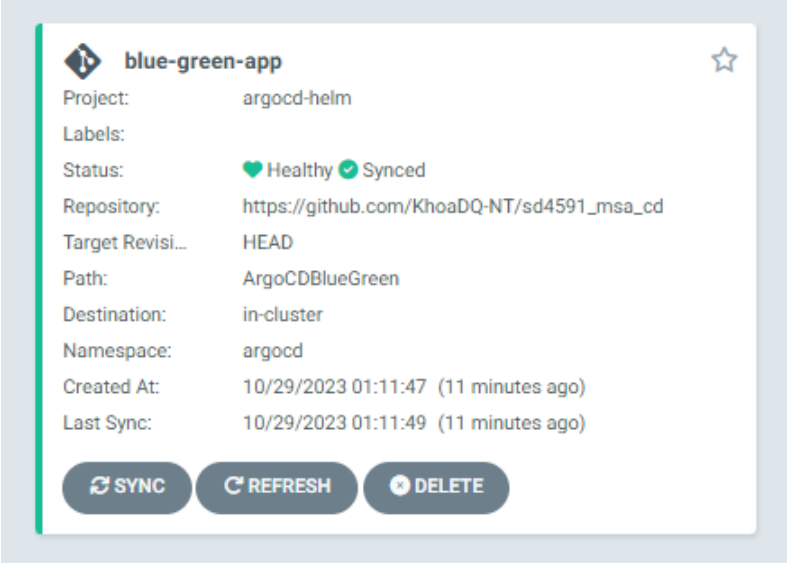


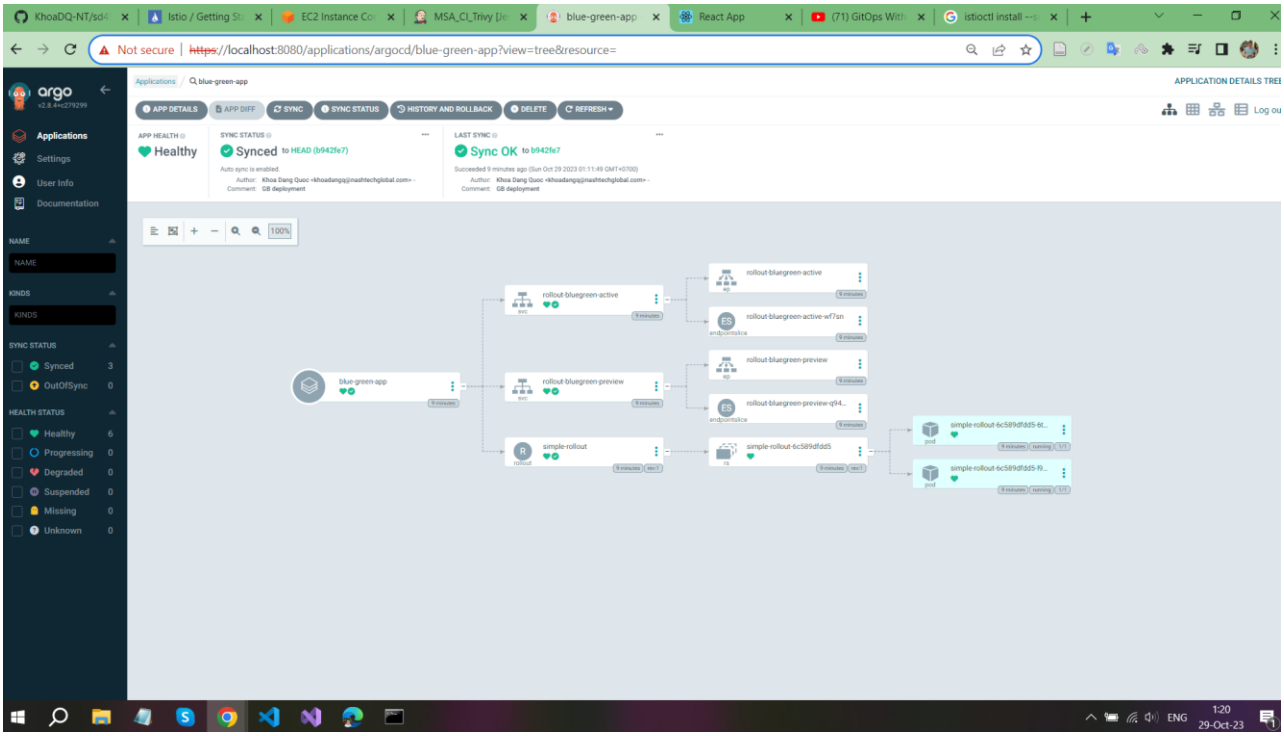
Use Blue/Green strategy

Install rollout-machine to switch between blue and green version of our application

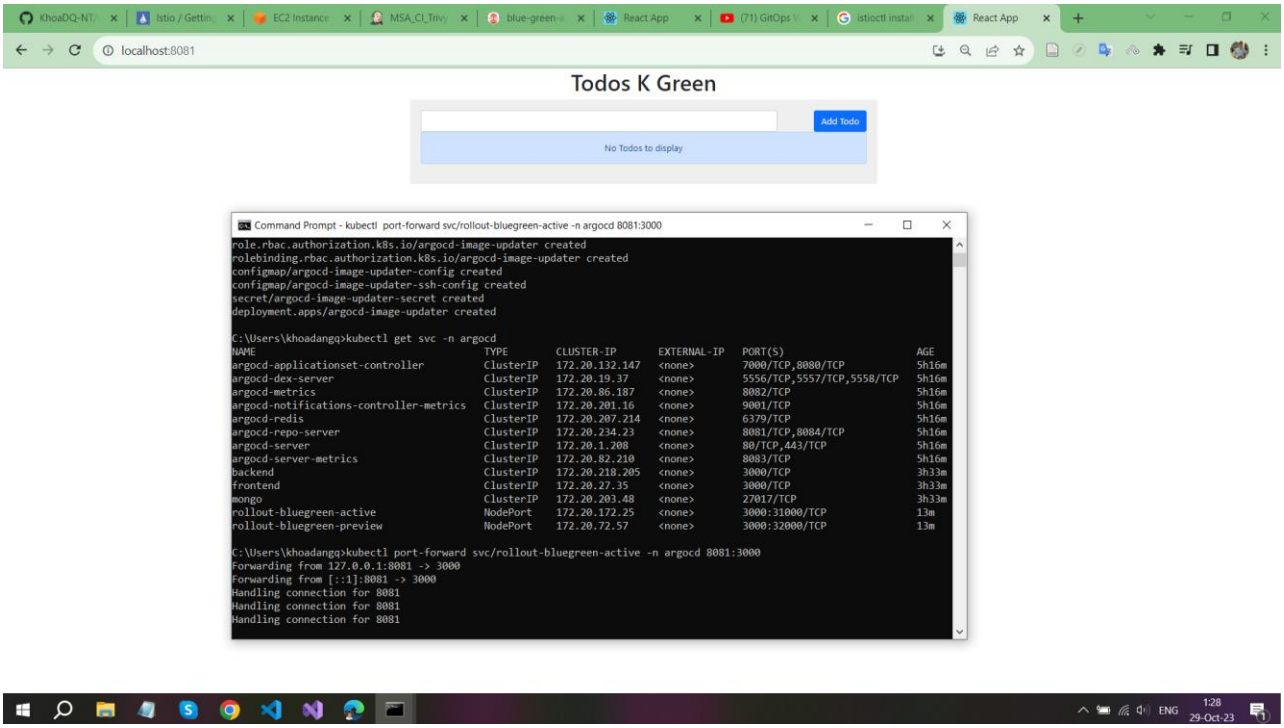


Create blue-green-app that will get deployment based on image version change





Check first deployment



After changing the image, some actions will be processed:

- ArgoCD trigger the changes and have a replica version of application

- Argo Rollouts creates another replica with the new version

- The existing version still works and gets live traffic

- ArgoCD will mark our application as out-of-sync

- Until the appplication blue-green-app is synchronized and the new version is completely initialized. Our web application will be switched to new version for using

After the changes were applied, forward port if it's necessary and continue to use the application