Practical DevOps for Devs – Assignment

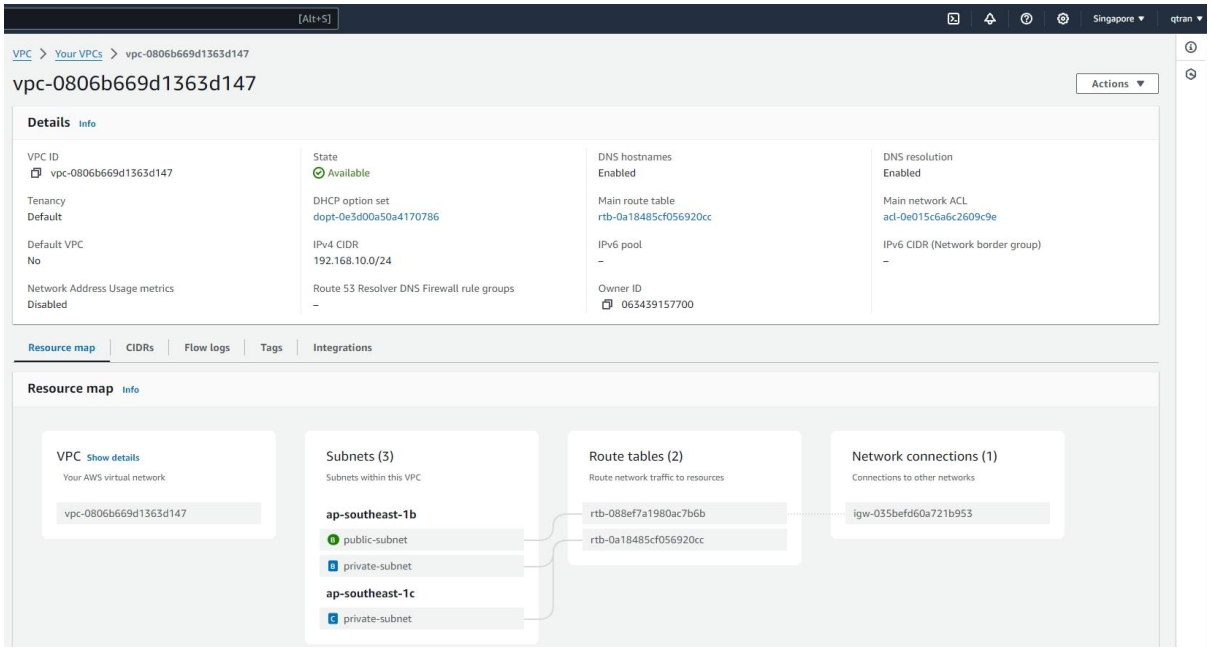
Source Code Management

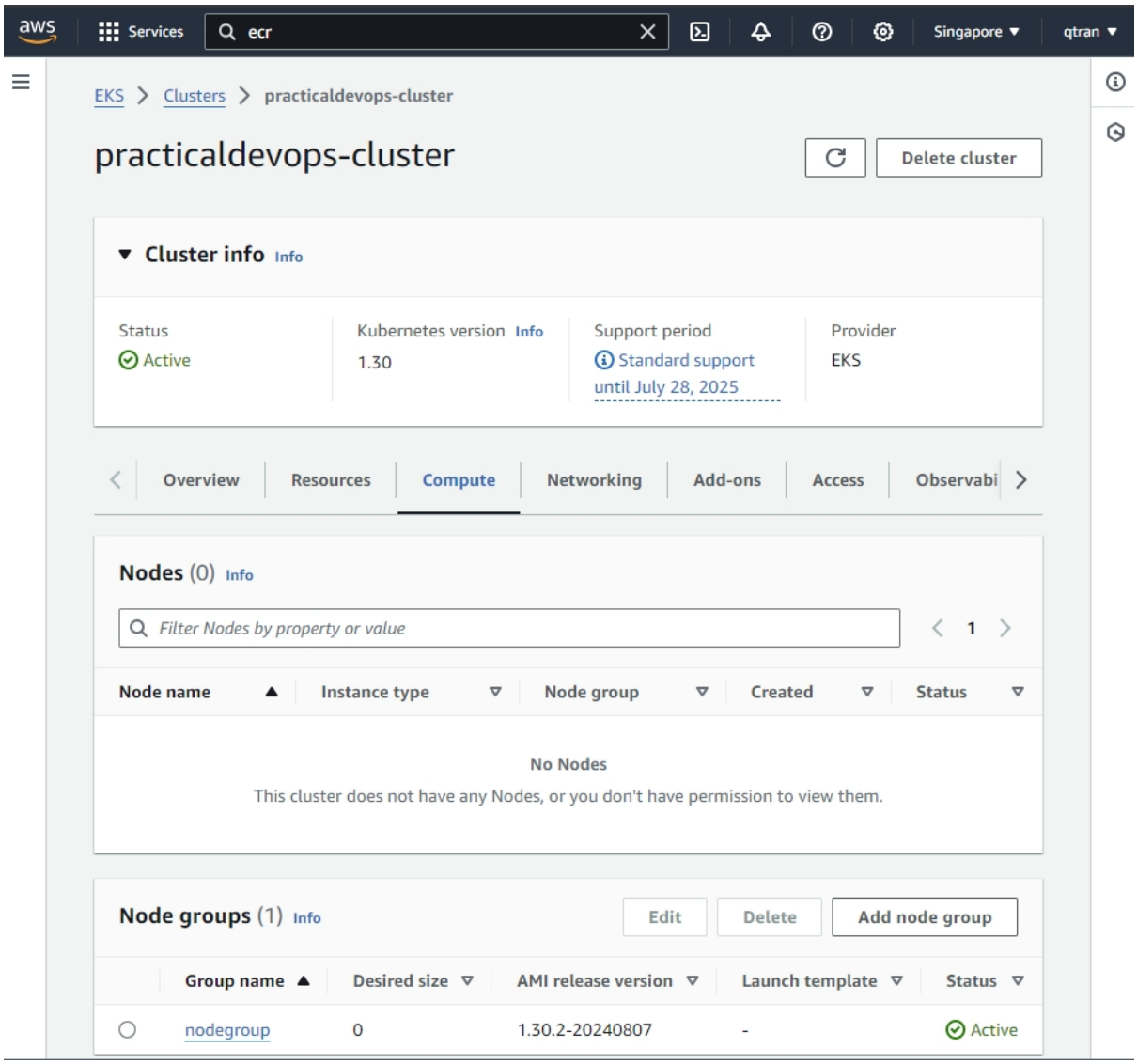
- Repository for the application: XQuyTran/sd5129_msa (github.com)
- Repository for the Infrastructure: XQuyTran/sd5129_infrastructure (github.com)

1. Setting up a CI/CD Pipeline and deploying applications on AWS EKS

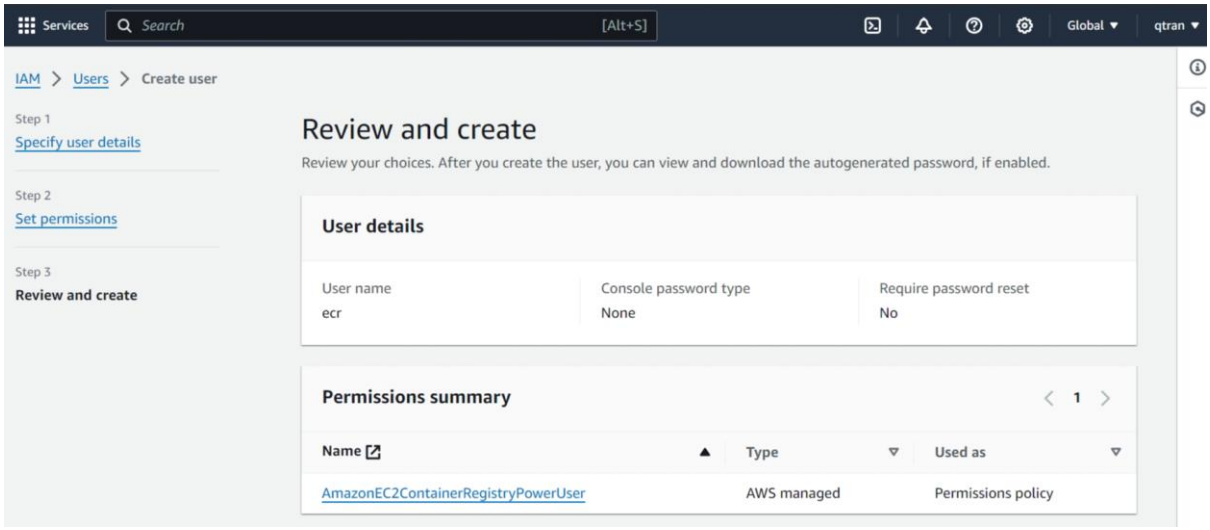
Provision AWS resources

```
(base) quytran@VNNOT01796:/mnt/d/training/Pratical DevOps/infra/aws$ terraform plan
data.aws_iam_policy.eks_worker_policy[2]: Reading...
data.aws_iam_policy.eks_worker_policy[0]: Reading...
data.aws_iam_policy.eks_worker_policy[1]: Reading...
data.aws_iam_policy_document.ec2_assume_role: Reading...
data.aws_iam_policy.eks_cluster_policy: Reading...
data.aws_iam_policy_document.eks_assume_role: Reading...
data.aws_iam_policy_document.eks_assume_role: Read complete after 0s [id=764451762]
data.aws_iam_policy_document.ec2_assume_role: Read complete after 0s [id=1470195627]
data.aws_iam_policy.eks_worker_policy[0]: Read complete after 5s [id=arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy]
data.aws_iam_policy.eks_cluster_policy: Read complete after 5s [id=arn:aws:iam::aws:policy/AmazonEKSClusterPolicy]
data.aws_iam_policy.eks_worker_policy[1]: Read complete after 5s [id=arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy]
data.aws_iam_policy.eks_worker_policy[2]: Read complete after 5s [id=arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_ecr_repository.backend will be created
  + resource "aws_ecr_repository" "backend" {
      + arn                  = (known after apply)
      + id                   = (known after apply)
      + image_tag_mutability = "MUTABLE"
      + name                 = "ntg-garage-backend"
      + registry_id          = (known after apply)
      + repository_url       = (known after apply)
      + tags_all             = {
          + "Org"     = "NT"
          + "Project" = "Pratical DevOps"
        }
    }

  # aws_ecr_repository.frontend will be created
  + resource "aws_ecr_repository" "frontend" {
      + arn                  = (known after apply)
      + id                   = (known after apply)
      + image_tag_mutability = "MUTABLE"
      + name                 = "ntg-garage-frontend"
      + registry_id          = (known after apply)
      + repository_url       = (known after apply)
      + tags_all             = {
          + "Org"     = "NT"
          + "Project" = "Pratical DevOps"
        }
    }

  # aws_eks_addon.this[0] will be created
  + resource "aws_eks_addon" "this" {
      + addon_name           = "kube-proxy"
      + addon_version        = (known after apply)
      + arn                  = (known after apply)
      + cluster_name         = "praticaldevops"
      + configuration_values = (known after apply)
      + created_at           = (known after apply)
```


[Alt+S]
Singapore
qtran
VPC > Your VPCs > vpc-0806b669d1363d147
vpc-0806b669d1363d147
Actions
Details Info
VPC ID
vpc-0806b669d1363d147
State
Available
DNS hostnames
Enabled
DNS resolution
Enabled
Tenancy
Default
DHCP option set
dopt-0e3d00a50a4170786
Main route table
rtb-0a18485cf056920cc
Main network ACL
acl-0e015c6a6c2609c9e
Default VPC
No
IPv4 CIDR
192.168.10.0/24
IPv6 pool
–
IPv6 CIDR (Network border group)
–
Network Address Usage metrics
Disabled
Route 53 Resolver DNS Firewall rule groups
–
Owner ID
063439157700
Resource map
CIDRs
Flow logs
Tags
Integrations
Resource map Info
VPC Show details
Your AWS virtual network
vpc-0806b669d1363d147
Subnets (3)
Subnets within this VPC
ap-southeast-1b
public-subnet
private-subnet
ap-southeast-1c
private-subnet
Route tables (2)
Route network traffic to resources
rtb-088ef7a1980ac7b6b
rtb-0a18485cf056920cc
Network connections (1)
Connections to other networks
igw-035befd60a721b953

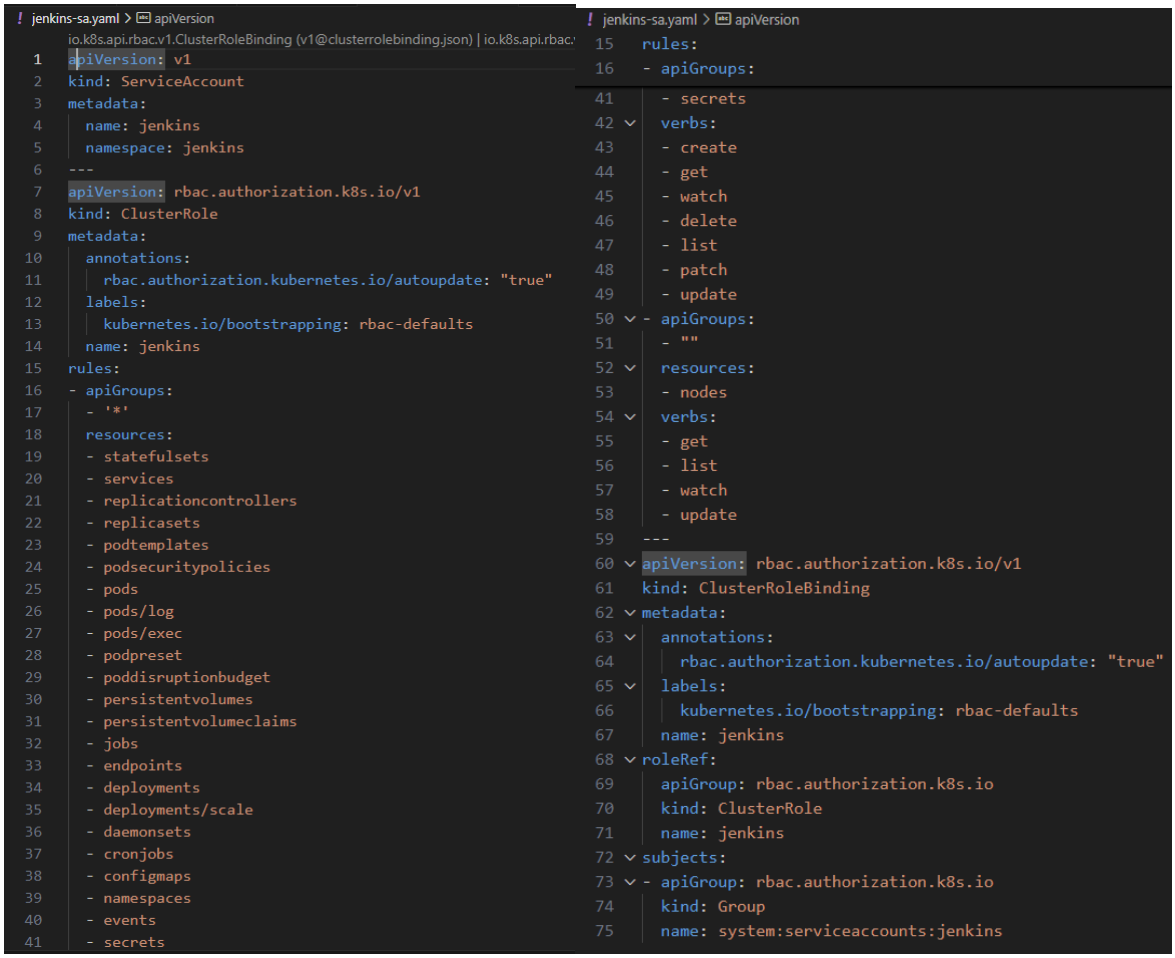

aws
Services
ecr
Singapore
qtran
EKS > Clusters > practicaldevops-cluster
practicaldevops-cluster
Delete cluster
Cluster info Info
Status
Active
Kubernetes version Info
1.30
Support period
Standard support
until July 28, 2025
Provider
EKS
Overview
Resources
Compute
Networking
Add-ons
Access
Observabi
Nodes (0) Info
Filter Nodes by property or value
1
Node name
Instance type
Node group
Created
Status
No Nodes
This cluster does not have any Nodes, or you don't have permission to view them.
Node groups (1) Info
Edit
Delete
Add node group
Group name
Desired size
AMI release version
Launch template
Status
nodegroup
0
1.30.2-20240807
-
Active

Create IAM user for Jenkins pipeline to push image



Jenkins Installation

Create namespace and service account for Jenkins



jenkins-sa.yaml > apiVersion

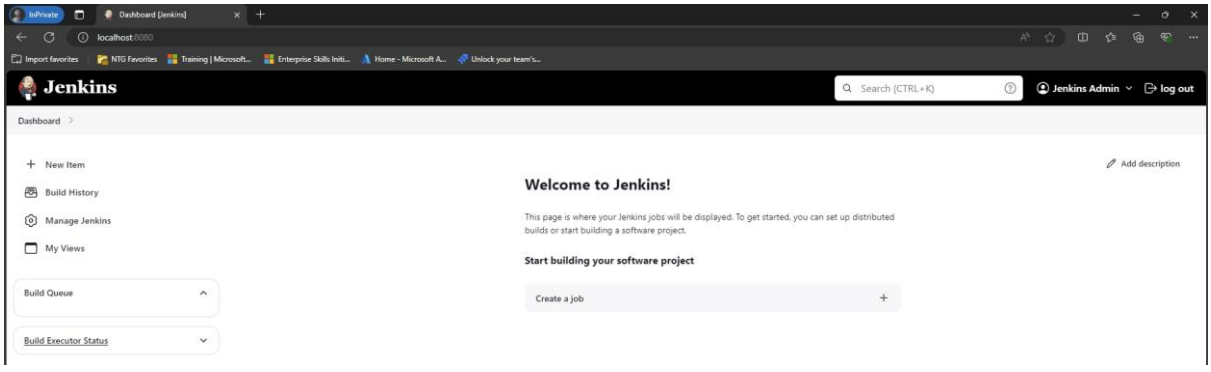
```yaml
apiVersion: v1
kind: ServiceAccount
metadata:
  name: jenkins
  namespace: jenkins
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  annotations:
    rbac.authorization.kubernetes.io/autoupdate: "true"
  labels:
    kubernetes.io/bootstrapping: rbac-defaults
  name: jenkins
rules:
- apiGroups:
  - '*'
  resources:
  - statefulsets
  - services
  - replicationcontrollers
  - replicasets
  - podtemplates
  - podsecuritypolicies
  - pods
  - pods/log
  - pods/exec
  - podpreset
  - poddisruptionbudget
  - persistentvolumes
  - persistentvolumeclaims
  - jobs
  - endpoints
  - deployments
  - deployments/scale
  - daemonsets
  - cronjobs
  - configmaps
  - namespaces
  - events
  - secrets
  verbs:
  - create
  - get
  - watch
  - delete
  - list
  - patch
  - update
- apiGroups:
  - ""
  resources:
  - nodes
  verbs:
  - get
  - list
  - watch
  - update
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  annotations:
    rbac.authorization.kubernetes.io/autoupdate: "true"
  labels:
    kubernetes.io/bootstrapping: rbac-defaults
  name: jenkins
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: jenkins
subjects:
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: system:serviceaccounts:jenkins
```

Install Jenkins Helm chart with custom values to use the pre-defined service account

D: > training > Pratical DevOps > infra > jenkins > ! jenkins-values.yml > {} controller > {} ingress > [ ] paths > {} 0 > {} bac

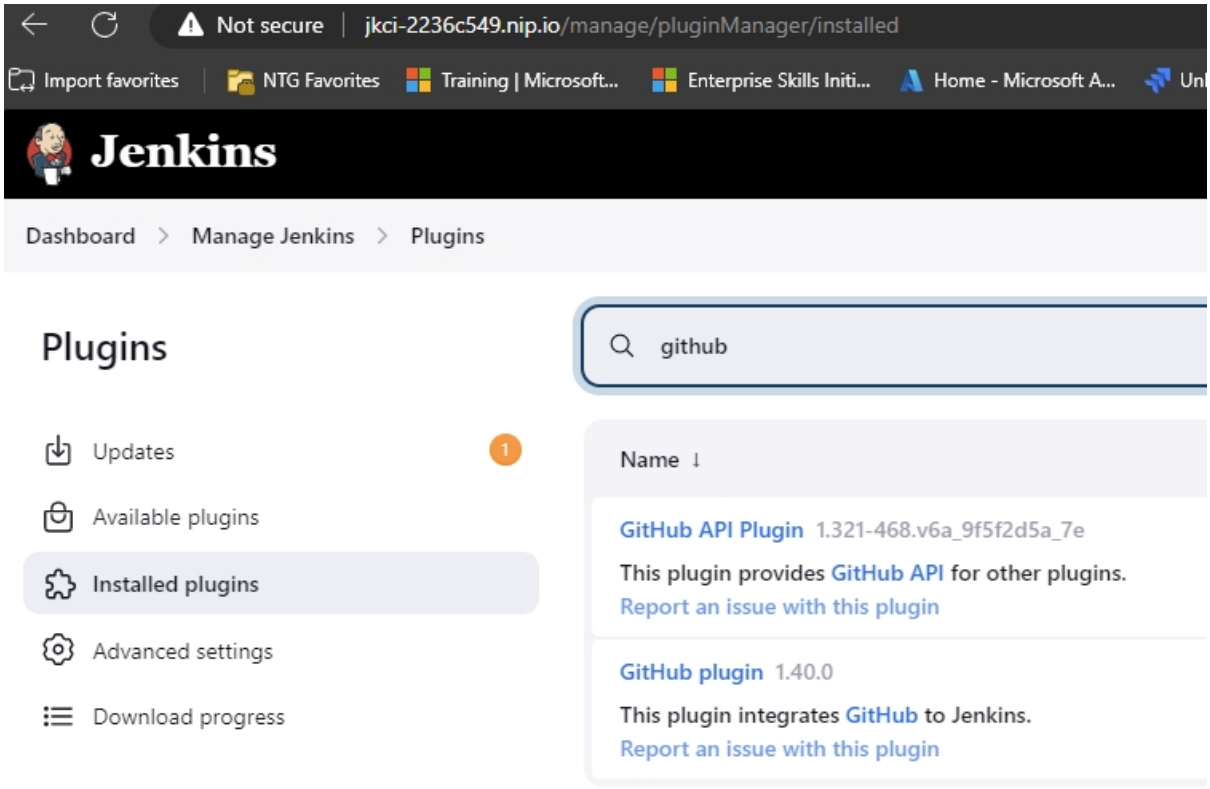
```yaml
controller:
  resources:
    limits:
      cpu: '1'
      memory: 1Gi
  initContainerResources:
    limits:
      cpu: "1"
      memory: "1Gi"
  ingress:
    enabled: true
    hostName: jkci-223104de.nip.io
    paths:
    - backend:
        service:
          name: jenkins
          port:
            number: 8080
      pathType: Prefix
      path: /
    annotations:
      nginx.ingress.kubernetes.io/enable-cors: "true"
      nginx.ingress.kubernetes.io/cors-allow-origin: "https://github.com"
      nginx.ingress.kubernetes.io/whitelist-source-range: "115.79.196.39/32"
persistence:
  storageClass: standard
agent:
  enabled: false
serviceAccount:
  create: false

```

```
(base) quytran@WNNOT01796:/mnt/d/training/Pratical DevOps/infra$ helm install -n jenkins -f jenkins-values.yml jenkins jenkinsci/jenkins
coalesce.go:289: warning: destination for jenkins.controller.InstallPlugins is a table. Ignoring non-table value ([kubernetes:4253.v7700d91739e5 workflow-aggregator:600.vb_57cdd26fdd7 git:5.2.2 configuration-as-code:1836.vccda_
4a_122a_a_e])
NAME: jenkins
LAST DEPLOYED: Wed Jul 31 16:55:10 2024
NAMESPACE: jenkins
STATUS: deployed
REVISION: 1
NOTES:
1. Get your 'admin' user password by running:
  kubectl exec --namespace jenkins -it svc/jenkins -c jenkins -- /bin/cat /run/secrets/additional/chart-admin-password && echo
2. Get the Jenkins URL to visit by running these commands in the same shell:
  echo http://127.0.0.1:8080
  kubectl --namespace jenkins port-forward svc/jenkins 8080:8080

3. Login with the password from step 1 and the username: admin
4. Configure security realm and authorization strategy
5. Use Jenkins Configuration as Code by specifying configScripts in your values.yaml file, see documentation: http://127.0.0.1:8080/configuration-as-code and examples: https://github.com/jenkinsci/configuration-as-code-plugin/t
ree/master/demos

For more information on running Jenkins on Kubernetes, visit:
https://cloud.google.com/solutions/jenkins-on-container-engine

For more information about Jenkins Configuration as Code, visit:
https://jenkins.io/projects/jcasc/


NOTE: Consider using a custom image with pre-installed plugins
(base) quytran@WNNOT01796:/mnt/d/training/Pratical DevOps/infra$ kubectl --namespace jenkins port-forward svc/jenkins 8080:8080
Forwarding from 127.0.0.1:8080 -> 8080
Forwarding from [::1]:8080 -> 8080
```

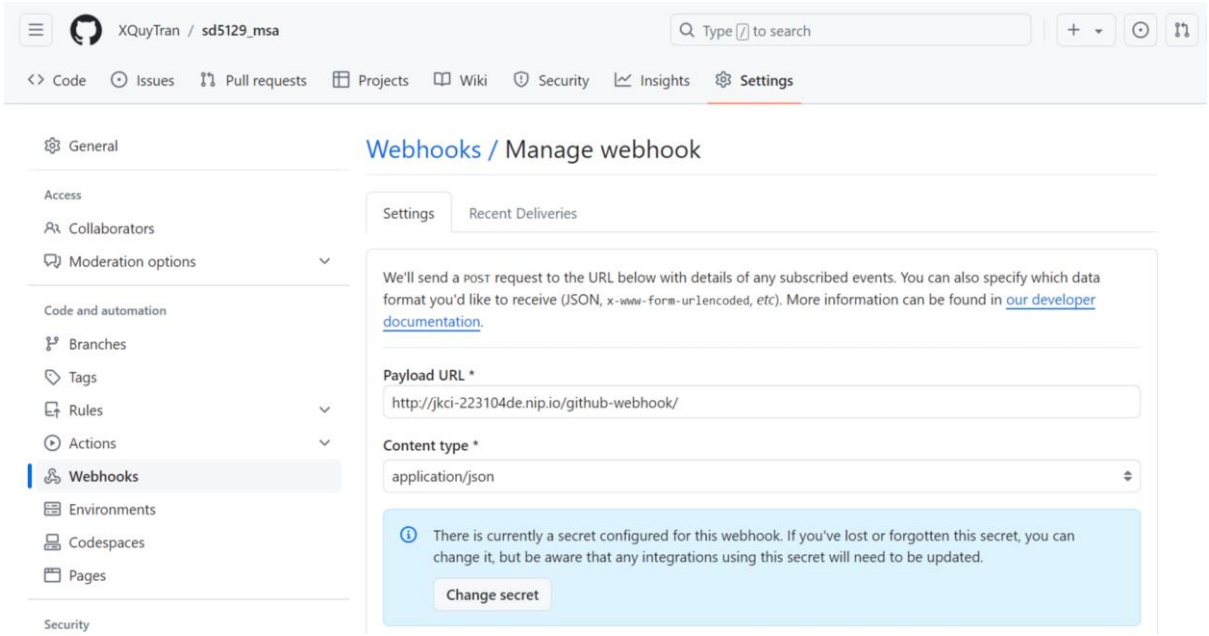


Setup Jenkins pipeline for CI/CD

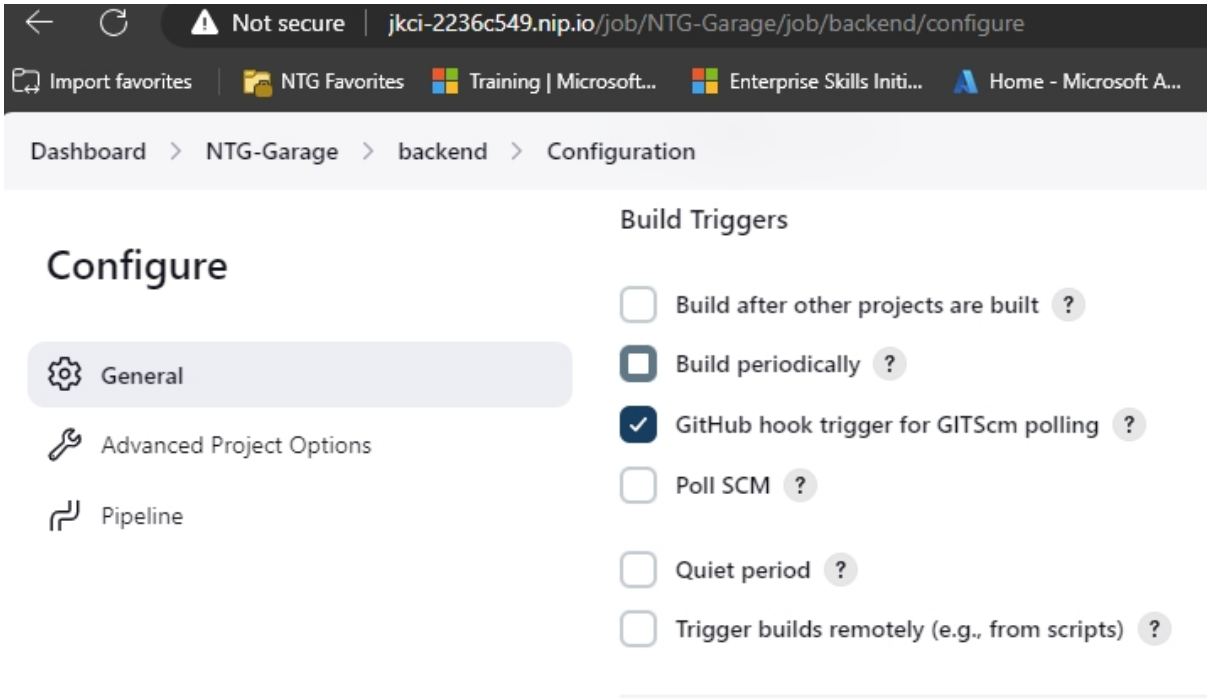
Install GitHub plugin to support receive webhooks

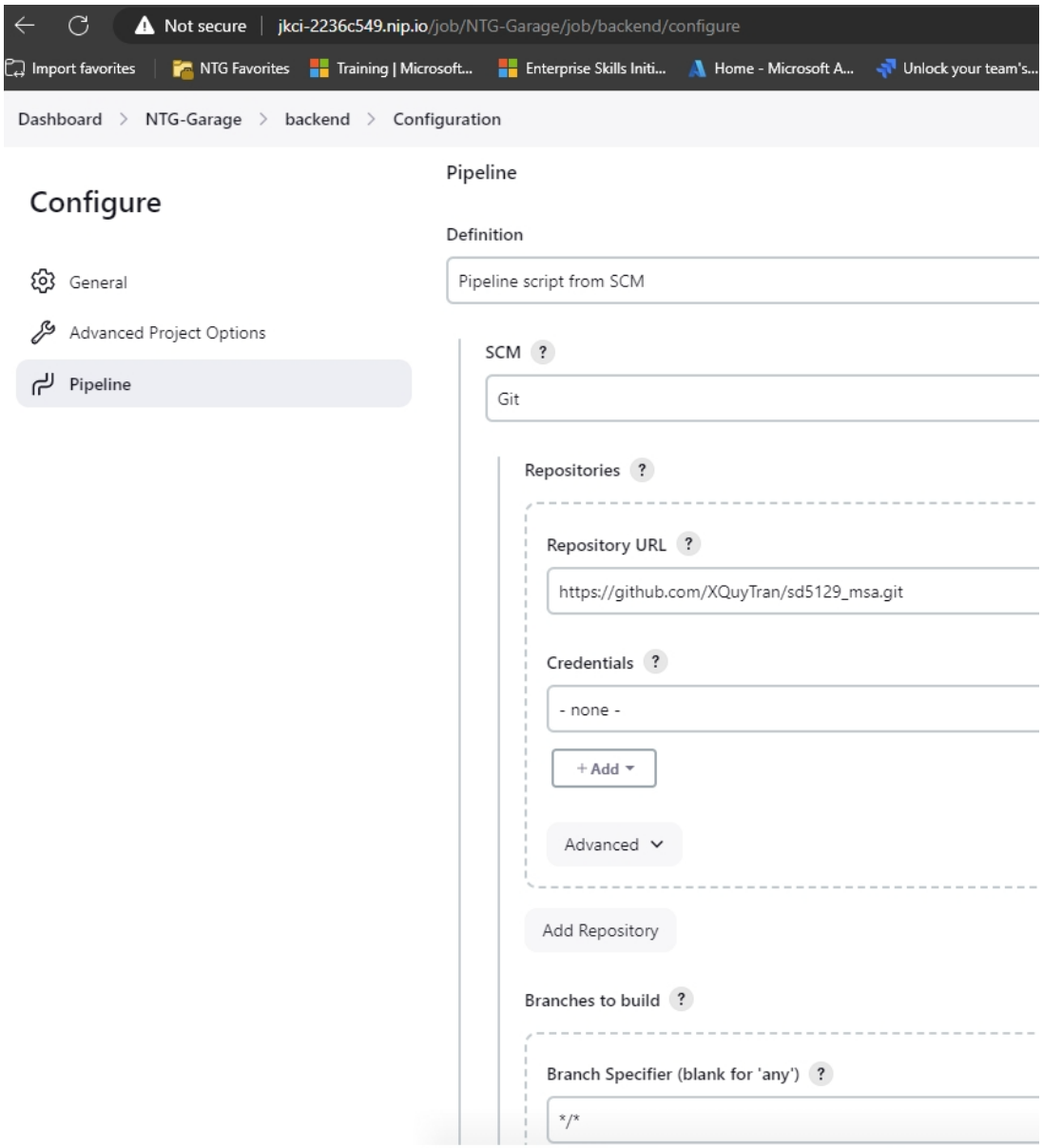


Configure webhook on GitHub to trigger pipeline on push

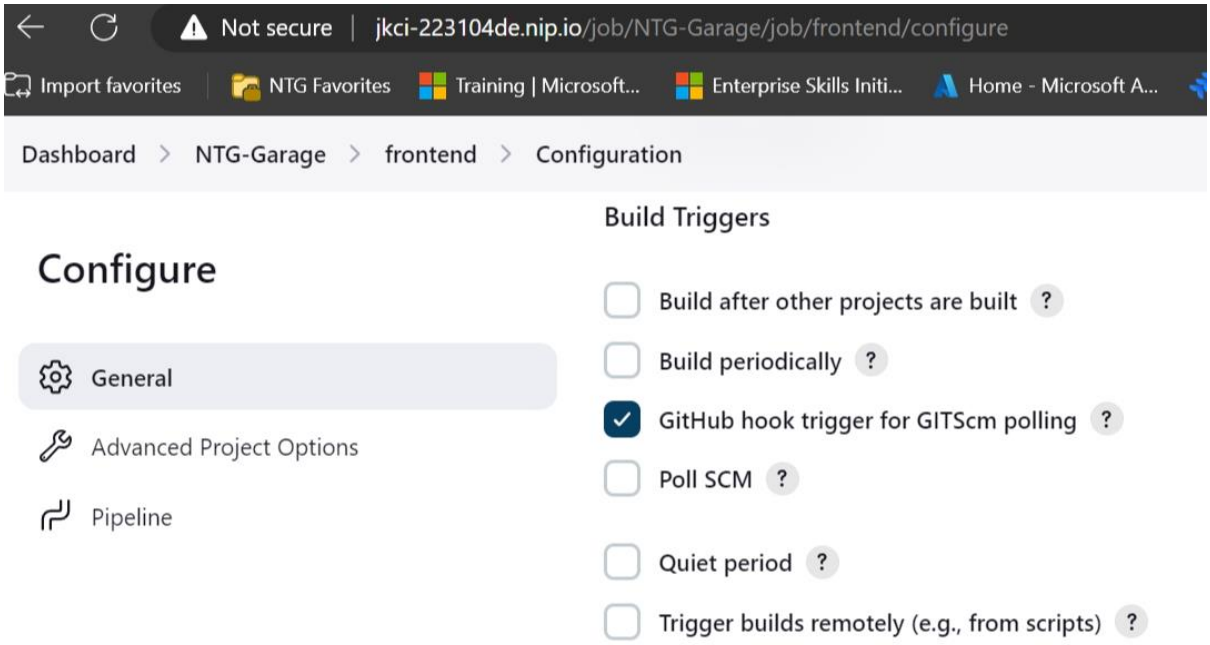


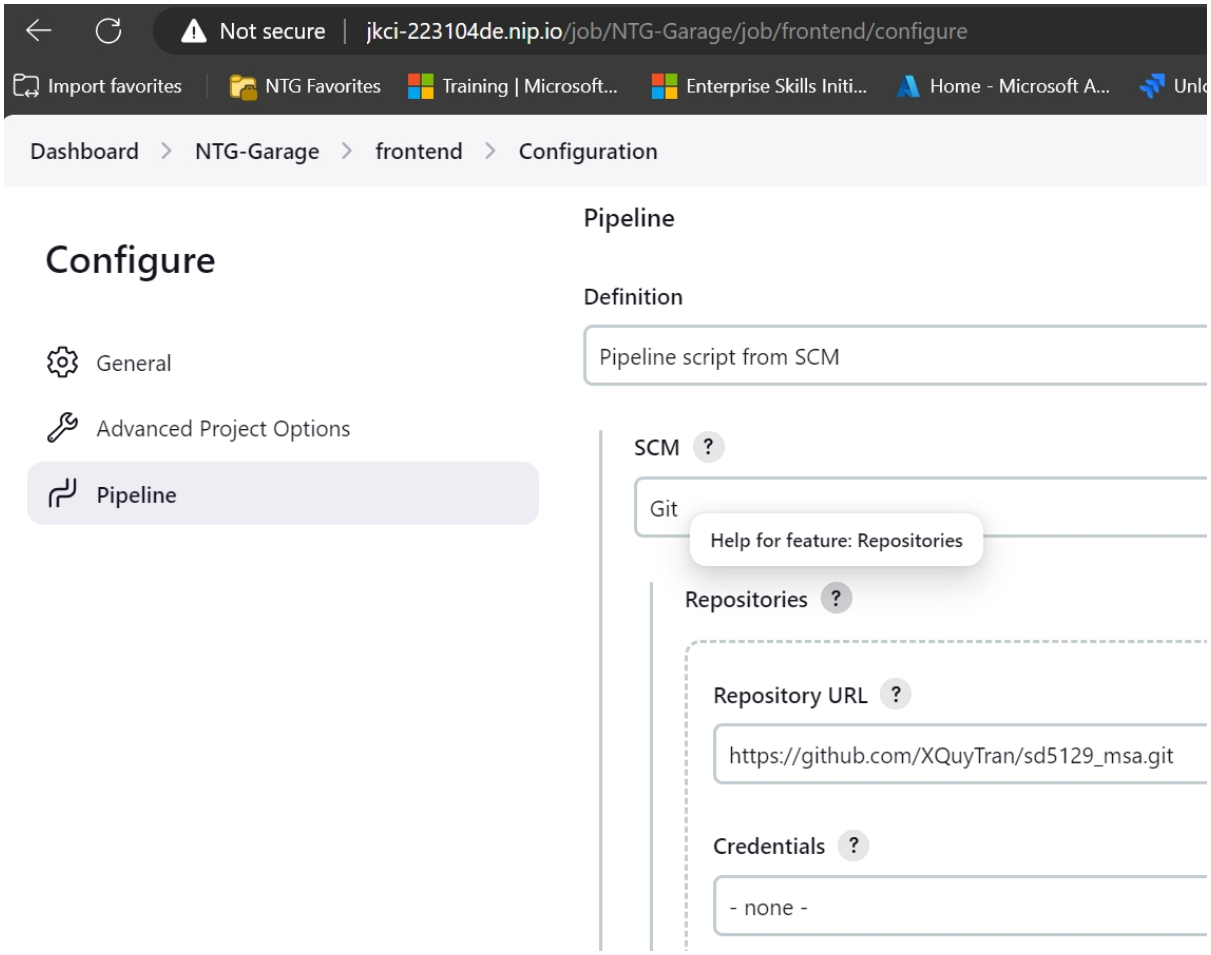
Create and configure pipeline file location for backend service

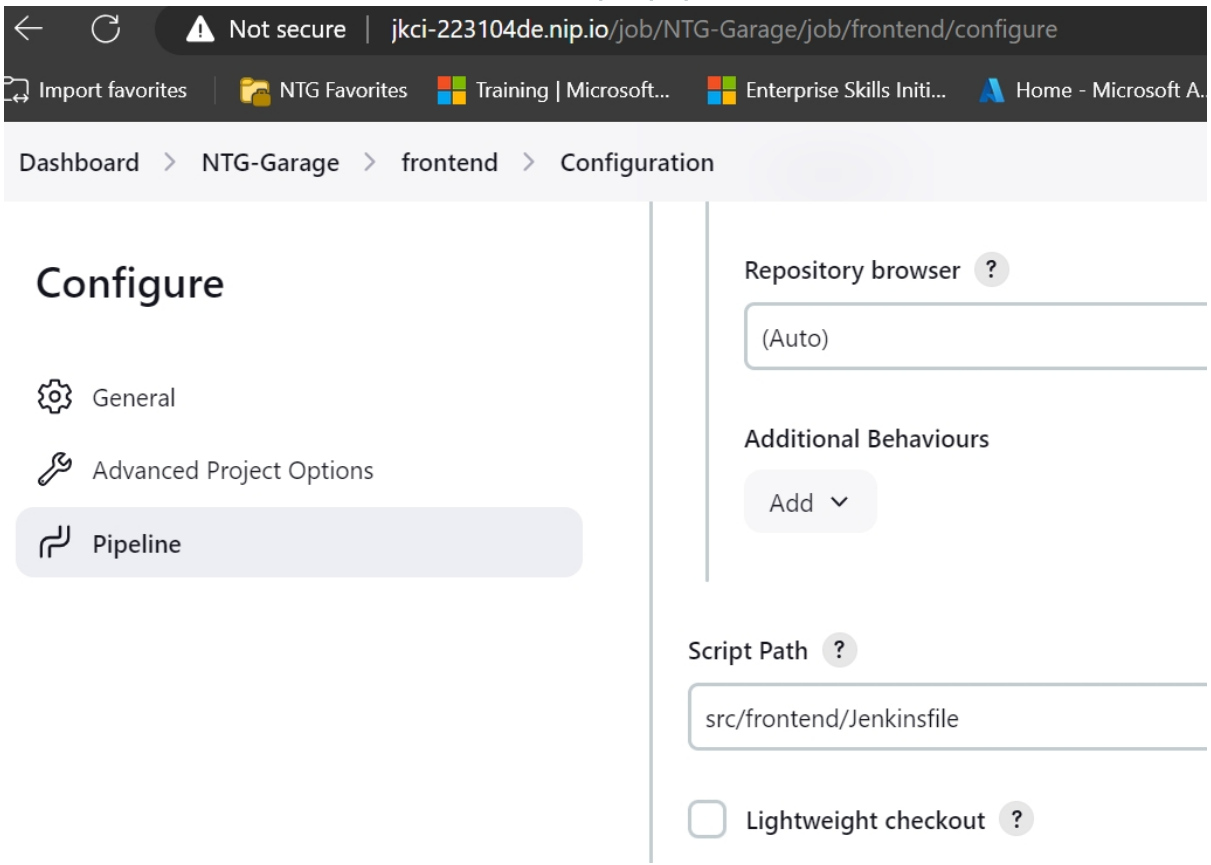




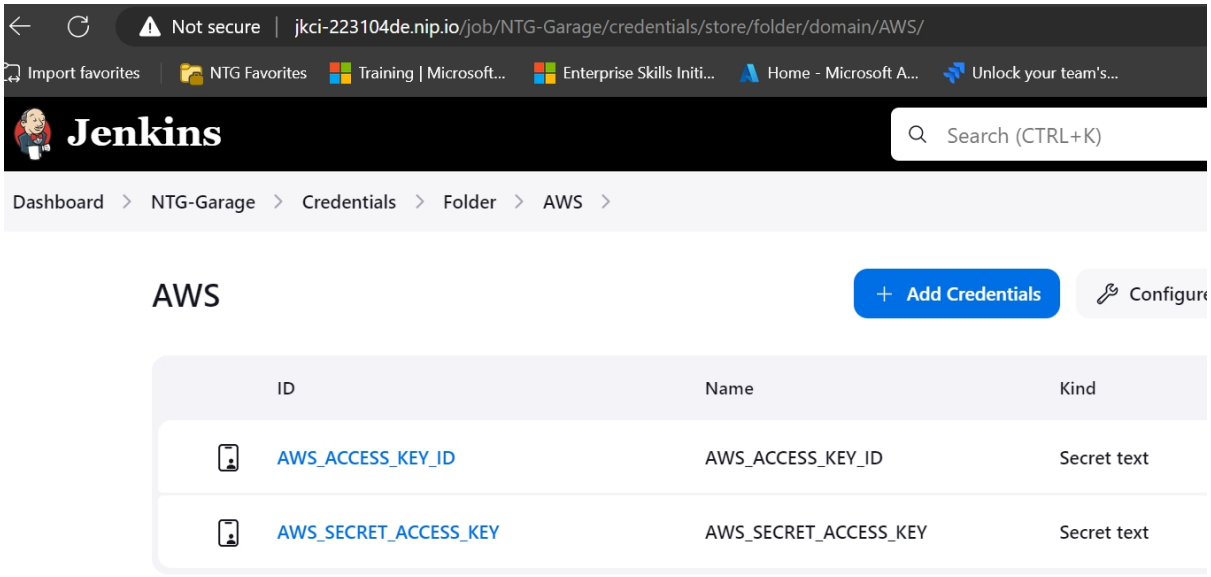
Create and configure pipeline file location for frontend service



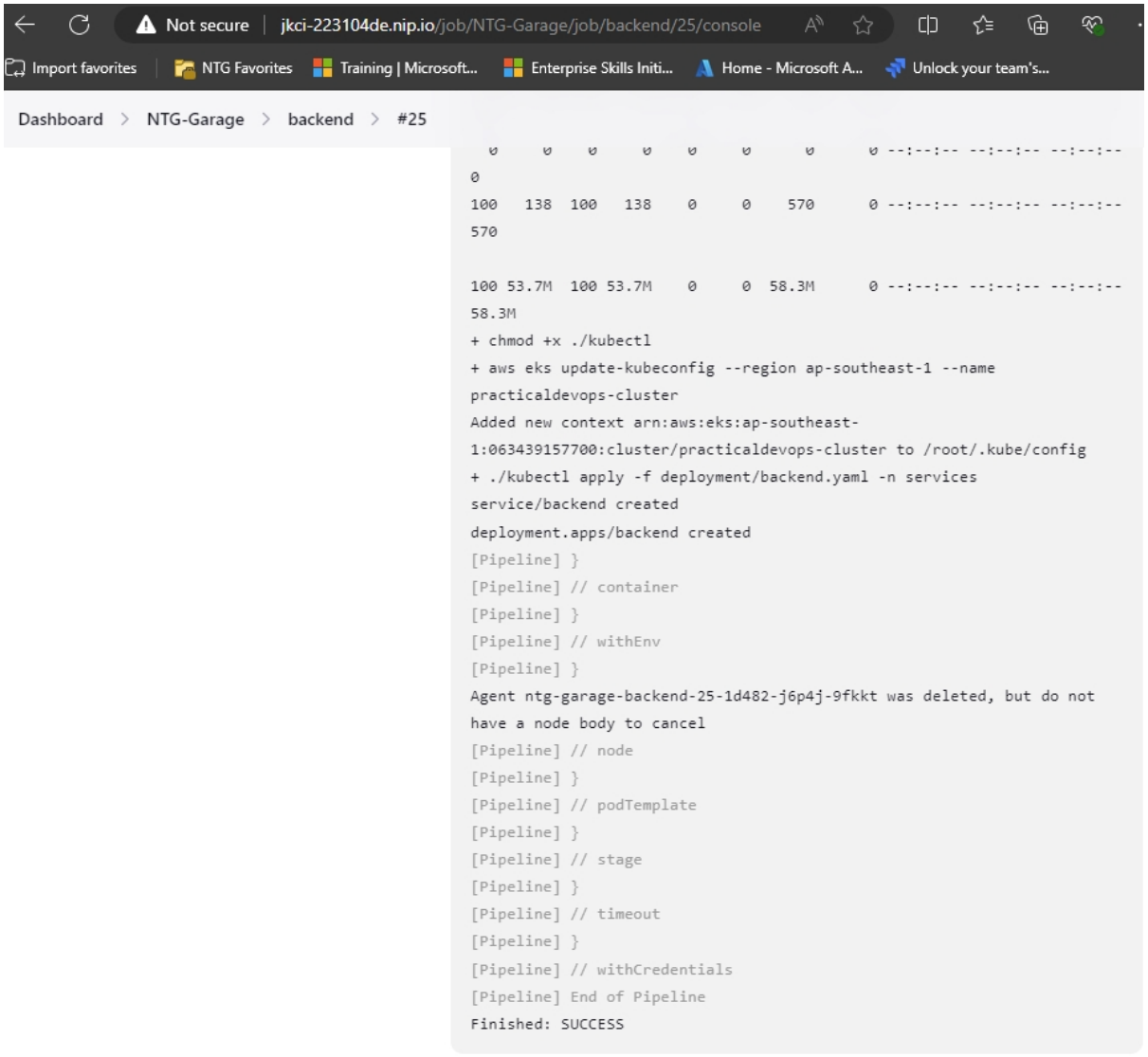

Not secure | jkci-223104de.nip.io/job/NTG-Garage/job/frontend/configure
Import favorites
NTG Favorites
Training | Microsoft...
Enterprise Skills Initi...
Home - Microsoft A...
Dashboard > NTG-Garage > frontend > Configuration
Configure
General
Advanced Project Options
Pipeline
Pipeline
Definition
Pipeline script from SCM
SCM
Git
Help for feature: Repositories
Repositories
Repository URL
https://github.com/XQuyTran/sd5129_msa.git
Credentials
- none -

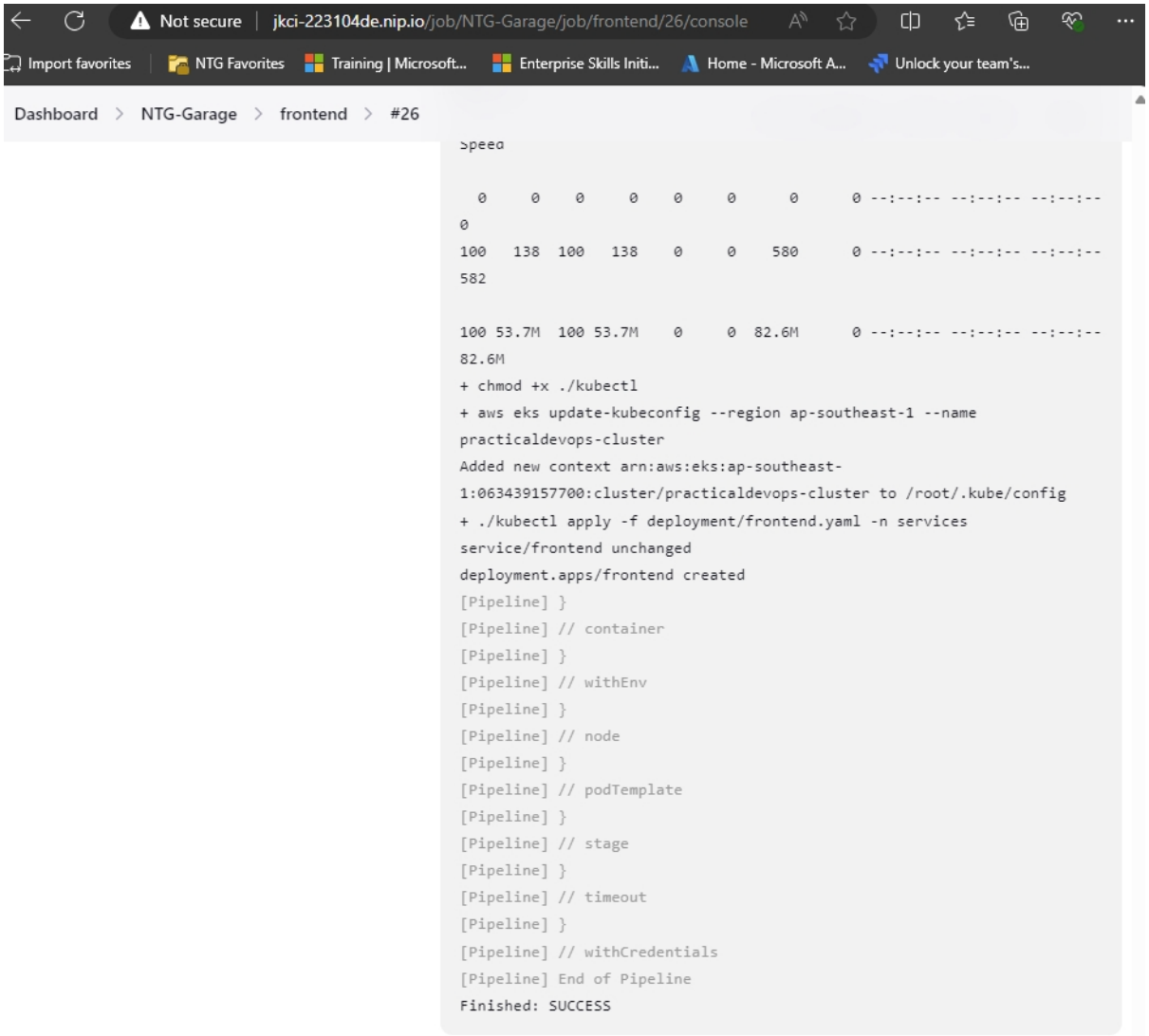

Not secure | jkci-223104de.nip.io/job/NTG-Garage/job/frontend/configure
Import favorites
NTG Favorites
Training | Microsoft...
Enterprise Skills Initi...
Home - Microsoft A...
Dashboard > NTG-Garage > frontend > Configuration
Configure
General
Advanced Project Options
Pipeline
Repository browser
(Auto)
Additional Behaviours
Add
Script Path
src/frontend/Jenkinsfile
Lightweight checkout

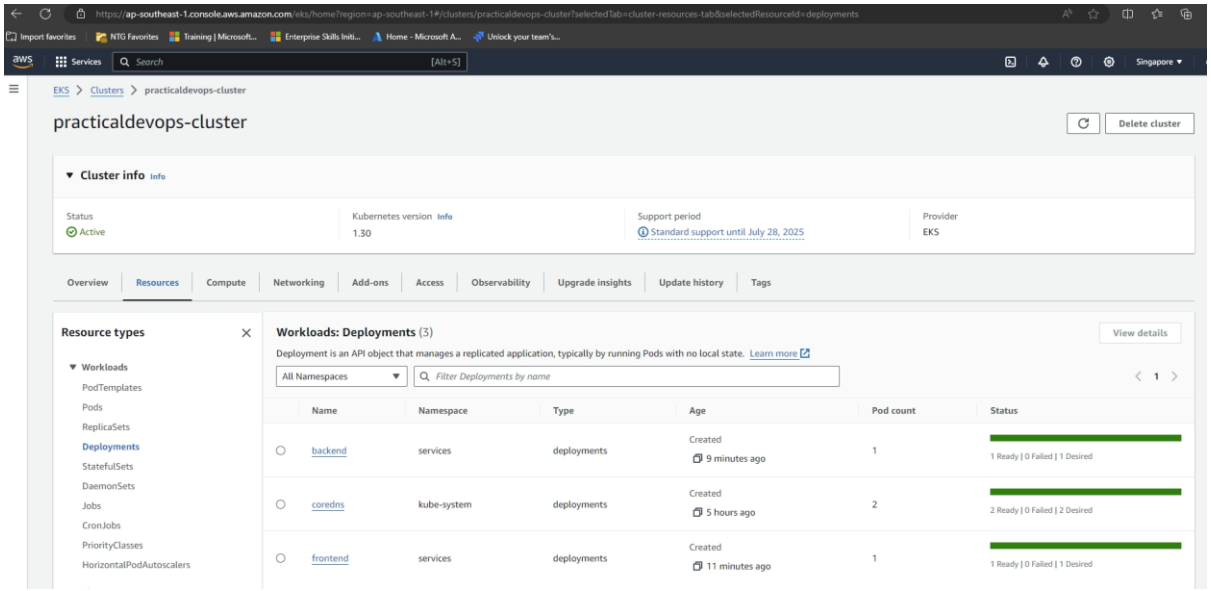
Configure AWS Access Key for Jenkins IAM user



Create Jenkins pipeline to build and deploy applications




Not secure
jkci-223104de.nip.io/job/NTG-Garage/job/frontend/26/console
Import favorites
NTG Favorites
Training | Microsoft...
Enterprise Skills Initi...
Home - Microsoft A...
Unlock your team's...
Dashboard
NTG-Garage
frontend
#26
speed
0 0 0 0 0 0 0 0 --:--:-- --:--:-- --:--:-- 0
100 138 100 138 0 0 580 0 --:--:-- --:--:-- --:--:-- 582
100 53.7M 100 53.7M 0 0 82.6M 0 --:--:-- --:--:-- --:--:-- 82.6M
+ chmod +x ./kubectl
+ aws eks update-kubeconfig --region ap-southeast-1 --name practicaldevops-cluster
Added new context arn:aws:eks:ap-southeast-1:063439157700:cluster/practicaldevops-cluster to /root/.kube/config
+ ./kubectl apply -f deployment/frontend.yaml -n services
service/frontend unchanged
deployment.apps/frontend created
[Pipeline] }
[Pipeline] // container
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] }
[Pipeline] // podTemplate
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // timeout
[Pipeline] }
[Pipeline] // withCredentials
[Pipeline] End of Pipeline
Finished: SUCCESS


EKS > Clusters > practicaldevops-cluster
practicaldevops-cluster
Delete cluster
Cluster info
Status
Active
Kubernetes version
1.30
Support period
Standard support until July 28, 2025
Provider
EKS
Overview
Resources
Compute
Networking
Add-ons
Access
Observability
Upgrade insights
Update history
Tags
Resource types
Workloads
PodTemplates
Pods
ReplicaSets
Deployments
StatefulSets
DaemonSets
Jobs
CronJobs
PriorityClasses
HorizontalPodAutoscalers
Workloads: Deployments (3)
View details
Deployment is an API object that manages a replicated application, typically by running Pods with no local state.
Learn more
All Namespaces
Filter Deployments by name
Name
Namespace
Type
Age
Pod count
Status
backend
services
deployments
Created 9 minutes ago
1
1 Ready | 0 Failed | 1 Desired
coredns
kube-system
deployments
Created 5 hours ago
2
2 Ready | 0 Failed | 2 Desired
frontend
services
deployments
Created 11 minutes ago
1
1 Ready | 0 Failed | 1 Desired

Setup Prometheus and Grafana to monitor EKS resource

Deploy Prometheus

```
(base) quytran@VNNOT01796:/mnt/d/training/Pratical DevOps/sd5129_msa$ helm repo add prometheus-community https://prometheus-community.github.io/helm-charts
"prometheus-community" has been added to your repositories
(base) quytran@VNNOT01796:/mnt/d/training/Pratical DevOps/sd5129_msa$ helm repo update
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "crossplane-stable" chart repository
...Successfully got an update from the "ingress-nginx" chart repository
...Successfully got an update from the "jetstack" chart repository
...Successfully got an update from the "kyverno" chart repository
...Successfully got an update from the "jenkinsci" chart repository
...Successfully got an update from the "argo" chart repository
...Successfully got an update from the "prometheus-community" chart repository
Update Complete. ⎈Happy Helming!⎈
(base) quytran@VNNOT01796:/mnt/d/training/Pratical DevOps/sd5129_msa$ helm install -n prometheus prometheus prometheus-community/prometheus
NAME: prometheus
LAST DEPLOYED: Tue Aug 20 16:01:30 2024
NAMESPACE: prometheus
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
The Prometheus server can be accessed via port 80 on the following DNS name from within your cluster:
prometheus-server.prometheus.svc.cluster.local


Get the Prometheus server URL by running these commands in the same shell:
  export POD_NAME=$(kubectl get pods --namespace prometheus -l "app.kubernetes.io/name=prometheus,app.kubernetes.io/instance=prometheus" -o jsonpath="{.items[0].metadata.name}")
  kubectl --namespace prometheus port-forward $POD_NAME 9090


The Prometheus alertmanager can be accessed via port 9093 on the following DNS name from within your cluster:
prometheus-alertmanager.prometheus.svc.cluster.local


Get the Alertmanager URL by running these commands in the same shell:
  export POD_NAME=$(kubectl get pods --namespace prometheus -l "app.kubernetes.io/name=alertmanager,app.kubernetes.io/instance=prometheus" -o jsonpath="{.items[0].metadata.name}")
  kubectl --namespace prometheus port-forward $POD_NAME 9093
#################################################################################
######   WARNING: Pod Security Policy has been disabled by default since    #####
######            it deprecated after k8s 1.25+. use                        #####
######            (index .Values "prometheus-node-exporter" "rbac"          #####
###### .          "pspEnabled") with (index .Values                         #####
######            "prometheus-node-exporter" "rbac" "pspAnnotations")       #####
######            in case you still need it.                                #####
#################################################################################


The Prometheus PushGateway can be accessed via port 9091 on the following DNS name from within your cluster:
prometheus-prometheus-pushgateway.prometheus.svc.cluster.local


Get the PushGateway URL by running these commands in the same shell:
```

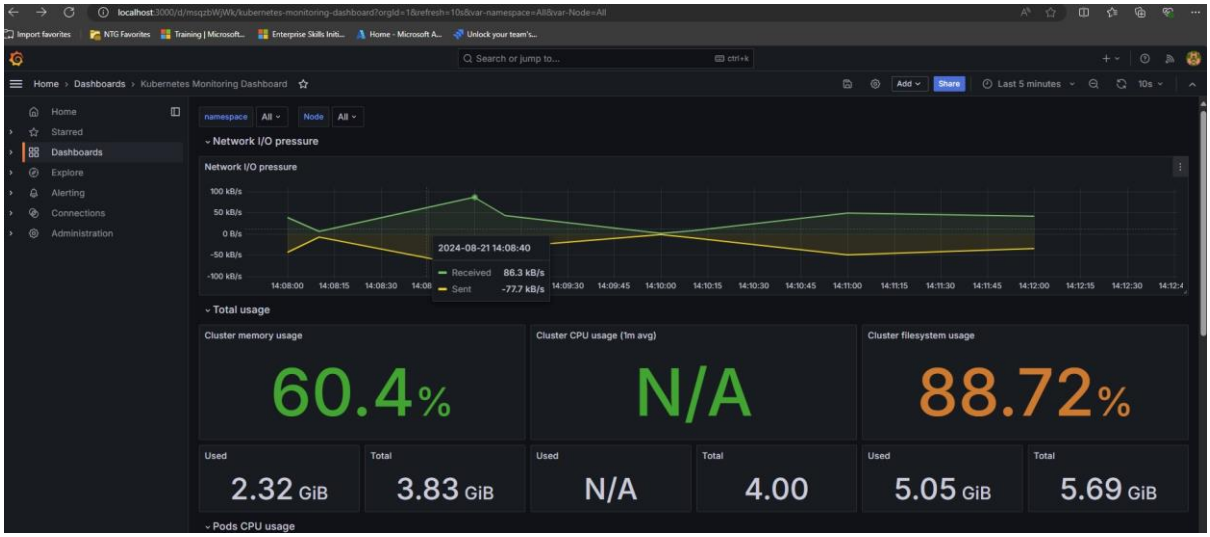
Deploy Grafana

environment > grafana > ! grafana.yaml > {} datasources > {} datasources.yaml > [ ] datasources > {} 0 > isDefault

```yaml
datasources:
  datasources.yaml:
    apiVersion: 1
    datasources:
    - name: Prometheus
      type: prometheus
      url: http://prometheus-server.prometheus.svc.cluster.local
      access: proxy
      isDefault: true
```

```
(base) quytran@VNNOT01796:/mnt/d/training/Pratical DevOps/infra$ helm repo add grafana https://grafana.github.io/helm-charts
"grafana" has been added to your repositories
(base) quytran@VNNOT01796:/mnt/d/training/Pratical DevOps/infra$ helm repo update
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "crossplane-stable" chart repository
...Successfully got an update from the "jetstack" chart repository
...Successfully got an update from the "ingress-nginx" chart repository
...Successfully got an update from the "kyverno" chart repository
...Successfully got an update from the "jenkinsci" chart repository
...Successfully got an update from the "prometheus-community" chart repository
...Successfully got an update from the "argo" chart repository
...Successfully got an update from the "grafana" chart repository
Update Complete. ⎈Happy Helming!⎈
(base) quytran@VNNOT01796:/mnt/d/training/Pratical DevOps/infra$ helm install grafana grafana/grafana \
    --namespace grafana \
> ^C
(base) quytran@VNNOT01796:/mnt/d/training/Pratical DevOps/infra$ ^C
(base) quytran@VNNOT01796:/mnt/d/training/Pratical DevOps/infra$ helm install grafana grafana/grafana \
> -n grafana \
> --values ./environment/grafana/grafana.yaml \
> --create-namespace
NAME: grafana
LAST DEPLOYED: Tue Aug 20 16:14:39 2024
NAMESPACE: grafana
STATUS: deployed
REVISION: 1
NOTES:
1. Get your 'admin' user password by running:

   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:
     export POD_NAME=$(kubectl get pods --namespace grafana -l "app.kubernetes.io/name=grafana,app.kubernetes.io/instance=grafana" -o jsonpath="{.items[0].metadata.name}")
     kubectl --namespace grafana port-forward $POD_NAME 3000

3. Login with the password from step 1 and the username: admin
#################################################################################
######   WARNING: Persistence is disabled!!! You will lose your data when   #####
######            the Grafana pod is terminated.                            #####
#################################################################################
```

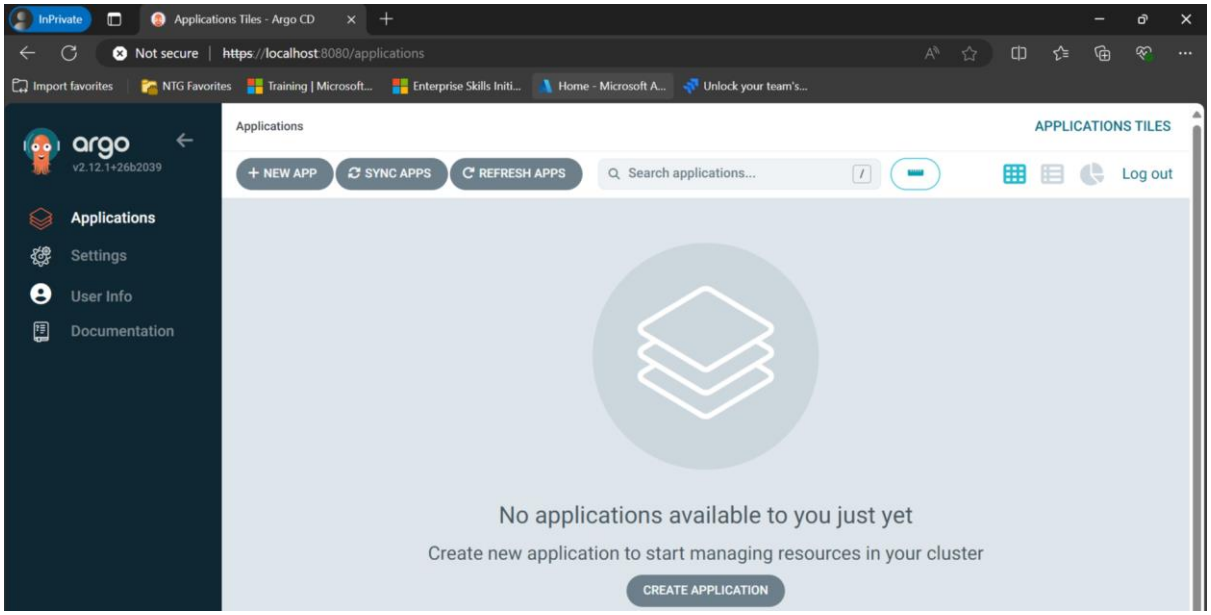
Query default EKS metrics



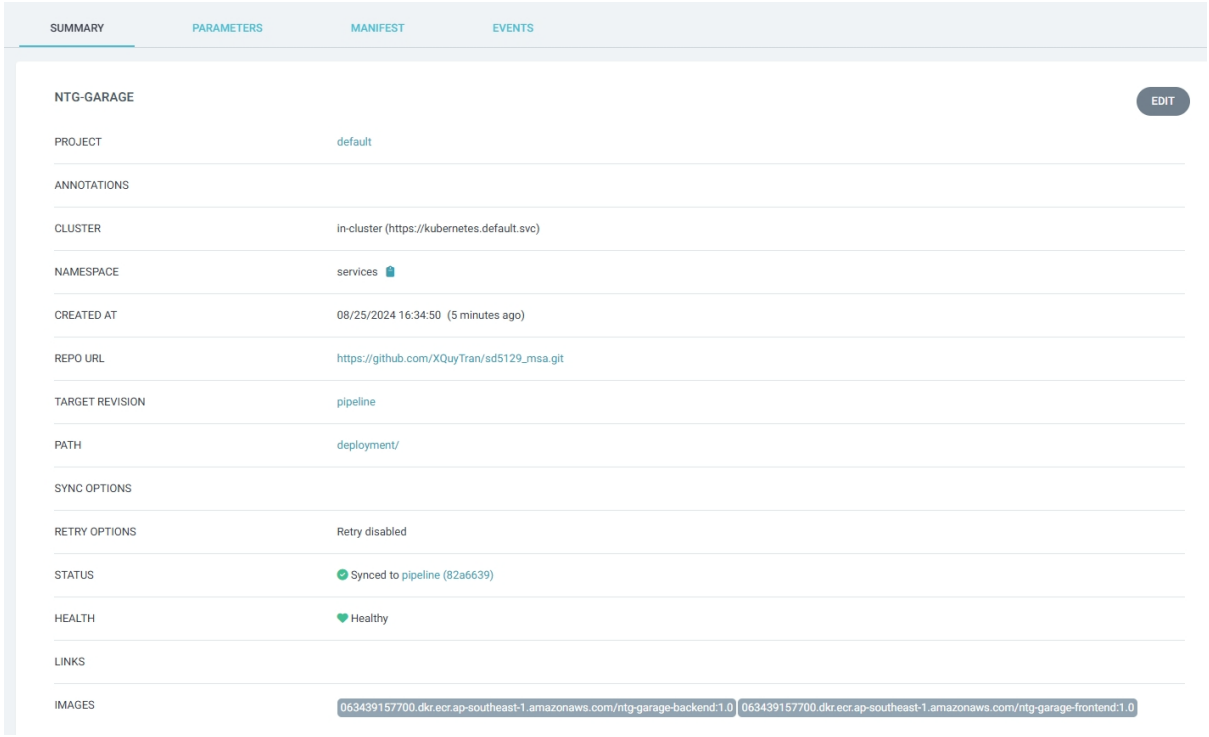
2. Reuse Jenkins CI and use GitOps for the CD pipeline.

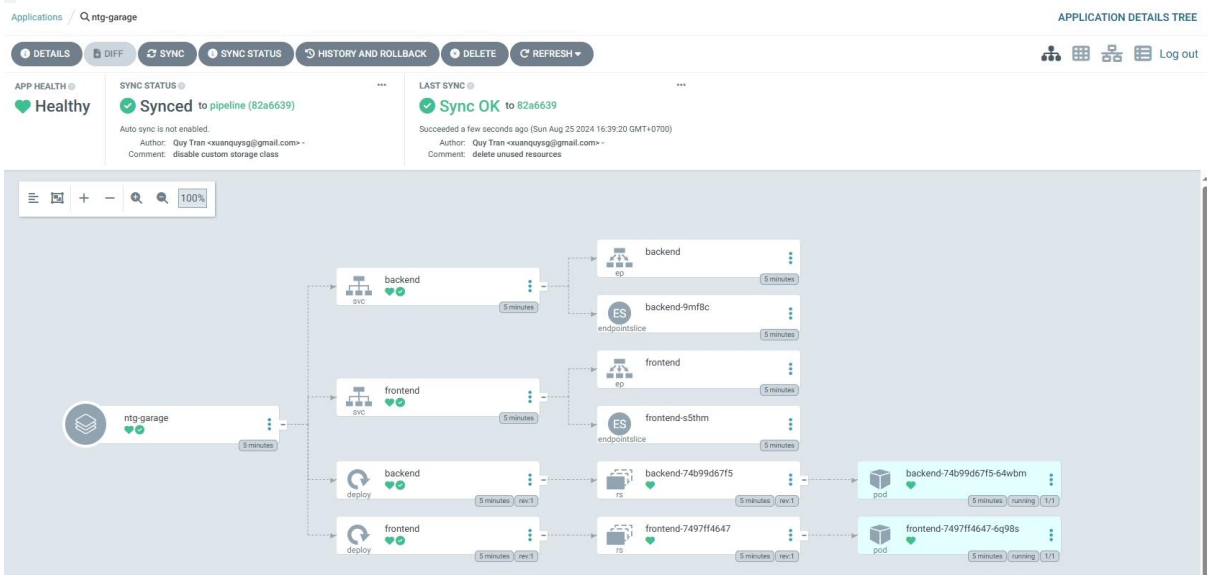
Install ArgoCD

```
(base) quytran@WNKOT01796:/mnt/d/training/Pratical DevOps/infra/aws$ helm install -n argocd --create-namespace argo-cd argo/argo-cd --version 7.4.4
NAME: argo-cd
LAST DEPLOYED: Wed Aug 21 14:58:38 2024
NAMESPACE: argocd
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
In order to access the server UI you have the following options:

1. kubectl port-forward service/argo-cd-argocd-server -n argocd 8080:443

    and then open the browser on http://localhost:8080 and accept the certificate

2. enable ingress in the values file `server.ingress.enabled` and either
      - Add the annotation for ssl passthrough: https://argo-cd.readthedocs.io/en/stable/operator-manual/ingress/#option-1-ssl-passthrough
      - Set the `configs.params."server.insecure"` in the values file and terminate SSL at your ingress: https://argo-cd.readthedocs.io/en/stable/operator-manual/ingress/#option-2-multiple-ingress-objects-and-hosts


After reaching the UI the first time you can login with username: admin and the random password generated during the installation. You can find the password by running:

kubectl -n argocd get secret argocd-initial-admin-secret -o jsonpath="{.data.password}" | base64 -d

(You should delete the initial secret afterwards as suggested by the Getting Started Guide: https://argo-cd.readthedocs.io/en/stable/getting_started/#4-login-using-the-cli)
```



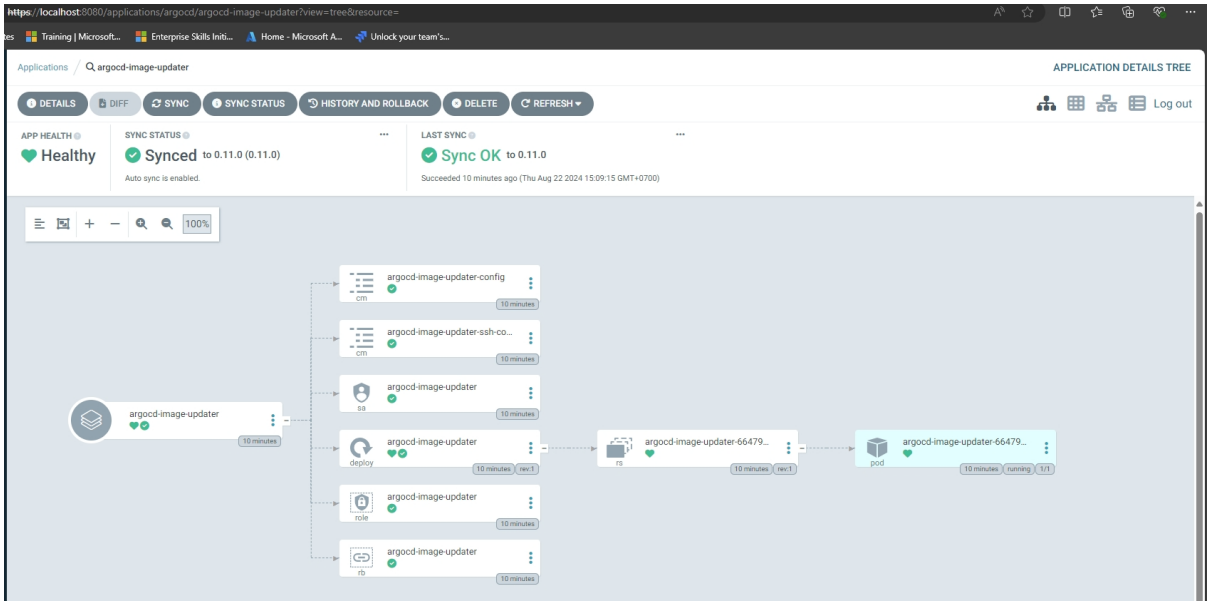
Create application in Argocd

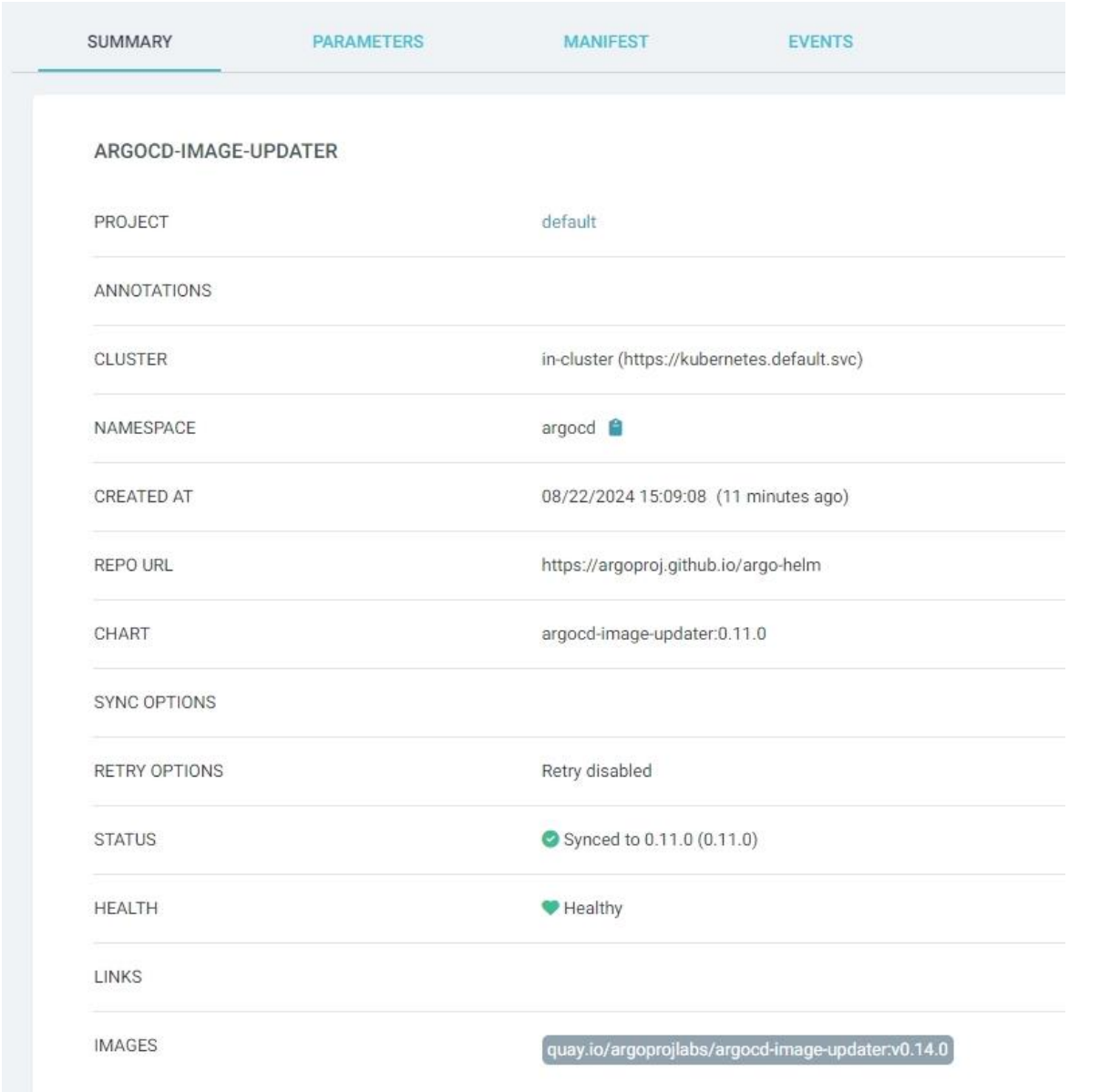




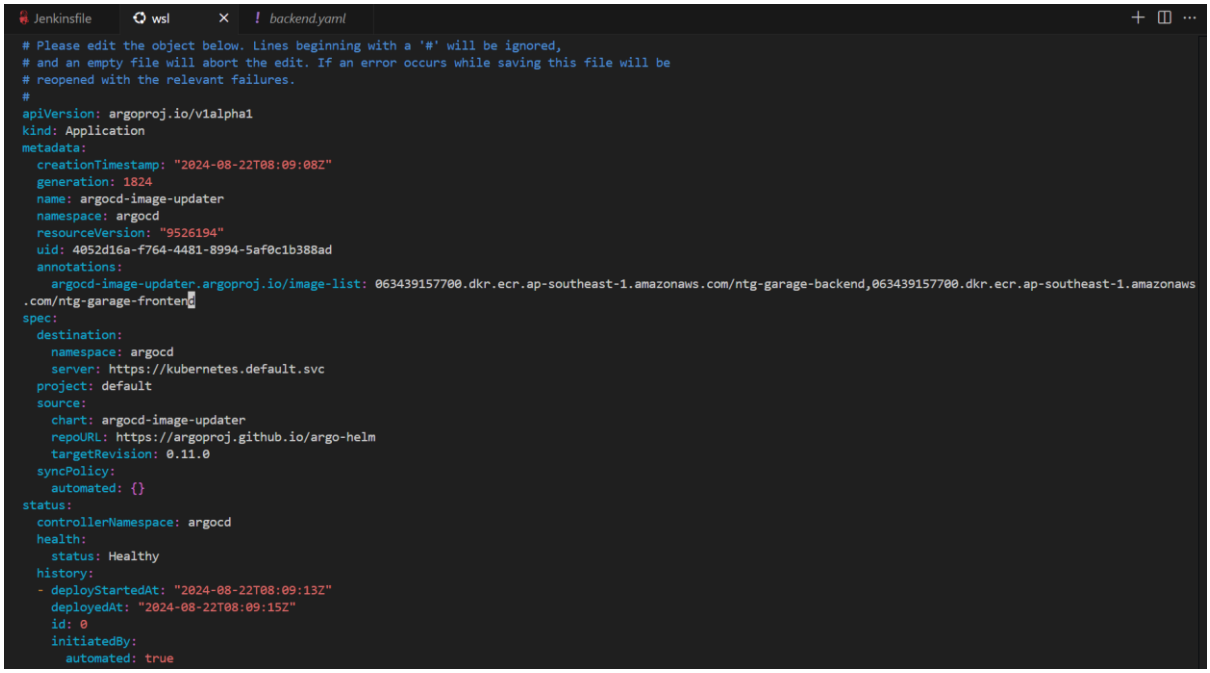
Update image with ArgoCD Image Updater

Install ArgoCD Image Updater





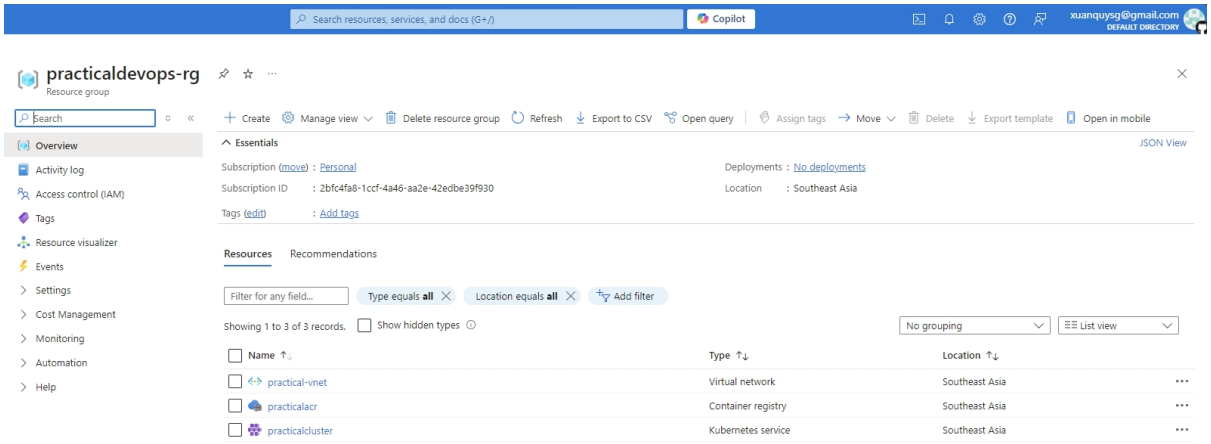
Marking application for image updatable



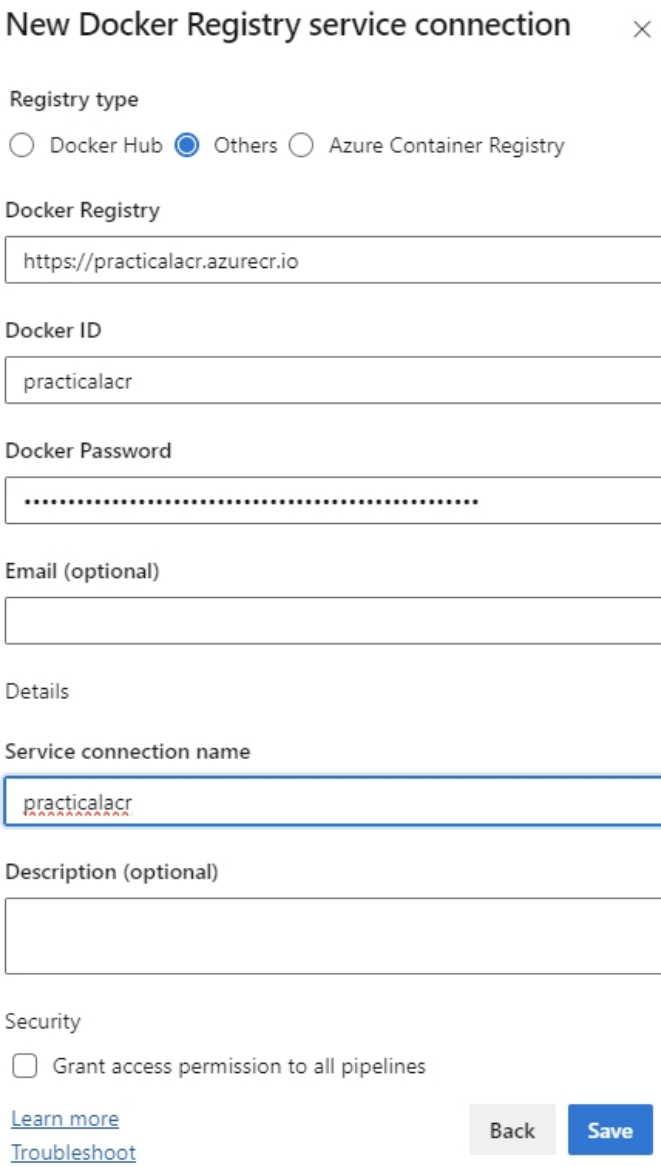
3. Setting up a CI/CD Pipeline and deploying applications on Azure AKS

Provision Azure resources

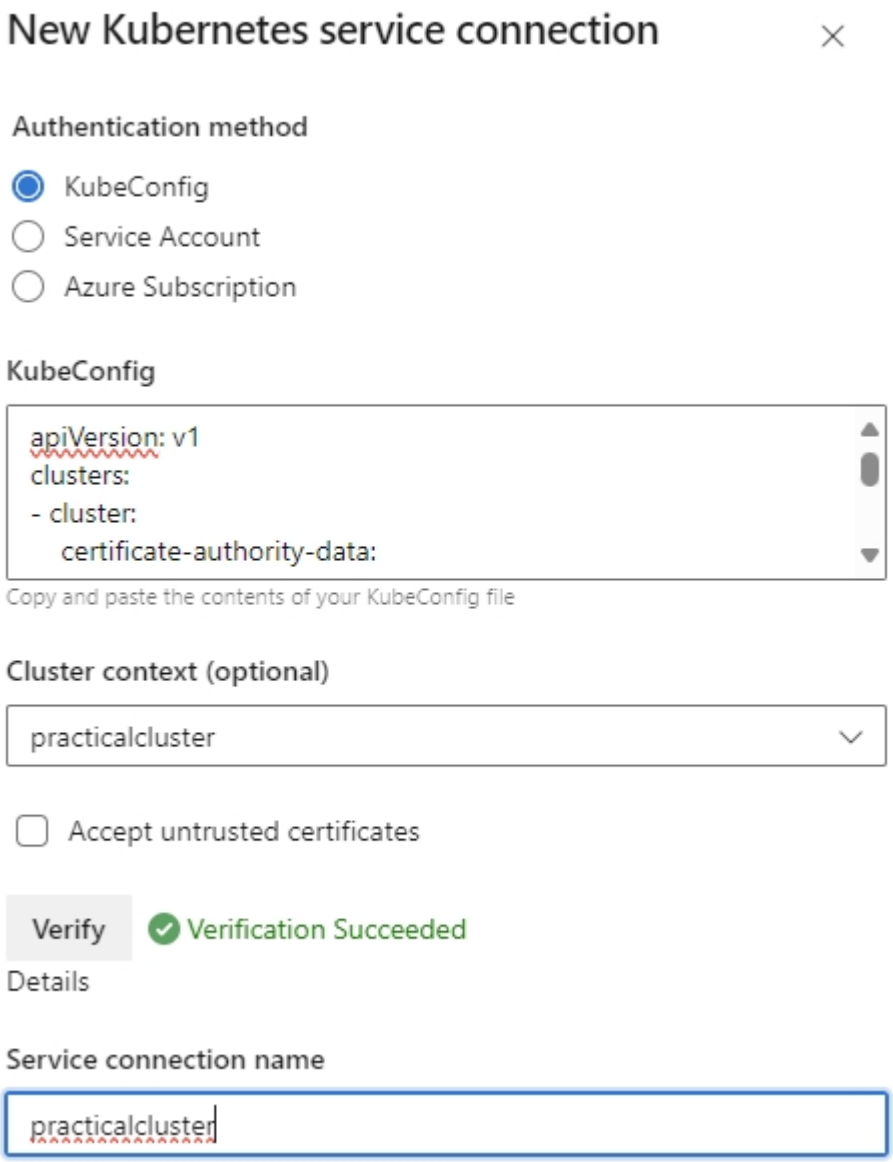
```
PS D:\training\Pratical DevOps\infra\azure> tf_1.5.7.exe apply

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # azurerm_container_registry.acr will be created
  + resource "azurerm_container_registry" "acr" {
      + admin_enabled                 = true
      + admin_password                = (sensitive value)
      + admin_username                = (known after apply)
      + encryption                    = (known after apply)
      + export_policy_enabled         = true
      + id                            = (known after apply)
      + location                      = "southeastasia"
      + login_server                  = (known after apply)
      + name                          = "practicalacr"
      + network_rule_bypass_option    = "AzureServices"
      + network_rule_set              = (known after apply)
      + public_network_access_enabled = true
      + resource_group_name           = "practicaldevops-rg"
      + sku                           = "Basic"
      + trust_policy_enabled          = false
      + zone_redundancy_enabled       = false
    }

  # azurerm_kubernetes_cluster.aks will be created
  + resource "azurerm_kubernetes_cluster" "aks" {
      + current_kubernetes_version          = (known after apply)
      + dns_prefix                          = "practicalcluster"
      + fqdn                                = (known after apply)
      + http_application_routing_zone_name  = (known after apply)
      + id                                  = (known after apply)
      + kube_admin_config                   = (sensitive value)
      + kube_admin_config_raw               = (sensitive value)
      + kube_config                         = (sensitive value)
      + kube_config_raw                     = (sensitive value)
      + kubernetes_version                  = (known after apply)
      + location                            = "southeastasia"
      + name                                = "practicalcluster"
      + node_os_upgrade_channel             = "NodeImage"
      + node_resource_group                 = (known after apply)
      + node_resource_group_id              = (known after apply)
      + oidc_issuer_url                     = (known after apply)
      + portal_fqdn                         = (known after apply)
      + private_cluster_enabled             = false
      + private_cluster_public_fqdn_enabled = false
      + private_dns_zone_id                 = (known after apply)
      + private_fqdn                        = (known after apply)
      + resource_group_name                 = "practicaldevops-rg"
```



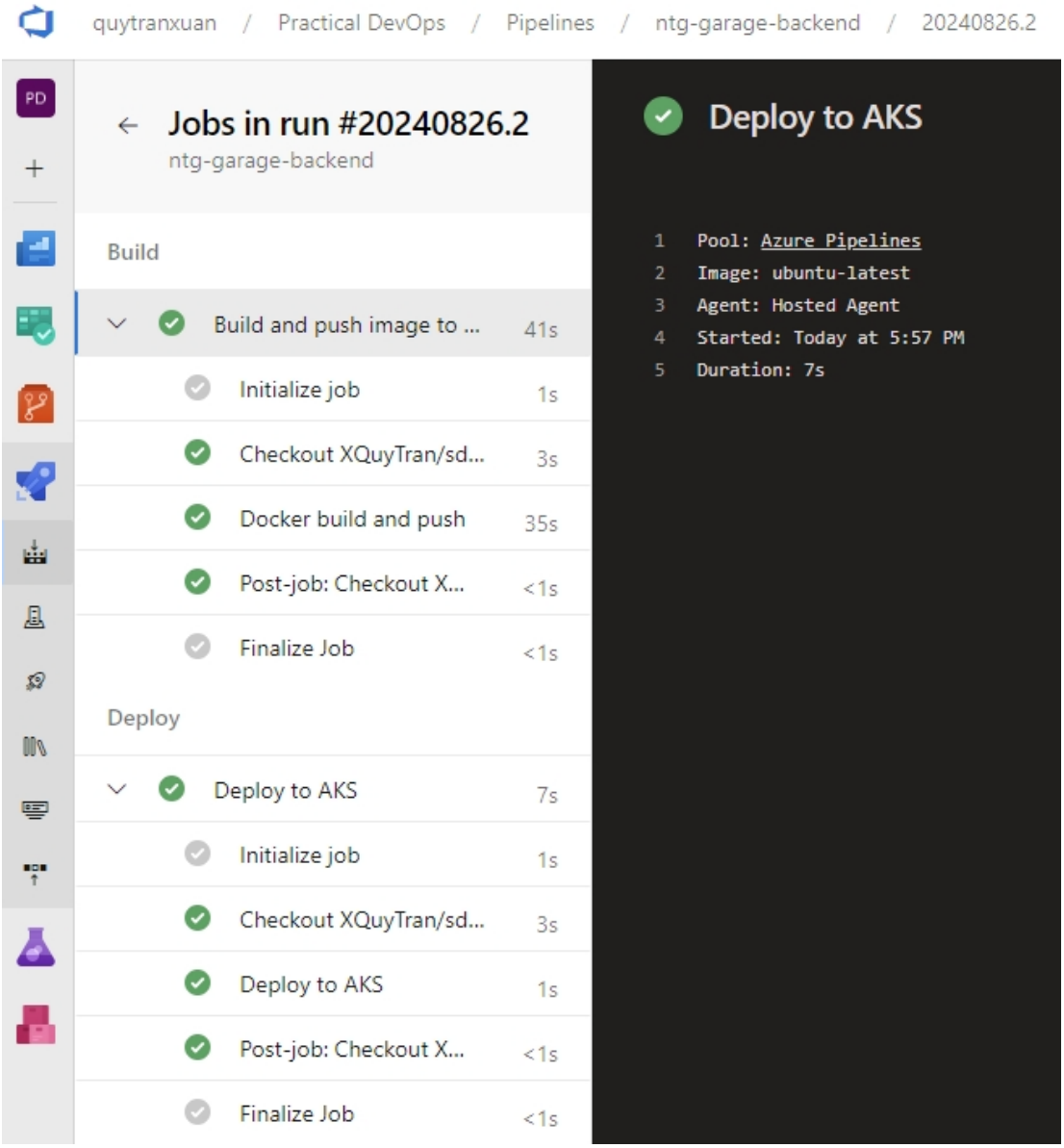
Create service connection for Azure Container Registry

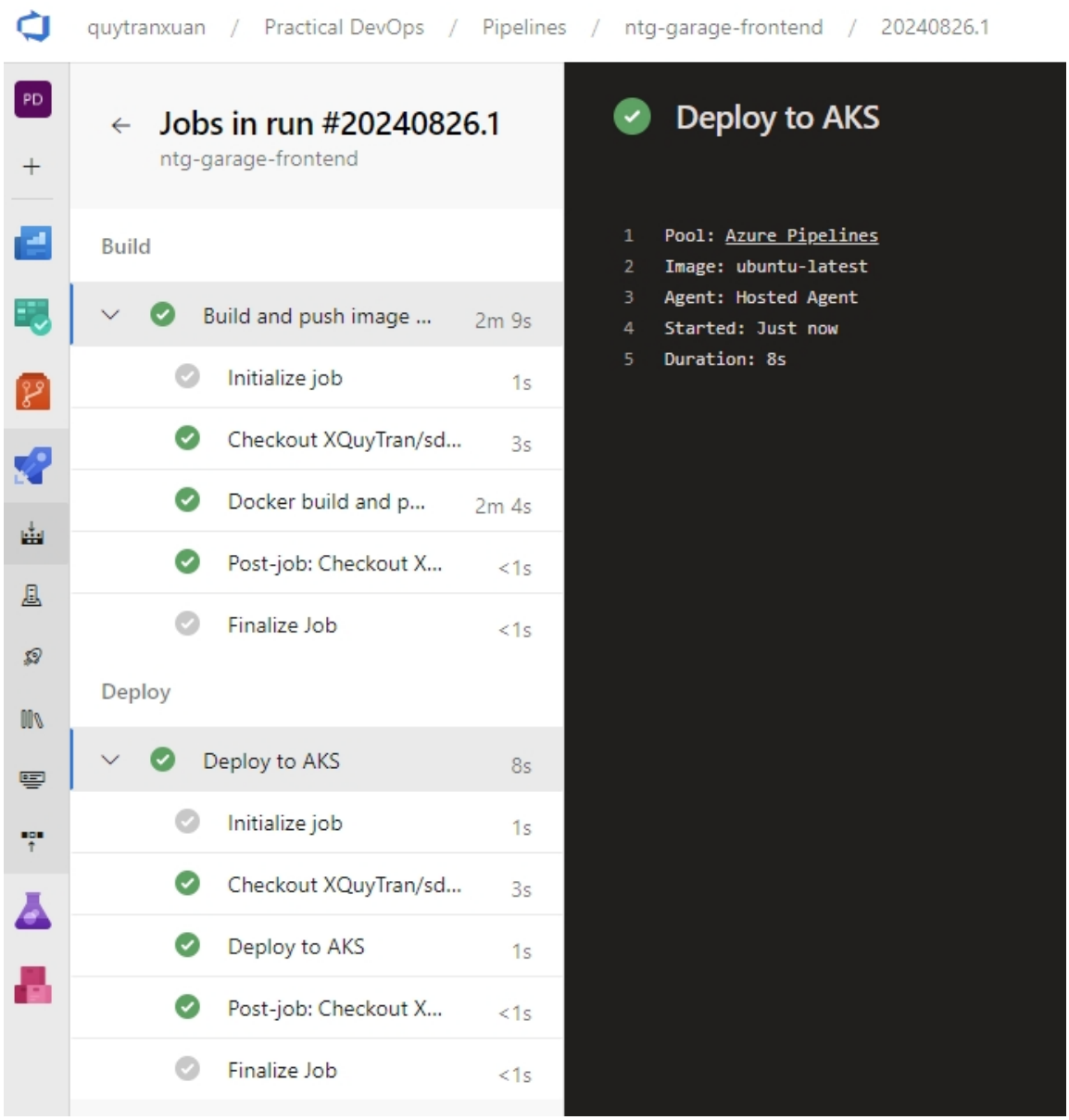


Create service connection for Kubernetes cluster



Create deployment pipeline




quytranxuan / Practical DevOps / Pipelines / ntg-garage-frontend / 20240826.1
Jobs in run #20240826.1
ntg-garage-frontend
Build
Build and push image ... 2m 9s
Initialize job 1s
Checkout XQuyTran/sd... 3s
Docker build and p... 2m 4s
Post-job: Checkout X... <1s
Finalize Job <1s
Deploy
Deploy to AKS 8s
Initialize job 1s
Checkout XQuyTran/sd... 3s
Deploy to AKS 1s
Post-job: Checkout X... <1s
Finalize Job <1s
Deploy to AKS
1 Pool: Azure Pipelines
2 Image: ubuntu-latest
3 Agent: Hosted Agent
4 Started: Just now
5 Duration: 8s