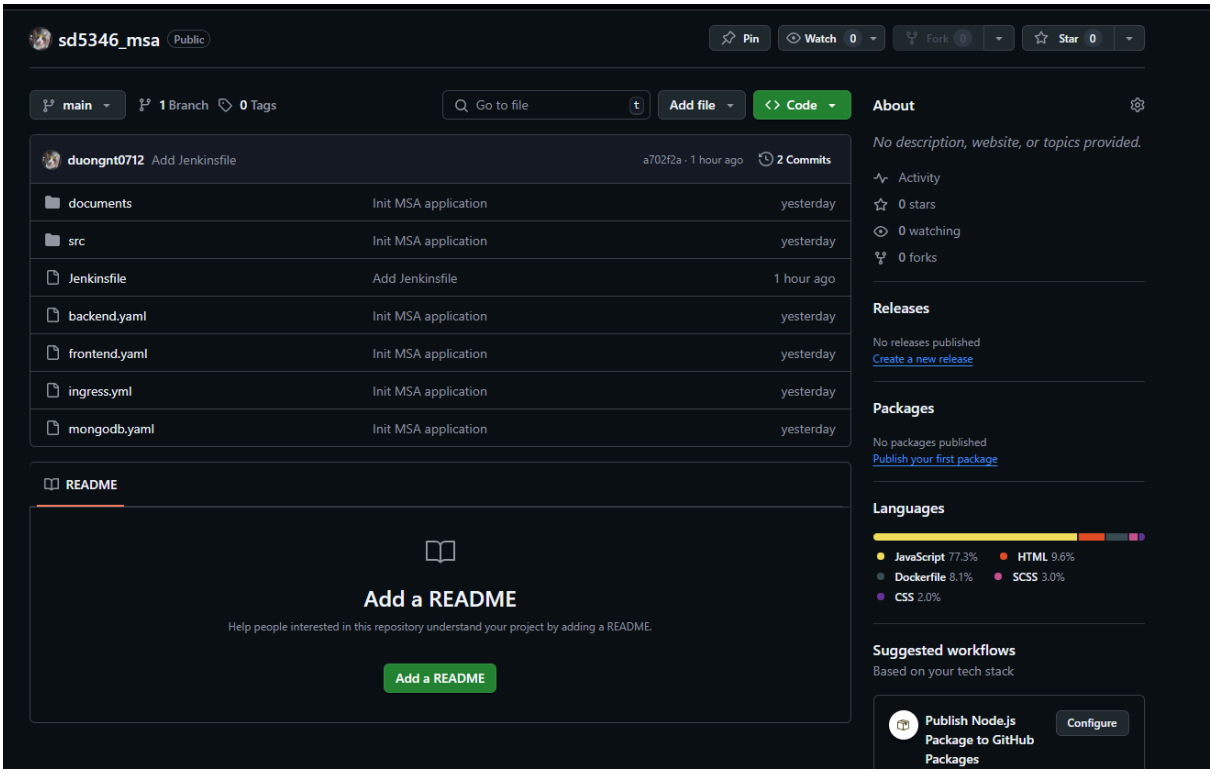
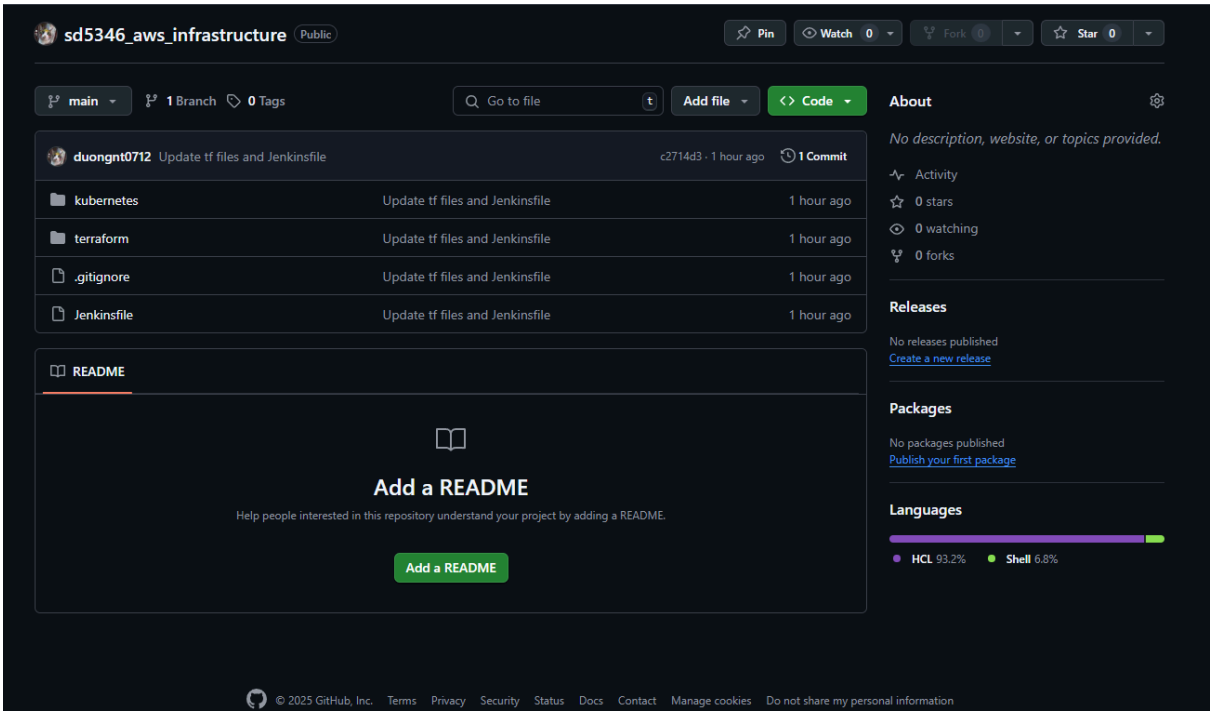
1. Setting up a CI/CD Pipeline and deploying applications on AWS EKS

1 - Source code Management

duongnt0712/sd5346_msa



duongnt0712/sd5346_aws_infrastructure



2 - Provision AWS resources

- VPC

```
# vpc module to create vpc, subnets, NATs, IGW etc..
module "vpc_and_subnets" {
  # invoke public vpc module
  source  = "terraform-aws-modules/vpc/aws"
  version = "5.0.0"

  # vpc name
  name = var.name

  # availability zones
  azs = slice(data.aws_availability_zones.available.names, 0, 3)

  # vpc cidr
  cidr = var.vpc_cidr

  # public and private subnets
  private_subnets = local.private_subnets
  public_subnets  = local.public_subnets

  # create nat gateways
  enable_nat_gateway     = var.enable_nat_gateway
  single_nat_gateway     = var.single_nat_gateway
  one_nat_gateway_per_az = var.one_nat_gateway_per_az

  # enable dns hostnames and support
  enable_dns_hostnames = var.enable_dns_hostnames
  enable_dns_support   = var.enable_dns_support

  # tags for public, private subnets and vpc
  tags                 = var.tags
  public_subnet_tags   = var.additional_public_subnet_tags
  private_subnet_tags  = var.additional_private_subnet_tags

  # create internet gateway
  create_igw       = var.create_igw
  instance_tenancy = var.instance_tenancy

}
```

- EC2

```
# Security Group for Jenkins EC2
resource "aws_security_group" "jenkins_sg" {
  name        = "practical-devops-jenkins-sg"
  description = "Allow SSH and HTTP"
  vpc_id      = module.network.vpc_id

  ingress {
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  ingress {
    from_port   = 8080
    to_port     = 8080
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }
}

resource "aws_instance" "jenkins" {
  key_name = var.jenkin_key_name

  ami                         = data.aws_ami.amazon_linux_2.id
  instance_type               = "t3.medium"
  subnet_id                   = module.network.public_subnets[0]
  vpc_security_group_ids      = [aws_security_group.jenkins_sg.id]
  associate_public_ip_address = true

  user_data = file("init-jenkins.sh")

  tags = module.jenkins-tags.tags
}
```

- ECR

terraform > modules > ecr > main.tf > ...

```
module "ecr_tags" {
  source = "../tags"

  name        = var.name
  project     = var.project
  environment = var.environment
  owner       = var.owner

  tags = {
    Description = "managed by terraform",
  }
}

resource "aws_ecr_repository" "default" {
  name                 = "${var.project}/${var.name}"
  image_tag_mutability = "MUTABLE"

  image_scanning_configuration {
    scan_on_push = var.enable_scan_on_push
  }

  tags = module.ecr_tags.tags
}
```

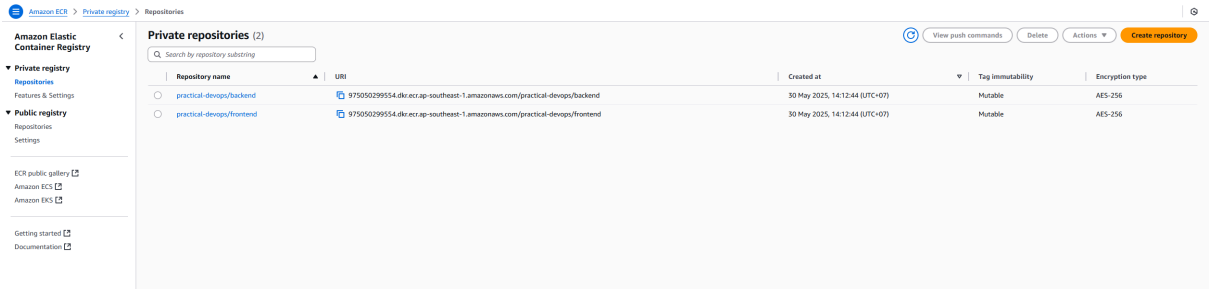
- EKS

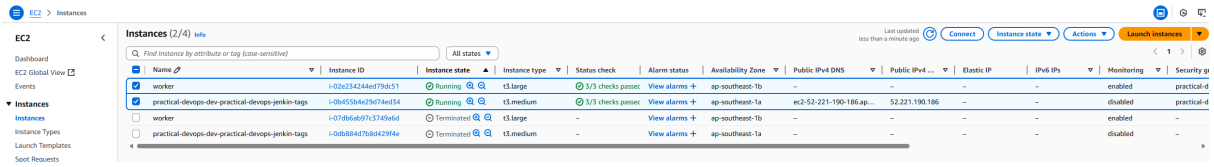
```
erraform > modules > eks > main.tf > module "eks" > version
module "eks" {
  # invoke public eks module
  source  = "terraform-aws-modules/eks/aws"
  version = "19.15.3"

  # eks cluster name and version
  cluster_name    = var.eks_cluster_name
  cluster_version = var.k8s_version

  # vpc id where the eks cluster security group needs to be created
  vpc_id = var.vpc_id

  # subnets where the eks cluster needs to be created
  control_plane_subnet_ids = var.control_plane_subnet_ids

  # to enable public and private access for eks cluster endpoint
  cluster_endpoint_private_access = true
  cluster_endpoint_public_access  = true

  # create an OpenID Connect Provider for EKS to enable IRSA
  enable_irsa = true

  # install eks managed addons
  # more details are here - https://docs.aws.amazon.com/eks/latest/userguide/eks-add-ons.html
  cluster_addons = {
    # extensible DNS server that can serve as the Kubernetes cluster DNS
    coredns = {
      preserve    = true
      most_recent = true
    }

    # maintains network rules on each Amazon EC2 node. It enables network communication to your Pods
    kube-proxy = {
      most_recent = true
    }

    # a Kubernetes container network interface (CNI) plugin that provides native VPC networking for your cluster
    vpc-cni = {
      most_recent = true
    }
  }

  # subnets where the eks node groups needs to be created
  subnet_ids = var.eks_node_groups_subnet_ids

  # eks managed node group named worker
  eks_managed_node_groups = var.workers_config

}
```

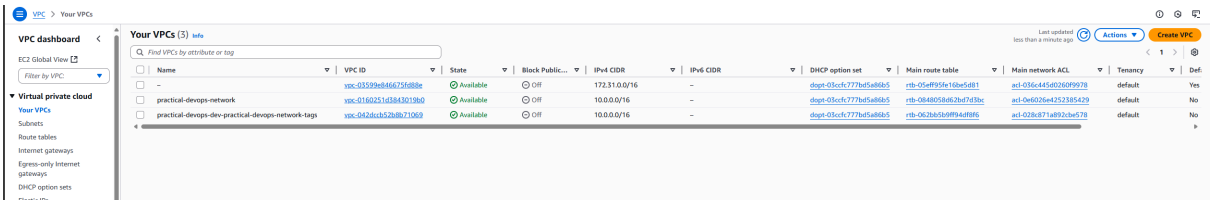
Run terraform

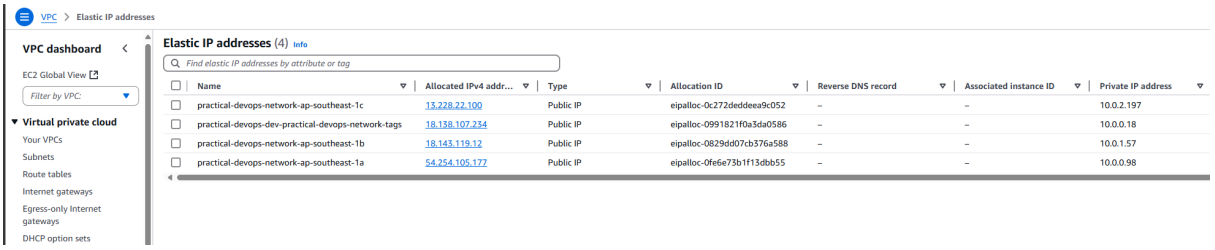
```
module.eks.module.eks.aws_eks_addon.this["kube-proxy"]: Still creating... [00m20s elapsed]
module.eks.module.eks.aws_eks_addon.this["coredns"]: Still creating... [00m20s elapsed]
module.eks.module.eks.aws_eks_addon.this["vpc-cni"]: Still creating... [00m30s elapsed]
module.eks.module.eks.aws_eks_addon.this["kube-proxy"]: Still creating... [00m30s elapsed]
module.eks.module.eks.aws_eks_addon.this["coredns"]: Still creating... [00m30s elapsed]
module.eks.module.eks.aws_eks_addon.this["vpc-cni"]: Still creating... [00m40s elapsed]
module.eks.module.eks.aws_eks_addon.this["kube-proxy"]: Still creating... [00m40s elapsed]
module.eks.module.eks.aws_eks_addon.this["coredns"]: Still creating... [00m40s elapsed]
module.eks.module.eks.aws_eks_addon.this["coredns"]: Creation complete after 45s [id=practical-devops-eks:coredns]
module.eks.module.eks.aws_eks_addon.this["kube-proxy"]: Creation complete after 45s [id=practical-devops-eks:kube-proxy]
module.eks.module.eks.aws_eks_addon.this["vpc-cni"]: Creation complete after 46s [id=practical-devops-eks:vpc-cni]

Warning: Argument is deprecated

  with module.eks.module.eks.aws_eks_addon.this["coredns"],
  on .terraform\modules\eks.eks\main.tf line 392, in resource "aws_eks_addon" "this":
  392:   resolve_conflicts        = try(each.value.resolve_conflicts, "OVERWRITE")

The "resolve_conflicts" attribute can't be set to "PRESERVE" on initial resource creation. Use "resolve_conflicts_on_create" and/or "resolve_conflicts_on_update" instead

(and 2 more similar warnings elsewhere)


Apply complete! Resources: 91 added, 0 changed, 0 destroyed.

Outputs:

cluster_certificate_authority_data = "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURCVENDQWJ6ZjBFUR5BtLS0tCk1JSURCVENDQWJ6ZjBJ0ZB1JTFc2QUFoa3Ad0F3RFFZSktvWk1odmNOQVFFTE3RQKdXVEVUTUFR0ExVUUKQXhNS2ErVmlaWEp1bl5bGN6QWVGdzB5TlRBMU16QkdOekV5TVRkYUZ3MHpOVEExTWpnd056RTNWVGRhTUJVeApFekFSQmdDVkJB1VRDbXQxbWI1WkhOtVjBuWE13ZzdFa
U1BMEdDU3FHU01PNRRRUJ8UAVBQTRJQkR3QKdn20VLCkFvSU8BUURFZehObXZ1UI1NvVlRST1kNFJMNNNrcXFIVzMTeTFxT2p4MC9OUXE3d8dOV1RE1VktIN1VEVmU1Z2QQUjdXbz86ck1CbVxZ7ZtYS94MRNvT3M5OWNaY3VJVZhQSkdhemRTSWUvSVEsMXASaDBJYVphL03qZGSxaWFNQgpVSUNxYk9QRD10aitnQUcSL0RhTWtLNWxPTVoZ2HUWHhLe1pSSVNDeUsvQTfNYXZpd8ho4mRD
MZpFW4duc4FIC1JncGNETI9lOVdZ3g0ZW9rZTVWpMNWV2E3a2JXb3Ixcn1aSU1aWT7WzHKDWD3MAk5ZNnNoBsFFUUVBNQV8hdHcKSzxGQQZneGNhY0xDYWU2eGRYdE1L5ZhvcEQzaFh1WmZDK0FVcUZ6WuNtTWX3UyRLOH8PNx9NLMmwrVns3JNjFsdwpaeXJIbKZyMUZ1HwNRN5BvBQ8S5TZGRzBoQkZEQ3WdNQkFBR2pXVtJYTUE8R8EvWuRE6NVCL3dRRUFJSJNwREFQCk3vTlI1UkICQAV4RUJ
UQURBUgoYUIwRExWdEZ1YXQCJRWTRxaXpoTmFWVVNMNEh1UXMbRTCid3hlRy9xQXVQedXNAYSRUVEakFNZ2dwa0QSmxjbTVSvZEwkqIBMEdDU3FHA01PNRRRUJ8DdVBQTRJQkFRQZZuaTg0wmkXQQQZTNTTRESRjhK2WVJWkSFlMzpSKZZNVVFNlZ1L2dNVFdrMNNNrIRSQWdlTkSHUNRCDWlcODkxbFp2EN8MWbRKcKQ9KWVFwkSQJAzMKpwcZ8TDF3YYYzeVVS3j6R8E3YKAZMTk
p1aFhWMVpaWklnbDQ3b23HaVhVWWhNSjE3TZQ0WEYxZVBYVVGZRDlnYzR0cINzUHovL2VlMlNSNfJVMNVncGFuWFIL01uVlZ2bWVMWdhK03CTZx0MWtUwkVVSwpzQZMSaG9NUUtRY2pjZ3dvaTBaazA3eVlkaRhOL3dVOGpxR3h2SHEzbn8ITE3LaSt2O0hRTdRNKxnbH1xa25TCKYMXDV3ZHpqV1p2Qksz amt5WWY0T24xRHo4eDFSRNF1UXNOaUxFNZVraWNYjWNdE1CNDdsOGd5c1RQQ
UQ4ekcXR01WlI1WG3GVEVuC10tLS0tRU5EIENFUlRJ0RkIDQ0RFLS0tLS0K"
cluster_endpoint = "https://9B051CE0717BD5501ADE8B8BC3CC8E18.yl4.ap-southeast-1.eks.amazonaws.com"
cluster_oidc_issuer_url = "https://oidc.eks.ap-southeast-1.amazonaws.com/id/9B051CE0717BD5501ADE8B8BC3CC8E18"
PS E:\Github-Repositories\Practical-DevOps-Developer-2025\sd5346_aws_infrastructure\terraform>
```

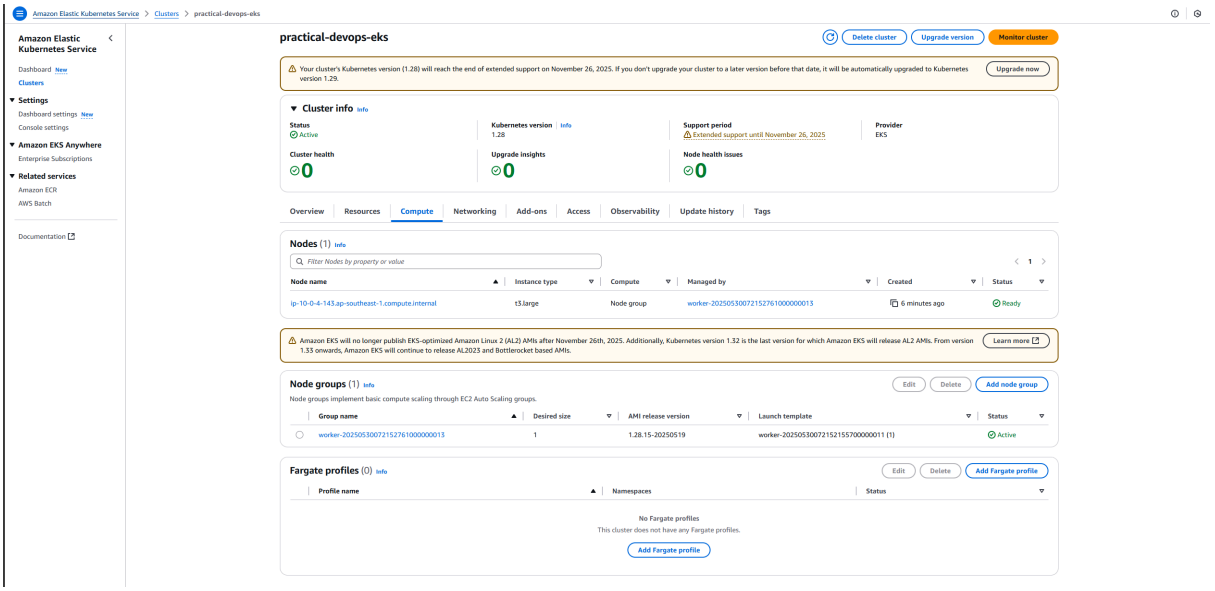
When it’s done, verify AWS resources:



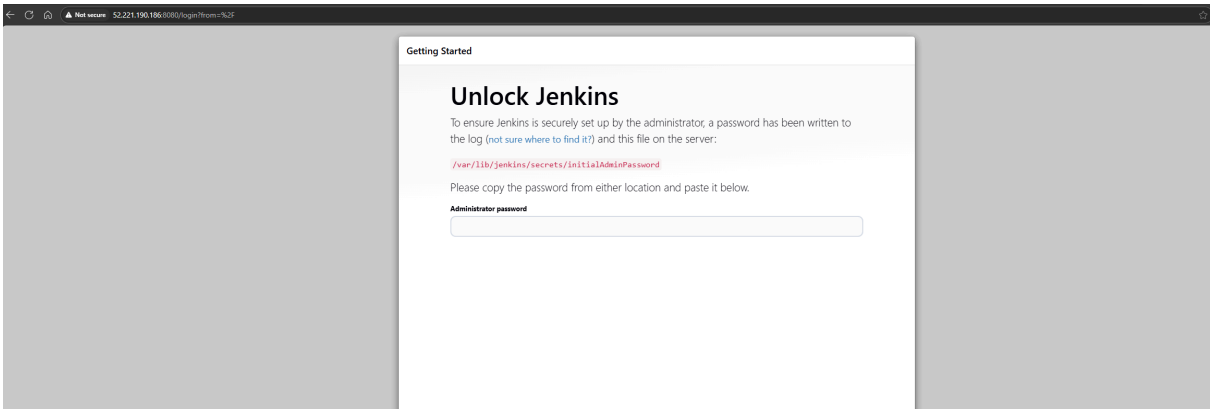


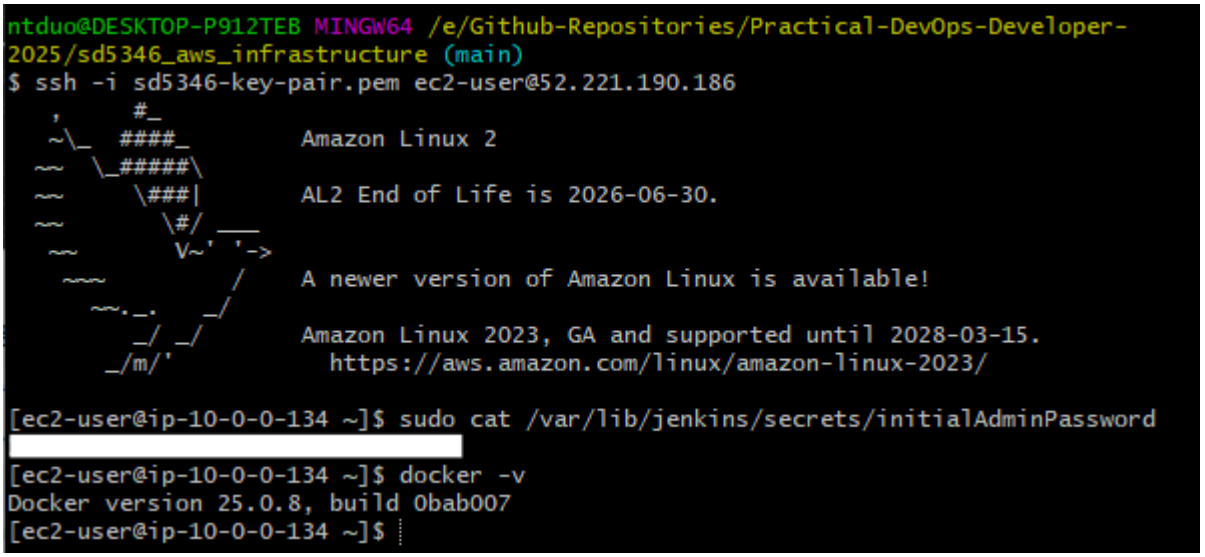


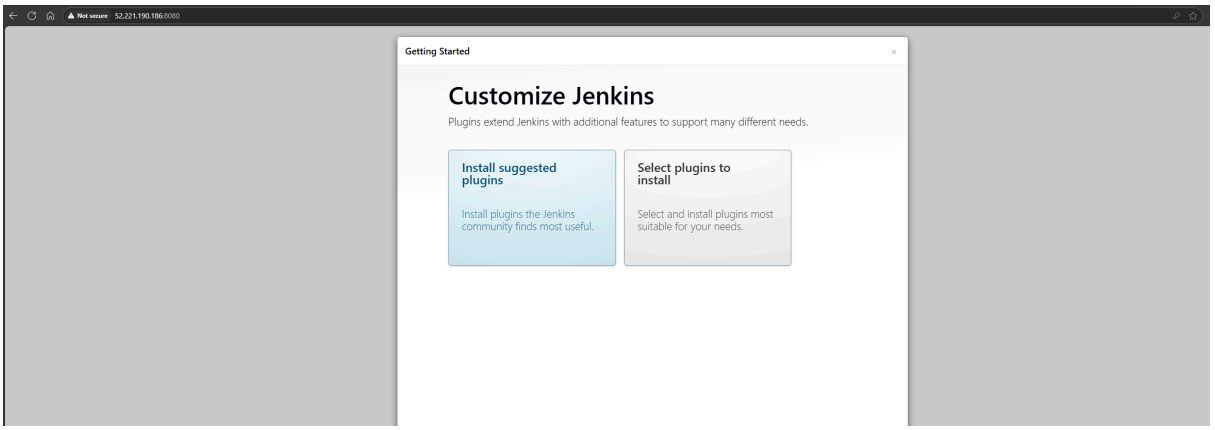




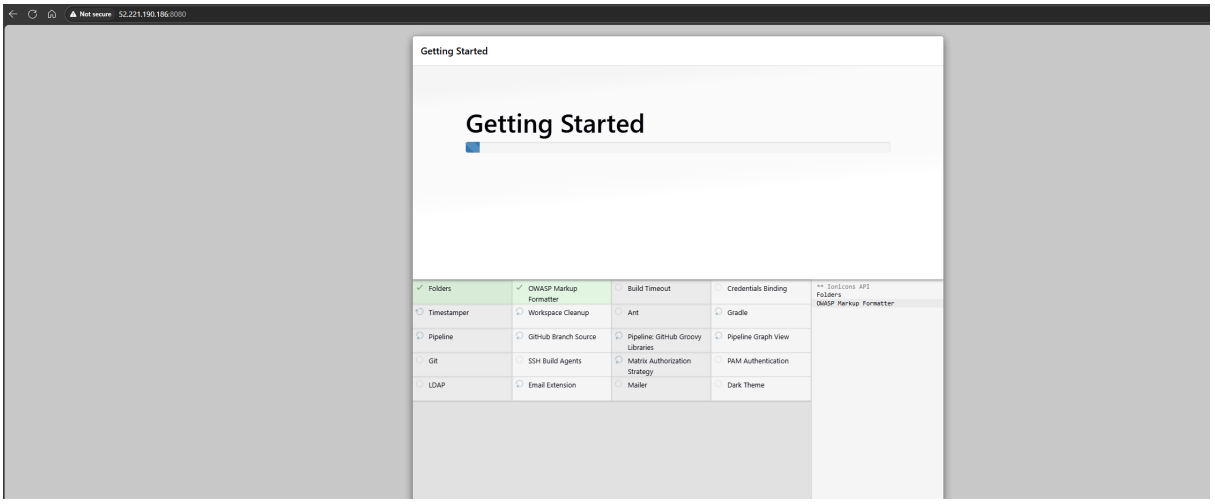
3 - Docker and Jenkins servers installation on EC2

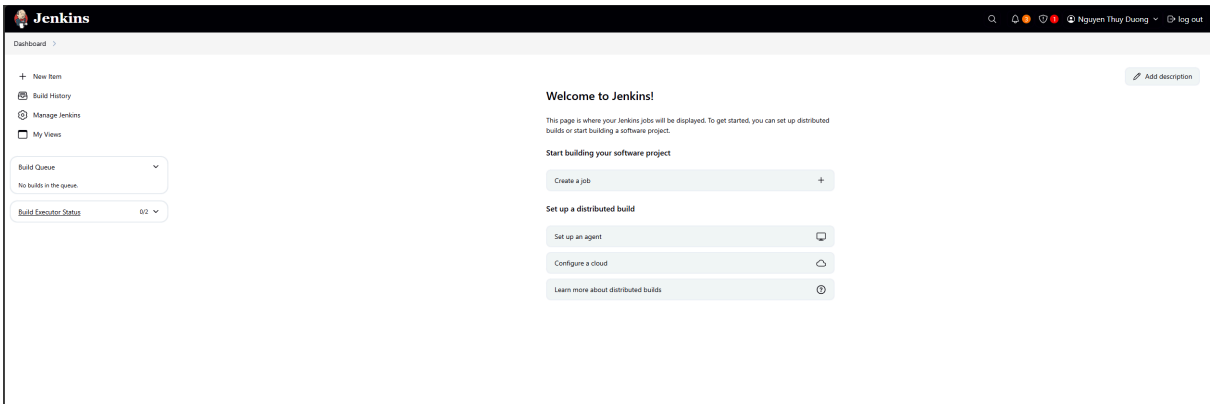






Choose option Install suggested plugins

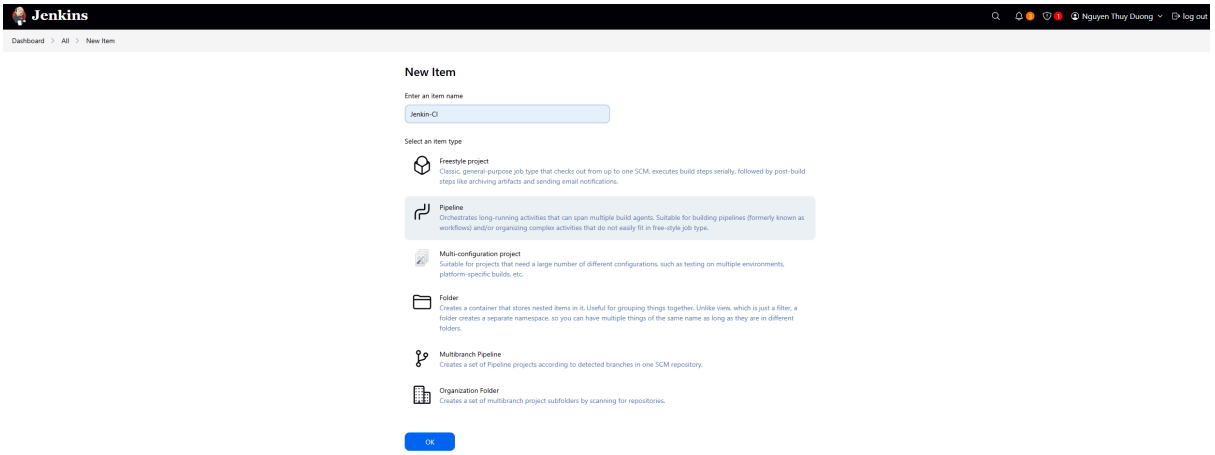


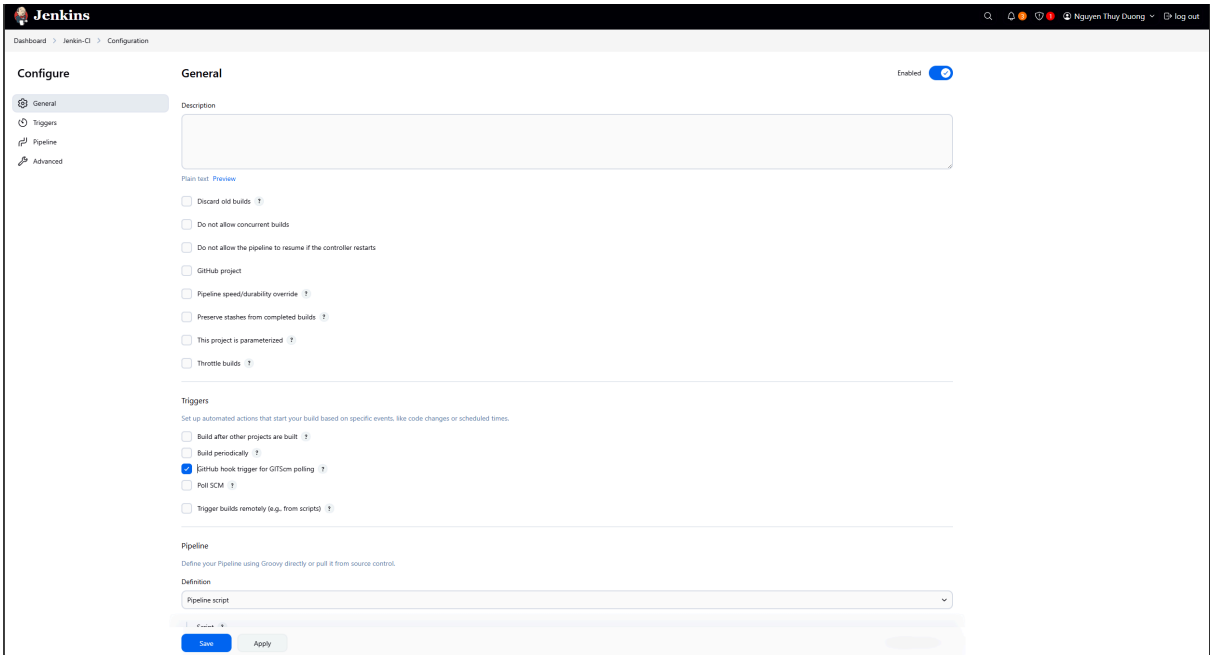


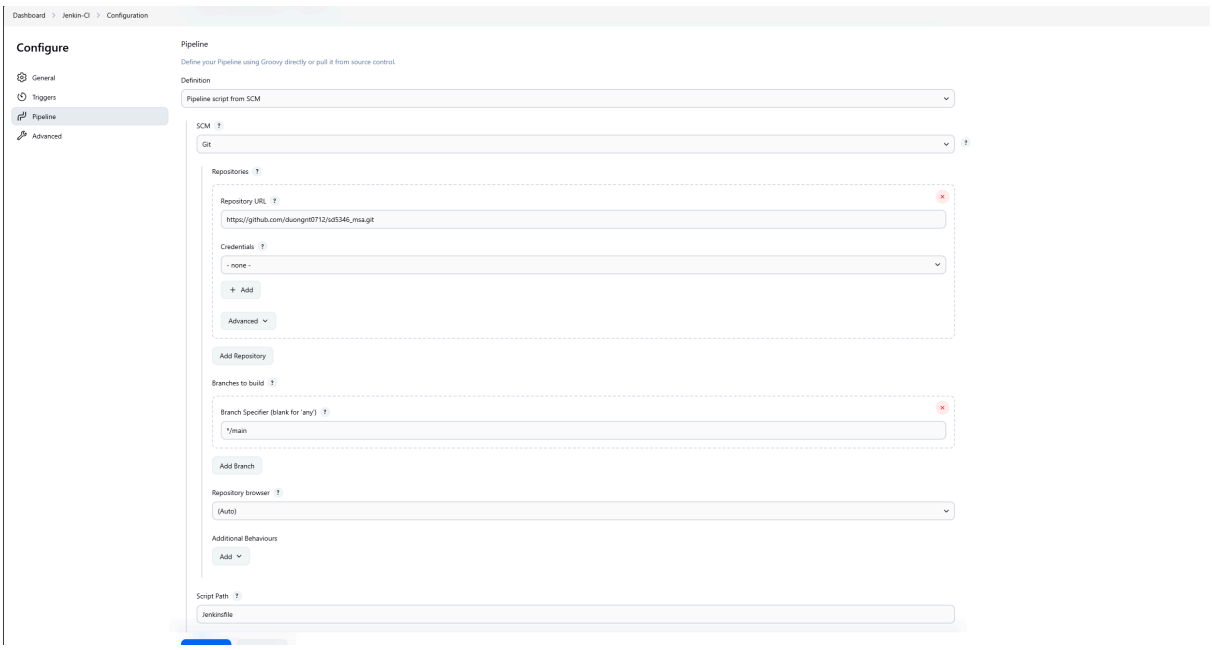
Installed related plugins

4 - Set up Jenkins pipeline for CI/CD

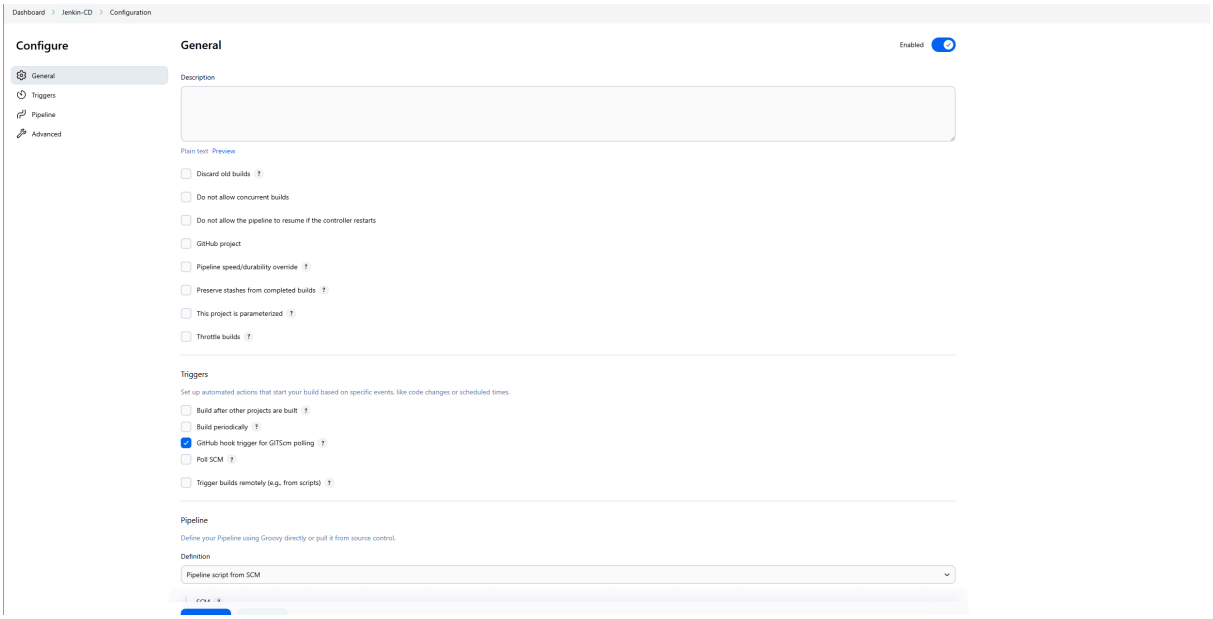
Create Jenkin-CI pipeline

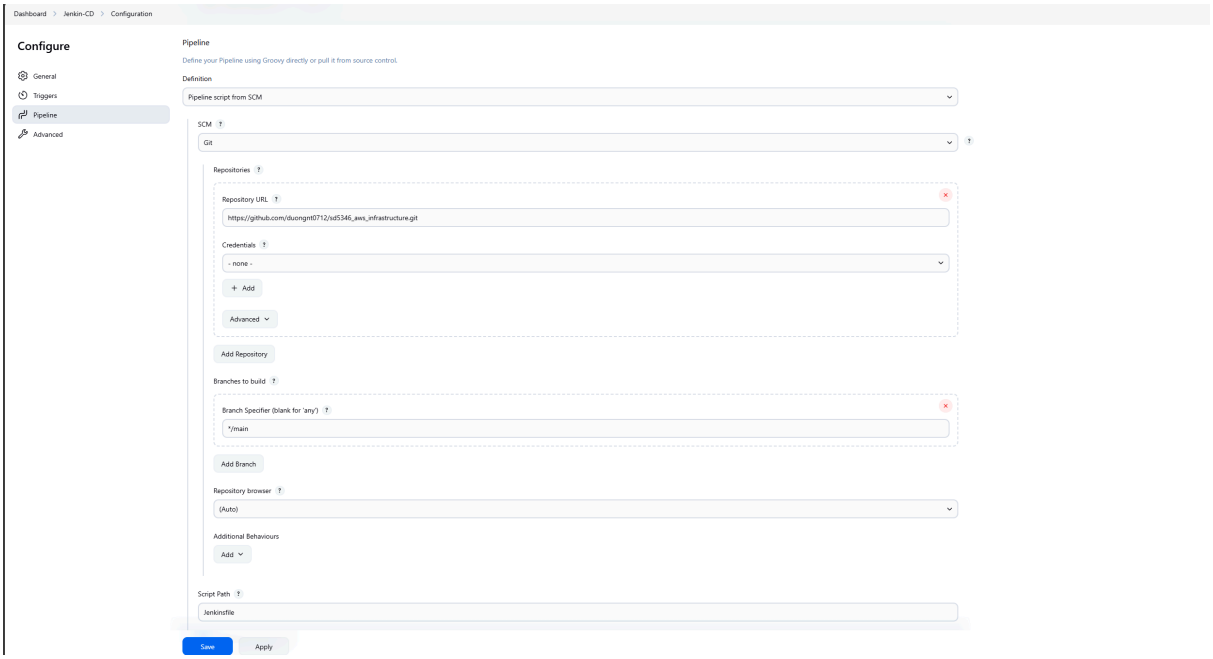




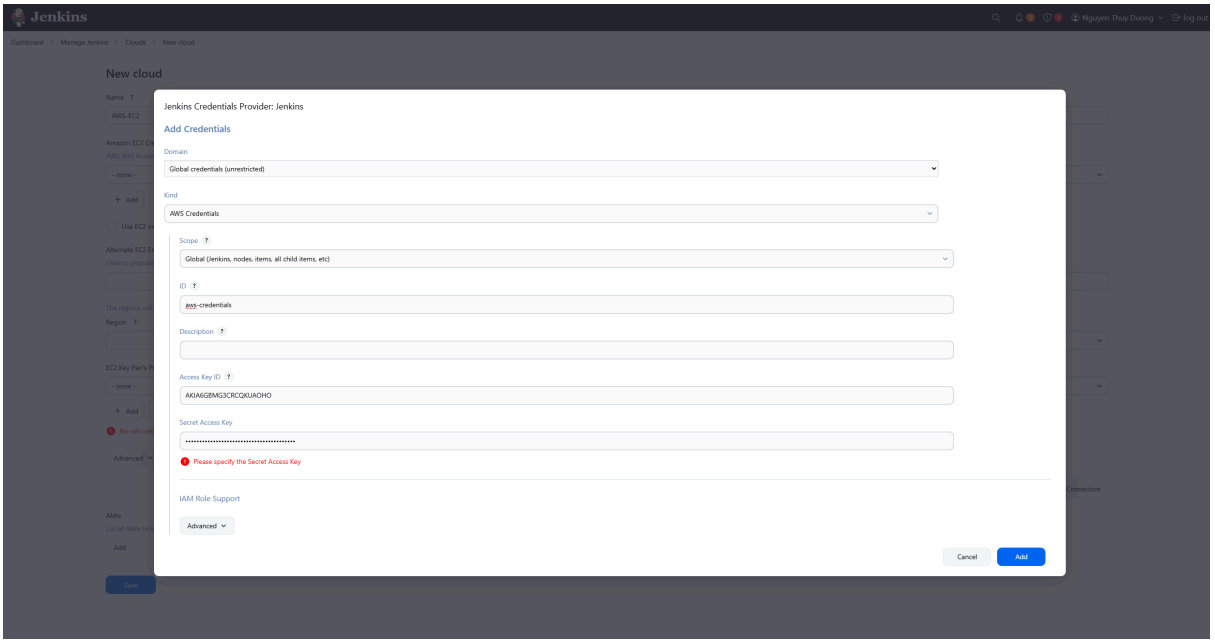


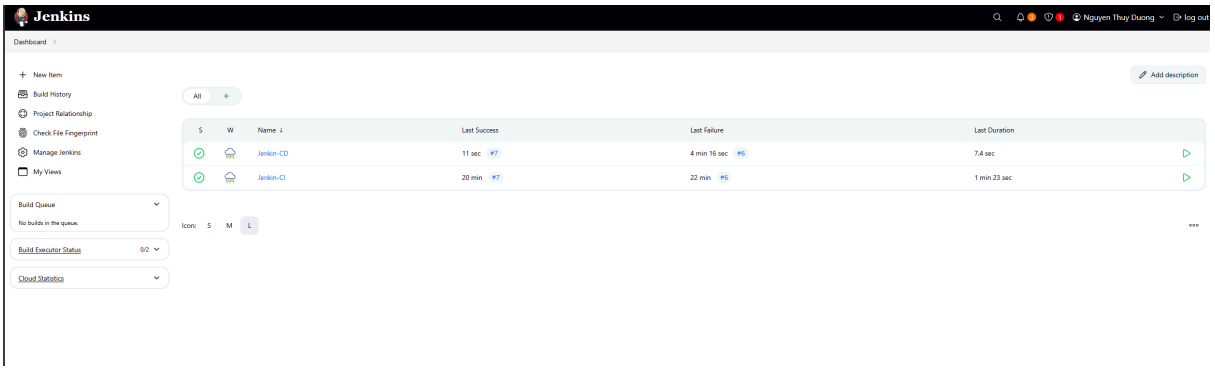
Create Jenkin-CD pipeline



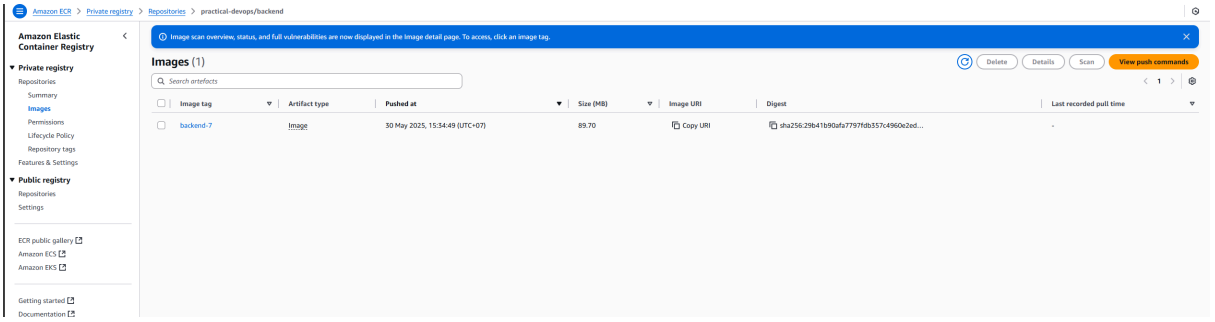


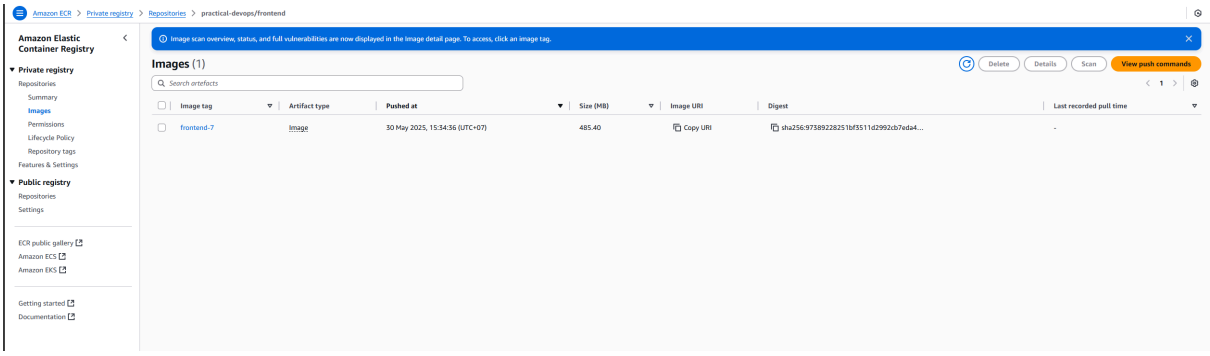
Add credentials

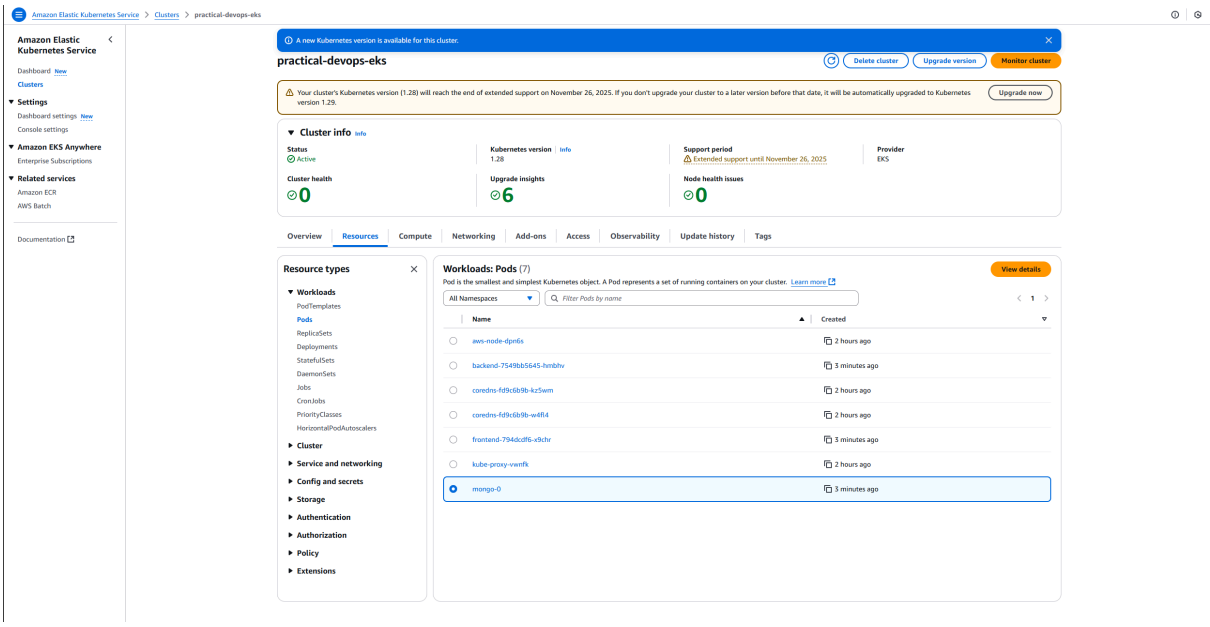




Verify image and resource in ECR and EKS







Check the up services

```
[ec2-user@ip-10-0-0-134 ~]$ kubectl get services
NAME          TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
backend       ClusterIP   172.20.150.42   <none>        3000/TCP   18m
frontend      ClusterIP   172.20.73.129   <none>        3000/TCP   18m
kubernetes    ClusterIP   172.20.0.1      <none>        443/TCP    114m
mongo         ClusterIP   172.20.17.174   <none>        27017/TCP  18m
```

Change the frontend.yaml file to use LoadBalancer, we can see the external ip

```
[ec2-user@ip-10-0-0-134 ~]$ kubectl get services
NAME          TYPE           CLUSTER-IP      EXTERNAL-IP                                                                  PORT(S)          AGE
backend       ClusterIP      172.20.150.42   <none>                                                                       3000/TCP         32m
frontend      LoadBalancer   172.20.73.129   a25584293390b496ca656e6cf309fcfe-26214298.ap-southeast-1.elb.amazonaws.com   3000:32291/TCP   32m
kubernetes    ClusterIP      172.20.0.1      <none>                                                                       443/TCP          128m
mongo         ClusterIP      172.20.17.174   <none>                                                                       27017/TCP        32m
[ec2-user@ip-10-0-0-134 ~]$
```

5 - Monitoring by Prometheus and Grafana

Install helm

```
[ec2-user@ip-10-0-0-134 ~]$ curl -fsSL -o get_helm.sh https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3
[ec2-user@ip-10-0-0-134 ~]$ chmod 700 get_helm.sh
[ec2-user@ip-10-0-0-134 ~]$ ./get_helm.sh
Downloading https://get.helm.sh/helm-v3.18.1-linux-amd64.tar.gz
Verifying checksum... Done.
Preparing to install helm into /usr/local/bin
helm installed into /usr/local/bin/helm
[ec2-user@ip-10-0-0-134 ~]$ helm version
version.BuildInfo{Version:"v3.18.1", GitCommit:"f6f8700a539c18101509434f3b59e6a21402a1b2", GitTreeState:"clean", GoVersion:"go1.24.3"}
[ec2-user@ip-10-0-0-134 ~]$
```

Install prometheus

```
[ec2-user@ip-10-0-0-134 ~]$ helm repo add prometheus-community https://prometheus-community.github.io/helm-charts
"prometheus-community" has been added to your repositories
[ec2-user@ip-10-0-0-134 ~]$ helm repo add grafana https://grafana.github.io/helm-charts
"grafana" has been added to your repositories
[ec2-user@ip-10-0-0-134 ~]$ helm repo update
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "grafana" chart repository
...Successfully got an update from the "prometheus-community" chart repository
Update Complete. ⎈Happy Helming!⎈
```

```
[ec2-user@ip-10-0-0-134 ~]$ helm install prometheus prometheus-community/prometheus \
>   --namespace monitoring --create-namespace
NAME: prometheus
LAST DEPLOYED: Fri May 30 09:39:41 2025
NAMESPACE: monitoring
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.monitoring.svc.cluster.local


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


Get the Alertmanager URL by running these commands in the same shell:
  export POD_NAME=$(kubectl get pods --namespace monitoring -l "app.kubernetes.io/name=alertmanager,app.kubernetes.io/instance=prometheus" -o jsonpath="{.items[0].metadata.name}")
  kubectl --namespace monitoring 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.monitoring.svc.cluster.local


Get the PushGateway URL by running these commands in the same shell:
  export POD_NAME=$(kubectl get pods --namespace monitoring -l "app=prometheus-pushgateway,component=pushgateway" -o jsonpath="{.items[0].metadata.name}")
  kubectl --namespace monitoring port-forward $POD_NAME 9091

For more information on running Prometheus, visit:
https://prometheus.io/
[ec2-user@ip-10-0-0-134 ~]$ kubectl get svc -n monitoring
NAME                                  TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)    AGE
prometheus-alertmanager               ClusterIP   172.20.227.140   <none>        9093/TCP   81s
prometheus-alertmanager-headless      ClusterIP   None             <none>        9093/TCP   81s
prometheus-kube-state-metrics         ClusterIP   172.20.126.96    <none>        8080/TCP   81s
prometheus-prometheus-node-exporter   ClusterIP   172.20.124.2     <none>        9100/TCP   81s
prometheus-prometheus-pushgateway     ClusterIP   172.20.90.85     <none>        9091/TCP   81s
prometheus-server                     ClusterIP   172.20.17.195    <none>        80/TCP     81s
[ec2-user@ip-10-0-0-134 ~]$
```

Install grafana

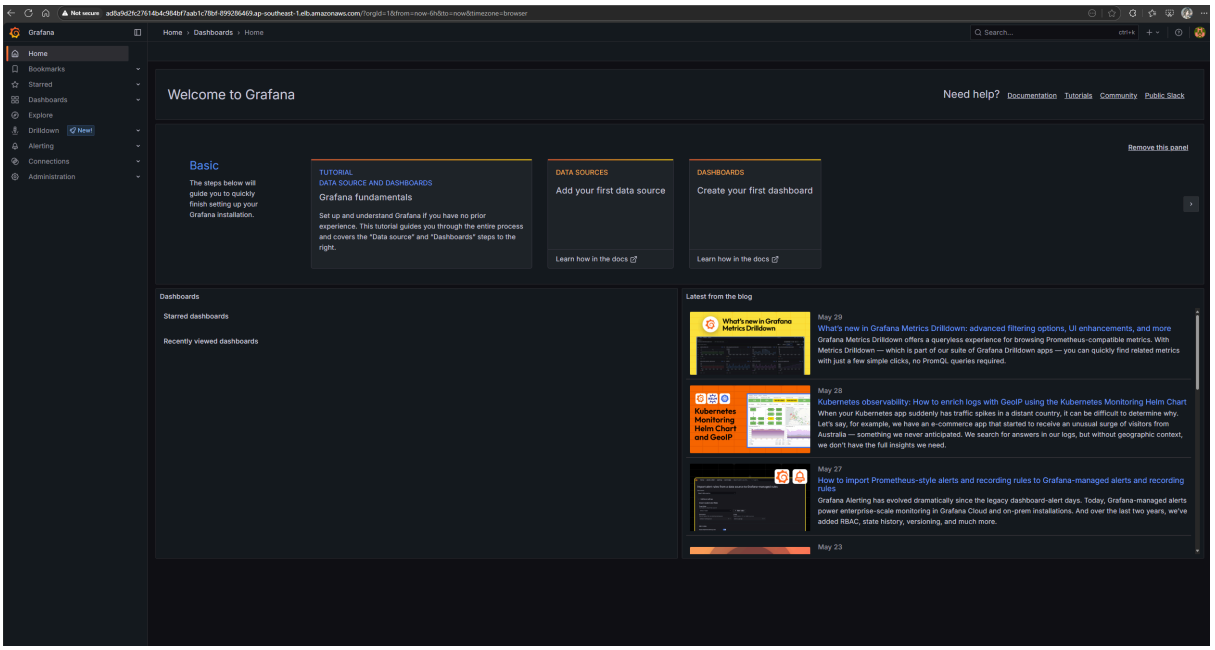
```
[ec2-user@ip-10-0-0-134 ~]$ helm install grafana grafana/grafana \
>   --namespace monitoring
NAME: grafana
LAST DEPLOYED: Fri May 30 09:41:36 2025
NAMESPACE: monitoring
STATUS: deployed
REVISION: 1
NOTES:
1. Get your 'admin' user password by running:

   kubectl get secret --namespace monitoring 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.monitoring.svc.cluster.local

   Get the Grafana URL to visit by running these commands in the same shell:
     export POD_NAME=$(kubectl get pods --namespace monitoring -l "app.kubernetes.io/name=grafana,app.kubernetes.io/instance=grafana" -o jsonpath="{.items[0].metadata.name}")
     kubectl --namespace monitoring 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.                             #####
#################################################################################
[ec2-user@ip-10-0-0-134 ~]$ kubectl get svc -n monitoring
NAME                                  TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)    AGE
grafana                               ClusterIP   172.20.148.85    <none>        80/TCP     8s
prometheus-alertmanager               ClusterIP   172.20.227.140   <none>        9093/TCP   2m2s
prometheus-alertmanager-headless      ClusterIP   None             <none>        9093/TCP   2m2s
prometheus-kube-state-metrics         ClusterIP   172.20.126.96    <none>        8080/TCP   2m2s
prometheus-prometheus-node-exporter   ClusterIP   172.20.124.2     <none>        9100/TCP   2m2s
prometheus-prometheus-pushgateway     ClusterIP   172.20.90.85     <none>        9091/TCP   2m2s
prometheus-server                     ClusterIP   172.20.17.195    <none>        80/TCP     2m2s
[ec2-user@ip-10-0-0-134 ~]$ kubectl patch svc grafana -n monitoring -p '{"spec": {"type": "LoadBalancer"}}'
service/grafana patched
[ec2-user@ip-10-0-0-134 ~]$ kubectl get svc -n monitoring
NAME                                  TYPE           CLUSTER-IP       EXTERNAL-IP                                                              PORT(S)        AGE
grafana                               LoadBalancer   172.20.148.85    ad8a9d2fc27614b4c984bf7aab1c78bf-899286469.ap-southeast-1.elb.amazonaws.com   80:32213/TCP   70s
prometheus-alertmanager               ClusterIP      172.20.227.140   <none>                                                                   9093/TCP       3m4s
prometheus-alertmanager-headless      ClusterIP      None             <none>                                                                   9093/TCP       3m4s
prometheus-kube-state-metrics         ClusterIP      172.20.126.96    <none>                                                                   8080/TCP       3m4s
prometheus-prometheus-node-exporter   ClusterIP      172.20.124.2     <none>                                                                   9100/TCP       3m4s
prometheus-prometheus-pushgateway     ClusterIP      172.20.90.85     <none>                                                                   9091/TCP       3m4s
prometheus-server                     ClusterIP      172.20.17.195    <none>                                                                   80/TCP         3m4s
[ec2-user@ip-10-0-0-134 ~]$
```

Open in browser

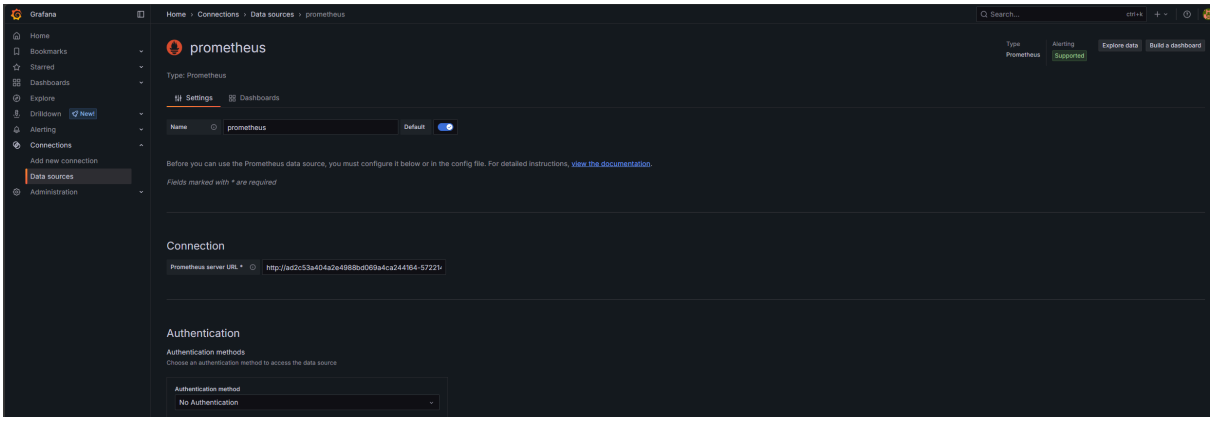
username: admin

password: lLow72xObHUYYdgkodUttTx921HgNGPdDimUgm9H

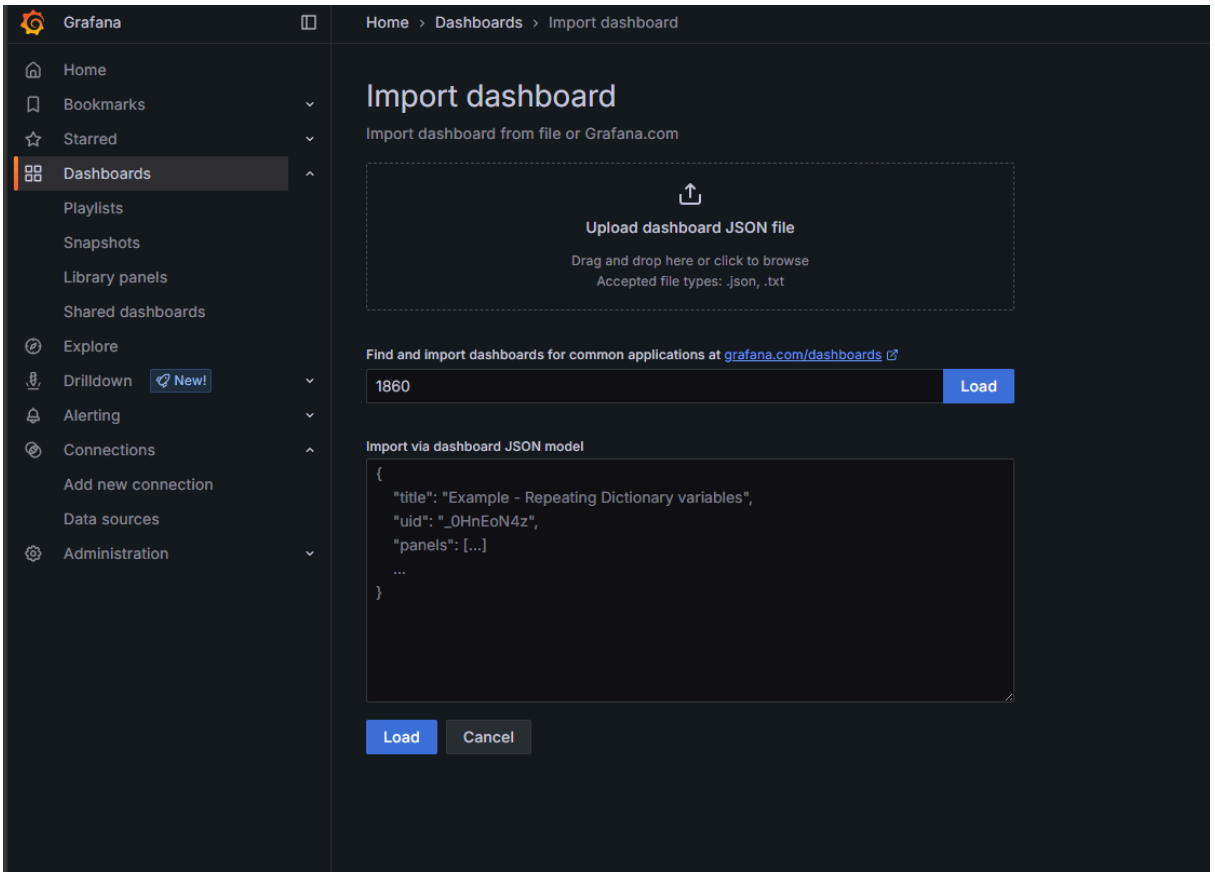
(Get from `kubectl get secret --namespace monitoring grafana -o jsonpath="{.data.admin-password}" | base64 --decode`)

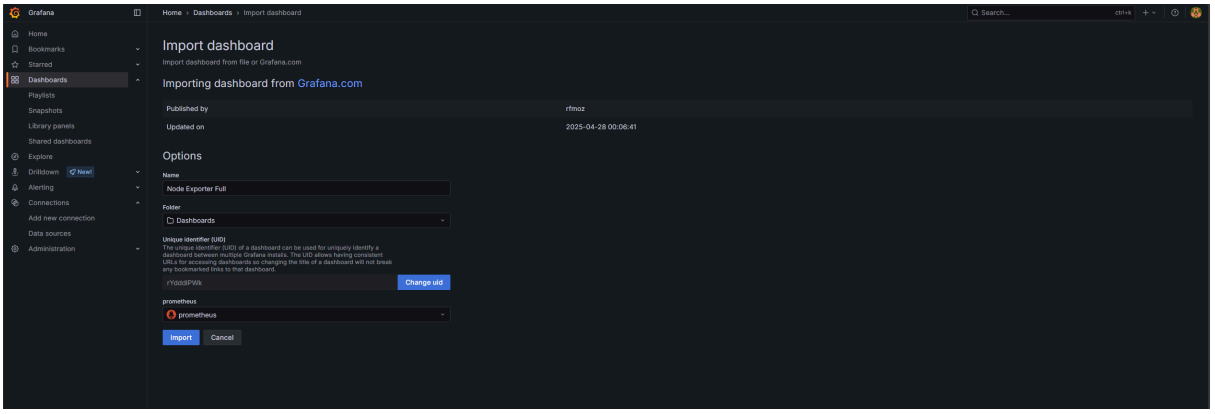


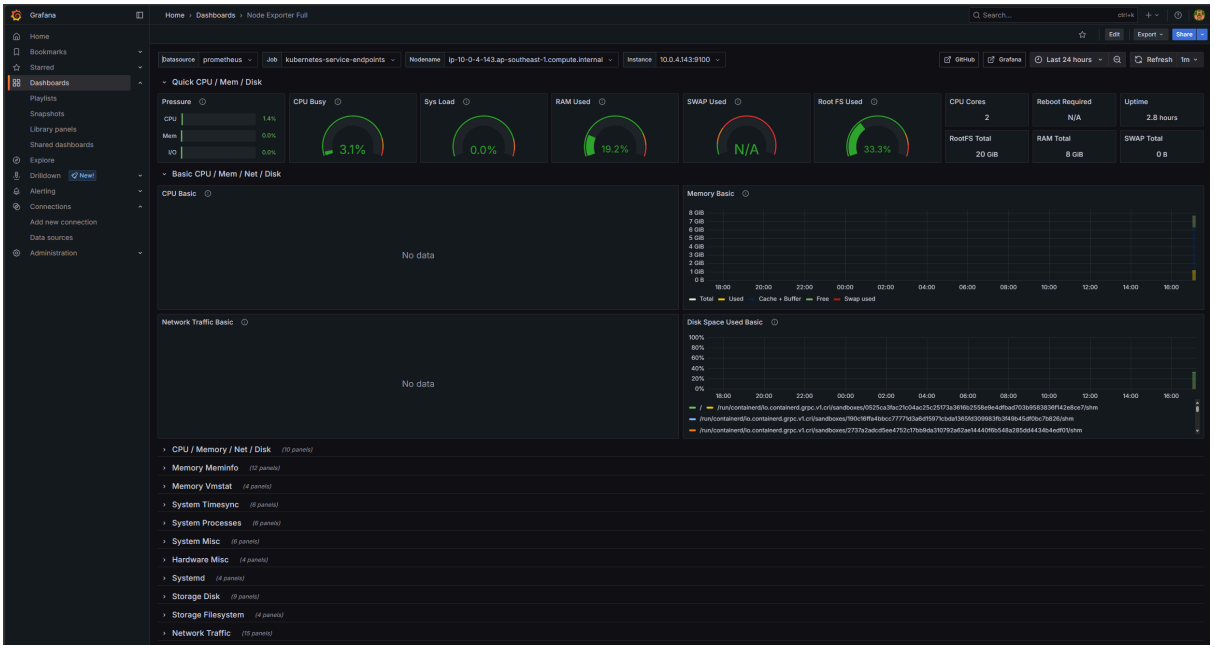
Add data source



Import dashboard




Grafana
Home › Dashboards › Import dashboard
Search...
Home
Bookmarks
Starred
Dashboards
Playlists
Snapshots
Library panels
Shared dashboards
Explore
Drilldown
New!
Alerting
Connections
Add new connection
Data sources
Administration
Import dashboard
Import dashboard from file or Grafana.com
Importing dashboard from Grafana.com
Published by
rfmoz
Updated on
2025-04-28 00:06:41
Options
Name
Node Exporter Full
Folder
Dashboards
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.
rYdddlPWk
Change uid
prometheus
prometheus
Import
Cancel


Grafana
Home › Dashboards › Node Exporter Full
Search...
Edit
Export
Share
Datasource
prometheus
Job
kubernetes-service-endpoints
Nodename
ip-10-0-4-143.ap-southeast-1.compute.internal
Instance
10.0.4.143:9100
GitHub
Grafana
Last 24 hours
Refresh
1m
Quick CPU / Mem / Disk
Pressure
CPU
1.4%
Mem
0.0%
I/O
0.0%
CPU Busy
3.1%
Sys Load
0.0%
RAM Used
19.2%
SWAP Used
N/A
Root FS Used
33.3%
CPU Cores
2
Reboot Required
N/A
Uptime
2.8 hours
RootFS Total
20 GiB
RAM Total
8 GiB
SWAP Total
0 B
Basic CPU / Mem / Net / Disk
CPU Basic
No data
Memory Basic
Total
Used
Cache + Buffer
Free
Swap used
Network Traffic Basic
No data
Disk Space Used Basic
CPU / Memory / Net / Disk (10 panels)
Memory Meminfo (12 panels)
Memory Vmstat (4 panels)
System Timesync (8 panels)
System Processes (8 panels)
System Misc (6 panels)
Hardware Misc (4 panels)
Systemd (4 panels)
Storage Disk (9 panels)
Storage Filesystem (4 panels)
Network Traffic (15 panels)