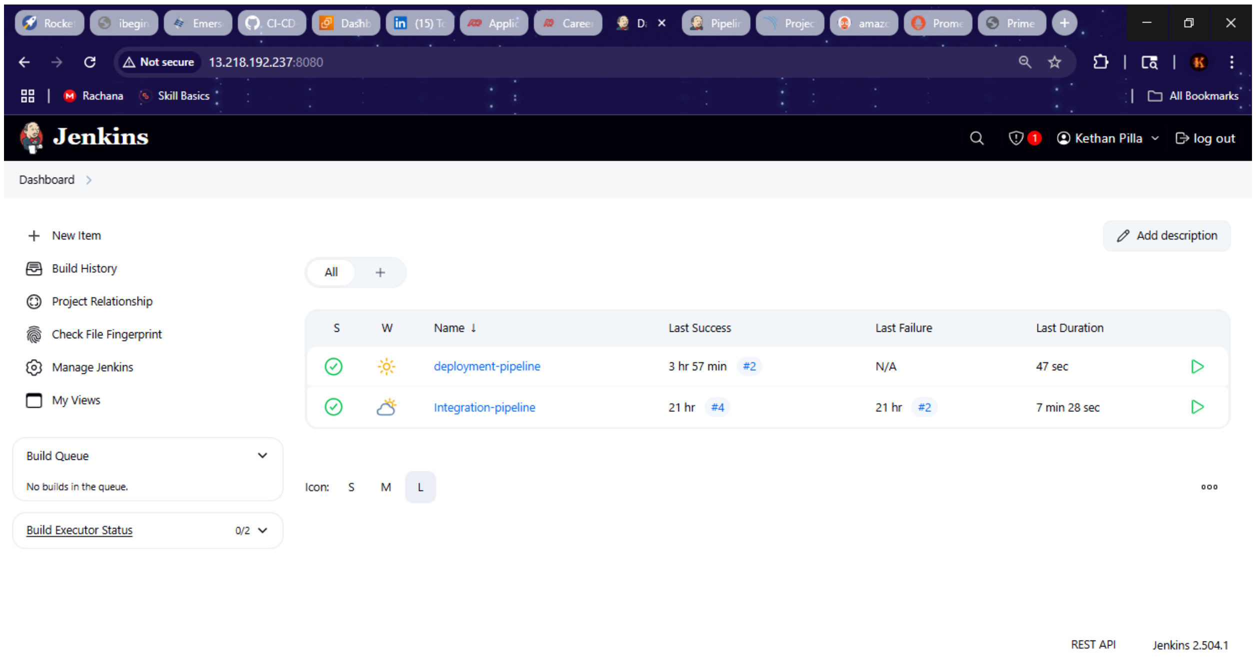
Jenkins Dashboard



S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	☀	deployment-pipeline	3 hr 57 min #2	N/A	47 sec
✓	⛅	Integration-pipeline	21 hr #4	21 hr #2	7 min 28 sec

Integration pipeline script

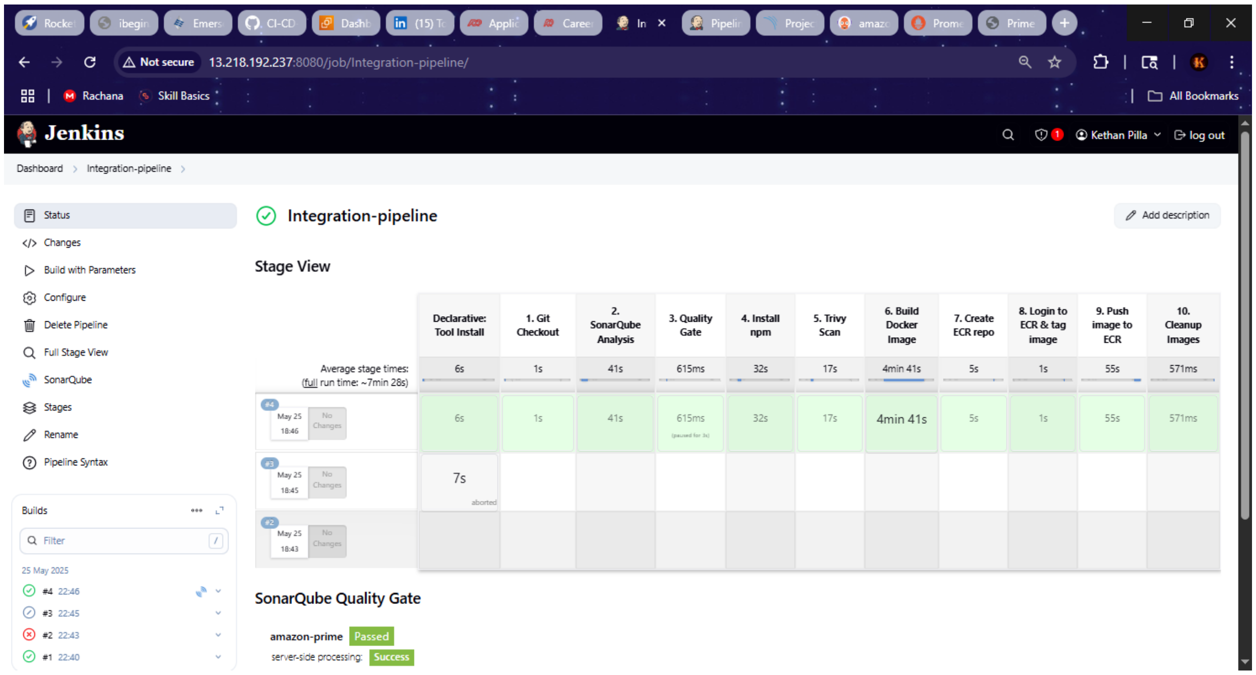
```
pipeline {
  agent any


  parameters {
    string(name: 'ECR_REPO_NAME', defaultValue: 'amazon-prime', description: 'Enter repository name')
    string(name: 'AWS_ACCOUNT_ID', defaultValue: '123456789012', description: 'Enter AWS Account ID') // Added missing quote
  }


  tools {
    jdk 'JDK'
  nodejs 'NodeJS'
}

environment {
  SCANNER_HOME = tool 'SonarQube Scanner'
}

stages {
  stage('1. Git Checkout') {
    steps {
      git branch: 'main', url: 'https://github.com/pandacloud1/DevopsProject2.git'
    }
  }

  stage('2. SonarQube Analysis') {
    steps {
      withSonarQubeEnv ('sonar-server') {
        sh """
        $SCANNER_HOME/bin/sonar-scanner \
        -Dsonar.projectName=amazon-prime \
        -Dsonar.projectKey=amazon-prime
        """
      }
    }
  }
```

```
stage('3. Quality Gate') {
  steps {
    waitForQualityGate abortPipeline: false,
    credentialsId: 'sonar-token'
  }
}

stage('4. Install npm') {
  steps {
    sh "npm install"
  }
}

stage('5. Trivy Scan') {
  steps {
    sh "trivy fs . > trivy.txt"
  }
}

stage('6. Build Docker Image') {
  steps {
    sh "docker build -t ${params.ECR_REPO_NAME} ."
  }
}

stage('7. Create ECR repo') {
        steps {
          withCredentials([string(credentialsId: 'access-key', variable: 'AWS_ACCESS_KEY'),
                          string(credentialsId: 'secret-key', variable: 'AWS_SECRET_KEY')]) {
            sh """
            aws configure set aws_access_key_id $AWS_ACCESS_KEY
            aws configure set aws_secret_access_key $AWS_SECRET_KEY
            aws ecr describe-repositories --repository-names ${params.ECR_REPO_NAME} --
region us-east-1 || \
            aws ecr create-repository --repository-name ${params.ECR_REPO_NAME} --
region us-east-1
            """
          }
        }
      }

      stage('8. Login to ECR & tag image') {
        steps {
          withCredentials([string(credentialsId: 'access-key', variable: 'AWS_ACCESS_KEY'),
                          string(credentialsId: 'secret-key', variable: 'AWS_SECRET_KEY')]) {
            sh """
            aws ecr get-login-password --region us-east-1 | docker login --username AWS --
password-stdin ${params.AWS_ACCOUNT_ID}.dkr.ecr.us-east-1.amazonaws.com
            docker tag ${params.ECR_REPO_NAME}
${params.AWS_ACCOUNT_ID}.dkr.ecr.us-east-
1.amazonaws.com/${params.ECR_REPO_NAME}:${BUILD_NUMBER}
            docker tag ${params.ECR_REPO_NAME}
${params.AWS_ACCOUNT_ID}.dkr.ecr.us-east-
1.amazonaws.com/${params.ECR_REPO_NAME}:latest
```

```
          """
        }
      }
    }

    stage('9. Push image to ECR') {
      steps {
        withCredentials([string(credentialsId: 'access-key', variable: 'AWS_ACCESS_KEY'),
                 string(credentialsId: 'secret-key', variable: 'AWS_SECRET_KEY')]) {
          sh """
          docker push ${params.AWS_ACCOUNT_ID}.dkr.ecr.us-east-
1.amazonaws.com/${params.ECR_REPO_NAME}:${BUILD_NUMBER}
          docker push ${params.AWS_ACCOUNT_ID}.dkr.ecr.us-east-
1.amazonaws.com/${params.ECR_REPO_NAME}:latest
          """
        }
      }
    }

    stage('10. Cleanup Images') {
      steps {
        sh """
        docker rmi ${params.AWS_ACCOUNT_ID}.dkr.ecr.us-east-
1.amazonaws.com/${params.ECR_REPO_NAME}:${BUILD_NUMBER}
        docker rmi ${params.AWS_ACCOUNT_ID}.dkr.ecr.us-east-
1.amazonaws.com/${params.ECR_REPO_NAME}:latest
             docker images
```

```
                """
            }
        }
    }
}
```

Integration pipeline



Integration-pipeline

Stage View

	Declarative: Tool Install	1. Git Checkout	2. SonarQube Analysis	3. Quality Gate	4. Install npm	5. Trivy Scan	6. Build Docker Image	7. Create ECR repo	8. Login to ECR & tag image	9. Push image to ECR	10. Cleanup Images
Average stage times: (full run time: ~7min 28s)	6s	1s	41s	615ms	32s	17s	4min 41s	5s	1s	55s	571ms
#4 May 25 18:46 No Changes	6s	1s	41s	615ms (paused for 3s)	32s	17s	4min 41s	5s	1s	55s	571ms
#3 May 25 18:45 No Changes	7s aborted										
#2 May 25 18:43 No Changes											

25 May 2025
- #4 22:46
- #3 22:45
- #2 22:43
- #1 22:40

SonarQube Quality Gate

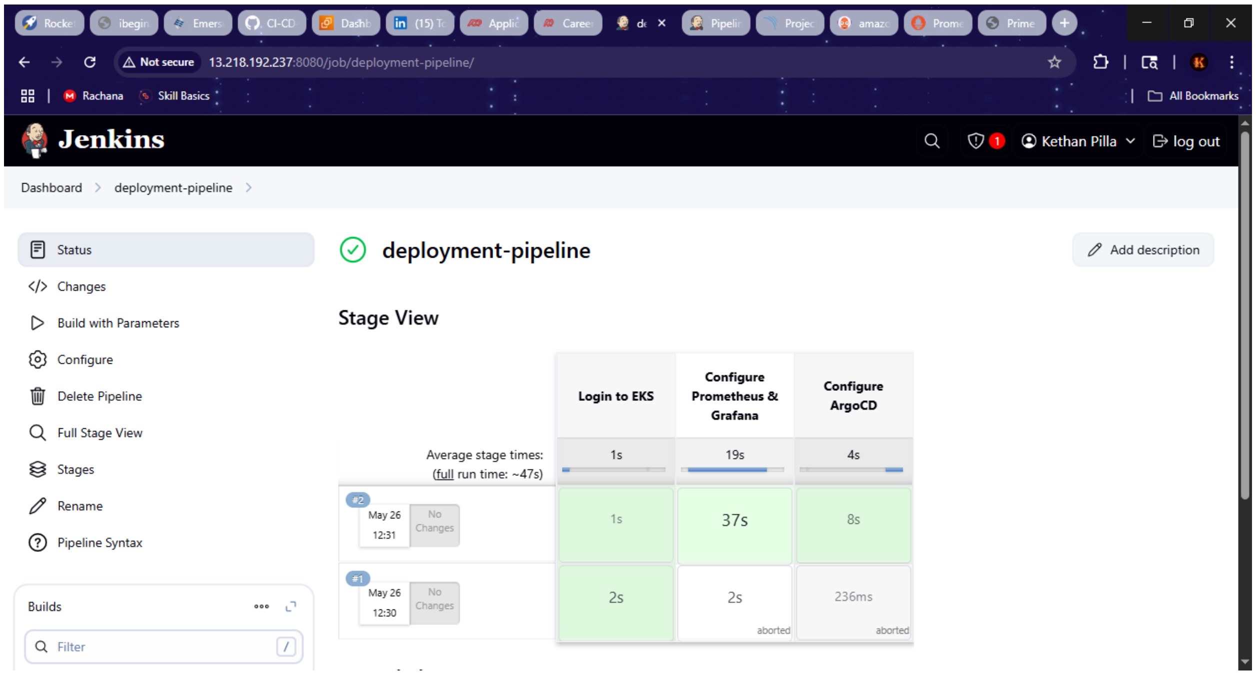
amazon-prime Passed
server-side processing: Success

Deployment pipeline script

```
pipeline {
  agent any

  environment {
    KUBECTL = '/usr/local/bin/kubectl'
  }
  parameters {
    string(name: 'CLUSTER_NAME', defaultValue: 'amazon-prime-cluster', description:
'Enter your EKS cluster name')
  }

  stages {
    stage("Login to EKS") {
      steps {
        script {
          withCredentials([string(credentialsId: 'access-key', variable: 'AWS_ACCESS_KEY'),
                   string(credentialsId: 'secret-key', variable: 'AWS_SECRET_KEY')]) {
            sh "aws eks --region us-east-1 update-kubeconfig --name
${params.CLUSTER_NAME}"
          }
        }
      }
    }

    stage("Configure Prometheus & Grafana") {
      steps {
        script {
          sh """
          helm repo add stable https://charts.helm.sh/stable || true
          helm repo add prometheus-community https://prometheus-
community.github.io/helm-charts || true
          # Check if namespace 'prometheus' exists
          if kubectl get namespace prometheus > /dev/null 2>&1; then
```

```
                    # If namespace exists, upgrade the Helm release
                    helm upgrade stable prometheus-community/kube-prometheus-stack -n
prometheus
                else
                    # If namespace does not exist, create it and install Helm release
                    kubectl create namespace prometheus
                    helm install stable prometheus-community/kube-prometheus-stack -n
prometheus
                fi
                kubectl patch svc stable-kube-prometheus-sta-prometheus -n prometheus -p
'{"spec": {"type": "LoadBalancer"}}'
                kubectl patch svc stable-grafana -n prometheus -p '{"spec": {"type":
"LoadBalancer"}}'
                """
            }
        }
    }

    stage("Configure ArgoCD") {
        steps {
            script {
                sh """
                # Install ArgoCD
                kubectl create namespace argocd || true
                kubectl apply -n argocd -f https://raw.githubusercontent.com/argoproj/argo-
cd/stable/manifests/install.yaml
                kubectl patch svc argocd-server -n argocd -p '{"spec": {"type": "LoadBalancer"}}'
                """
            }
          }
        }


      }
    }
```

Jenkins Deployment pipeline



13.218.192.237:8080/job/deployment-pipeline/

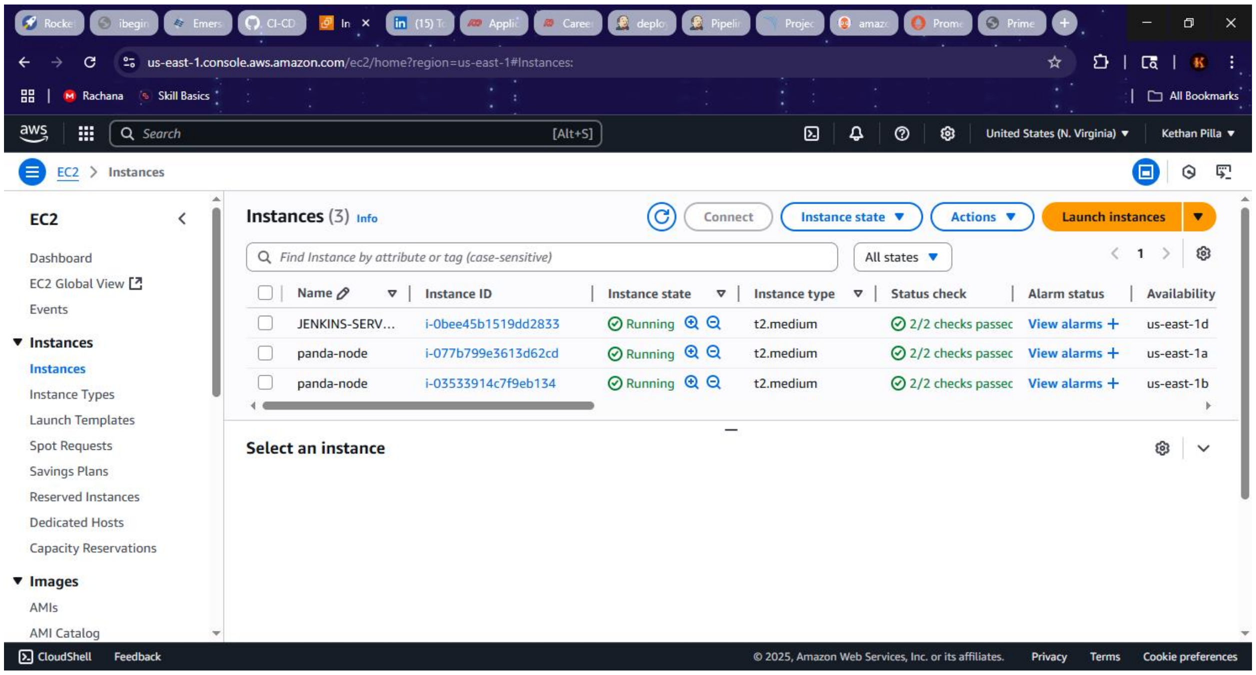
Dashboard > deployment-pipeline >

deployment-pipeline

Stage View

	Login to EKS	Configure Prometheus & Grafana	Configure ArgoCD
Average stage times: (full run time: ~47s)	1s	19s	4s
#2 May 26 12:31 No Changes	1s	37s	8s
#1 May 26 12:30 No Changes	2s	2s aborted	236ms aborted

AWS EC2 instances



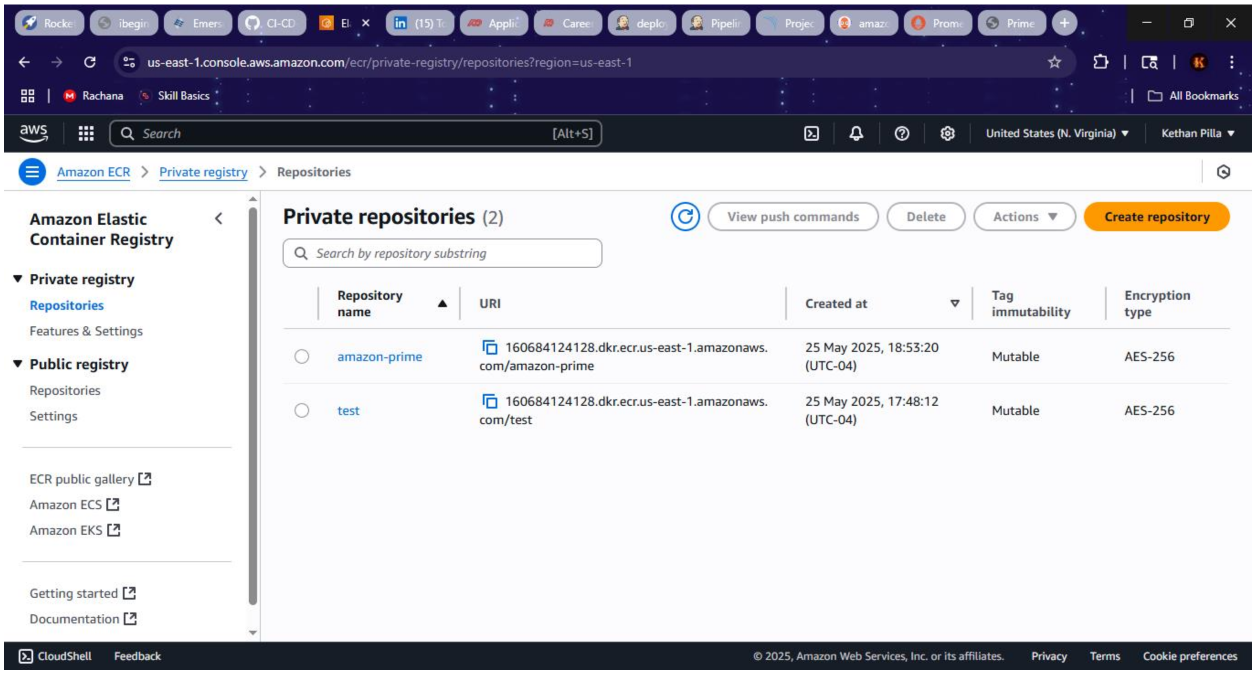
us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:

Instances (3)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
JENKINS-SERV...	i-0bee45b1519dd2833	Running	t2.medium	2/2 checks passec	View alarms +	us-east-1d
panda-node	i-077b799e3613d62cd	Running	t2.medium	2/2 checks passec	View alarms +	us-east-1a
panda-node	i-03533914c7f9eb134	Running	t2.medium	2/2 checks passec	View alarms +	us-east-1b

Select an instance

Repositories

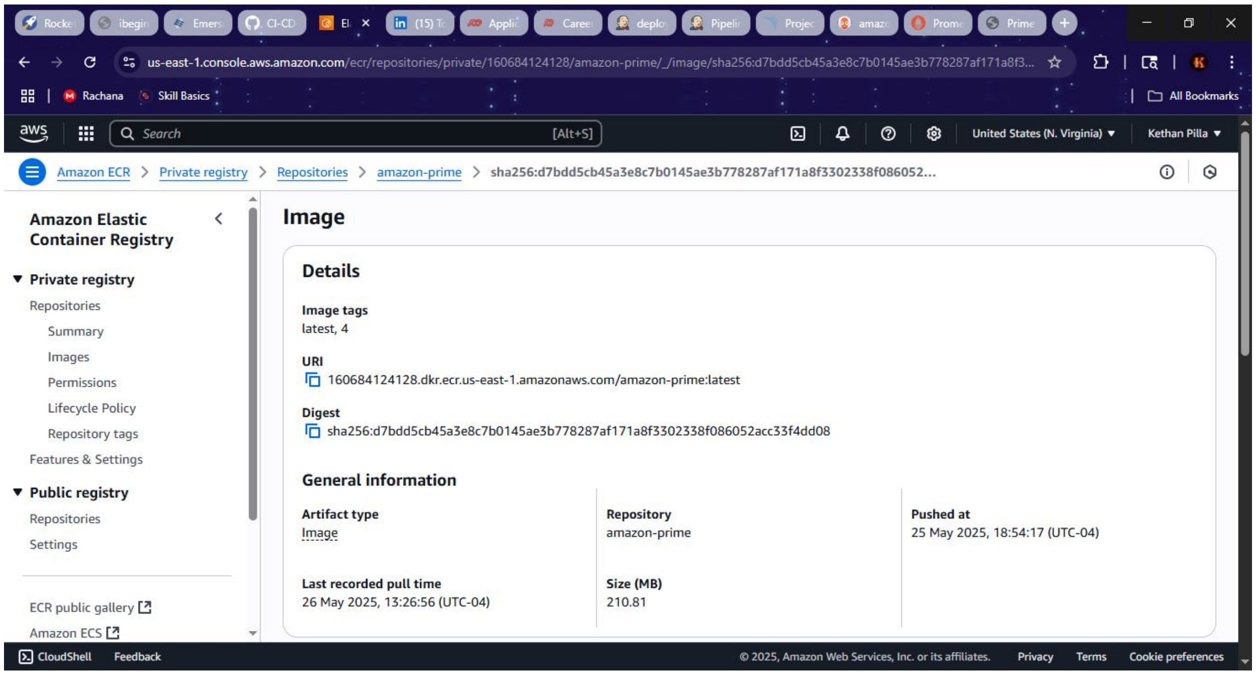


us-east-1.console.aws.amazon.com/ecr/private-registry/repositories?region=us-east-1

Private repositories (2)

Repository name	URI	Created at	Tag immutability	Encryption type
amazon-prime	160684124128.dkr.ecr.us-east-1.amazonaws.com/amazon-prime	25 May 2025, 18:53:20 (UTC-04)	Mutable	AES-256
test	160684124128.dkr.ecr.us-east-1.amazonaws.com/test	25 May 2025, 17:48:12 (UTC-04)	Mutable	AES-256

Amazon Elastic Container Registry (ECR)


us-east-1.console.aws.amazon.com/ecr/repositories/private/160684124128/amazon-prime/_/image/sha256:d7bdd5cb45a3e8c7b0145ae3b778287af171a8f3...

Amazon ECR > Private registry > Repositories > amazon-prime > sha256:d7bdd5cb45a3e8c7b0145ae3b778287af171a8f3302338f086052...

Amazon Elastic Container Registry

Private registry
- Repositories
  - Summary
  - Images
  - Permissions
  - Lifecycle Policy
  - Repository tags
- Features & Settings

Public registry
- Repositories
- Settings

ECR public gallery
Amazon ECS

Image

Details

Image tags
latest, 4

URI
160684124128.dkr.ecr.us-east-1.amazonaws.com/amazon-prime:latest

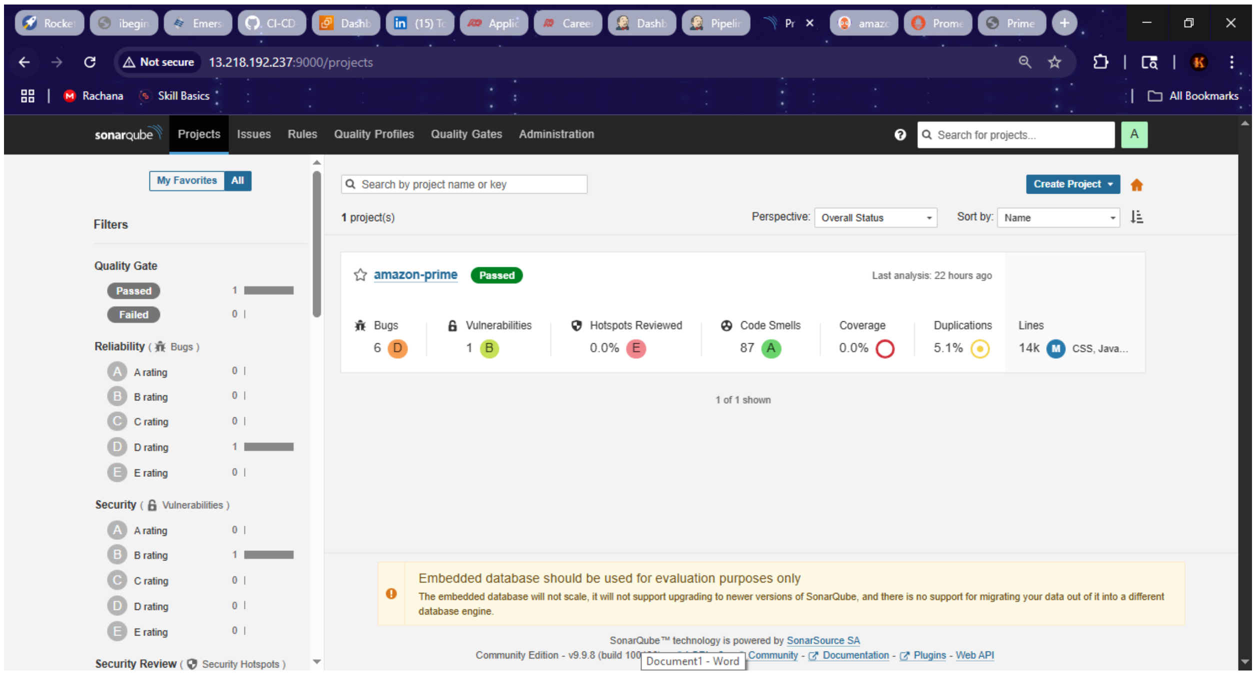
Digest
sha256:d7bdd5cb45a3e8c7b0145ae3b778287af171a8f3302338f086052acc33f4dd08

General information

Artifact type	Repository	Pushed at
Image	amazon-prime	25 May 2025, 18:54:17 (UTC-04)
Last recorded pull time	**Size (MB)**	
26 May 2025, 13:26:56 (UTC-04)	210.81	

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

SonarQube


Not secure 13.218.192.237:9000/projects

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration

My Favorites All

Filters

Quality Gate	
Passed	1
Failed	0

Reliability (Bugs)	
A rating	0
B rating	0
C rating	0
D rating	1
E rating	0

Security (Vulnerabilities)	
A rating	0
B rating	1
C rating	0
D rating	0
E rating	0

Security Review (Security Hotspots)

1 project(s) Perspective: Overall Status Sort by: Name

amazon-prime Passed Last analysis: 22 hours ago

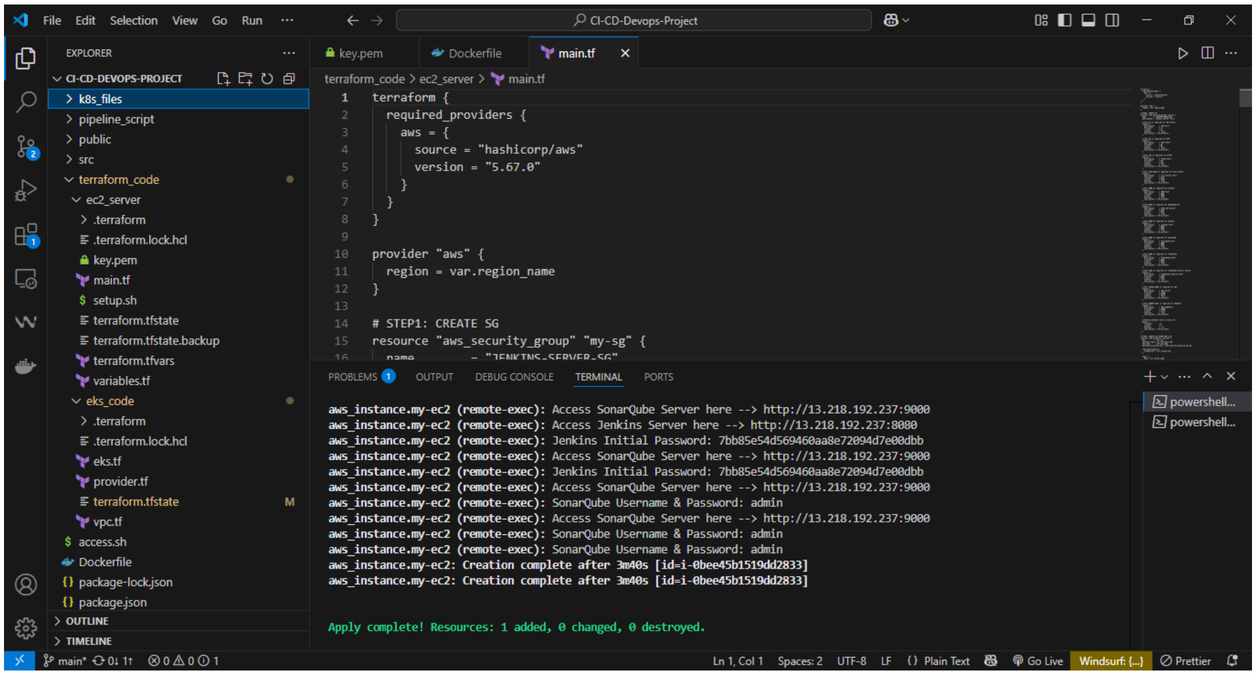
Bugs	Vulnerabilities	Hotspots Reviewed	Code Smells	Coverage	Duplications	Lines
6 D	1 B	0.0% E	87 A	0.0%	5.1%	14k M CSS, Java...

1 of 1 shown

Embedded database should be used for evaluation purposes only
The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.

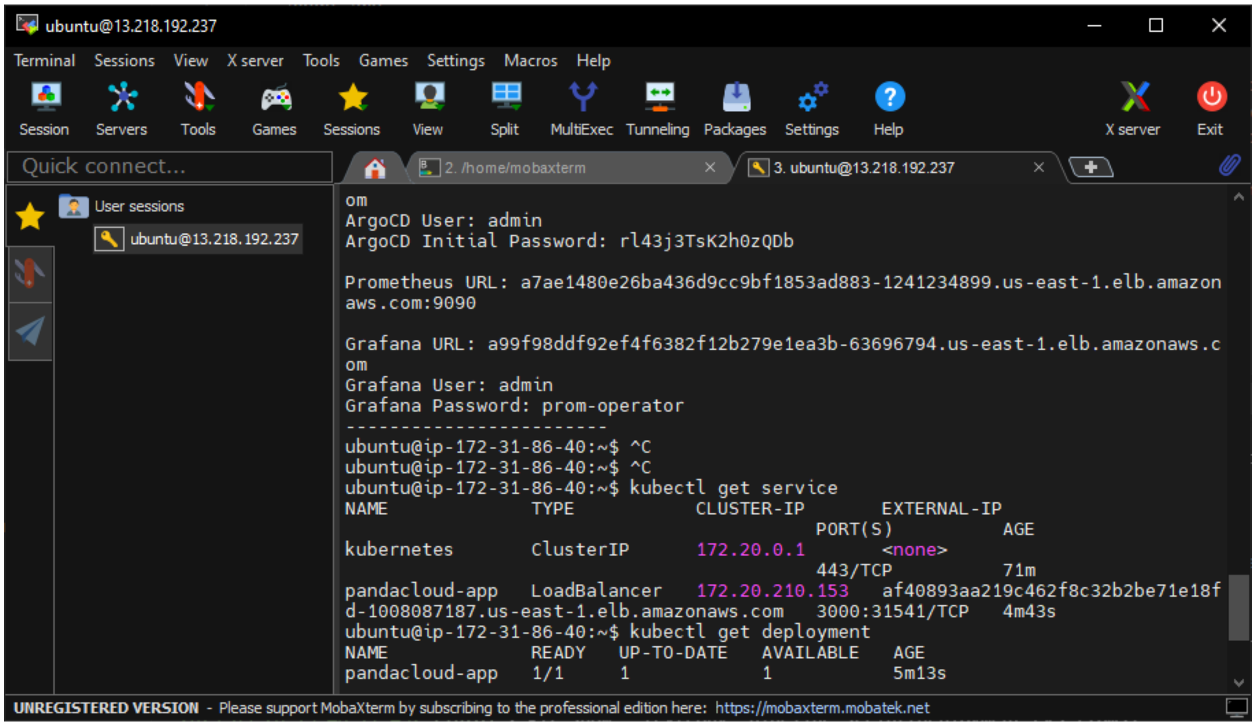
SonarQube™ technology is powered by SonarSource SA
Community Edition - v9.9.8 (build 100 ... Community - Documentation - Plugins - Web API

Terraform



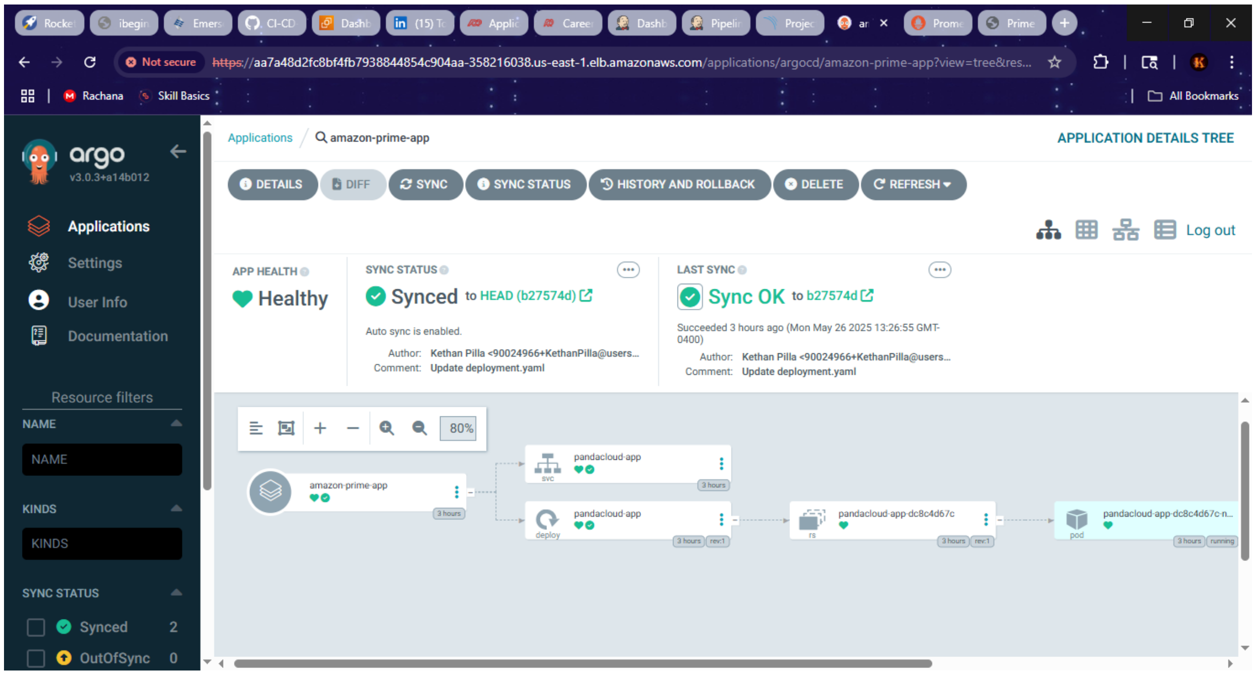
```
terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "5.67.0"
    }
  }
}

provider "aws" {
  region = var.region_name
}

# STEP1: CREATE SG
resource "aws_security_group" "my-sg" {
  name         = "JENKINS-SERVER-SG"
```

```
aws_instance.my-ec2 (remote-exec): Access SonarQube Server here --> http://13.218.192.237:9000
aws_instance.my-ec2 (remote-exec): Access Jenkins Server here --> http://13.218.192.237:8080
aws_instance.my-ec2 (remote-exec): Jenkins Initial Password: 7bb85e54d569460aa8e72094d7e00dbb
aws_instance.my-ec2 (remote-exec): Access SonarQube Server here --> http://13.218.192.237:9000
aws_instance.my-ec2 (remote-exec): Jenkins Initial Password: 7bb85e54d569460aa8e72094d7e00dbb
aws_instance.my-ec2 (remote-exec): Access SonarQube Server here --> http://13.218.192.237:9000
aws_instance.my-ec2 (remote-exec): SonarQube Username & Password: admin
aws_instance.my-ec2 (remote-exec): Access SonarQube Server here --> http://13.218.192.237:9000
aws_instance.my-ec2 (remote-exec): SonarQube Username & Password: admin
aws_instance.my-ec2 (remote-exec): SonarQube Username & Password: admin
aws_instance.my-ec2: Creation complete after 3m40s [id=i-0bee45b1519dd2833]
aws_instance.my-ec2: Creation complete after 3m40s [id=i-0bee45b1519dd2833]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```



```
om
ArgoCD User: admin
ArgoCD Initial Password: rl43j3TsK2h0zQDb

Prometheus URL: a7ae1480e26ba436d9cc9bf1853ad883-1241234899.us-east-1.elb.amazon
aws.com:9090

Grafana URL: a99f98ddf92ef4f6382f12b279e1ea3b-63696794.us-east-1.elb.amazonaws.c
om
Grafana User: admin
Grafana Password: prom-operator
------------------------
ubuntu@ip-172-31-86-40:~$ ^C
ubuntu@ip-172-31-86-40:~$ ^C
ubuntu@ip-172-31-86-40:~$ kubectl get service
NAME              TYPE          CLUSTER-IP      EXTERNAL-IP
                                         PORT(S)            AGE
kubernetes        ClusterIP     172.20.0.1      <none>
                                         443/TCP            71m
pandacloud-app    LoadBalancer  172.20.210.153  af40893aa219c462f8c32b2be71e18f
d-1008087187.us-east-1.elb.amazonaws.com  3000:31541/TCP    4m43s
ubuntu@ip-172-31-86-40:~$ kubectl get deployment
NAME              READY   UP-TO-DATE    AVAILABLE    AGE
pandacloud-app    1/1     1             1            5m13s
```

Argo application

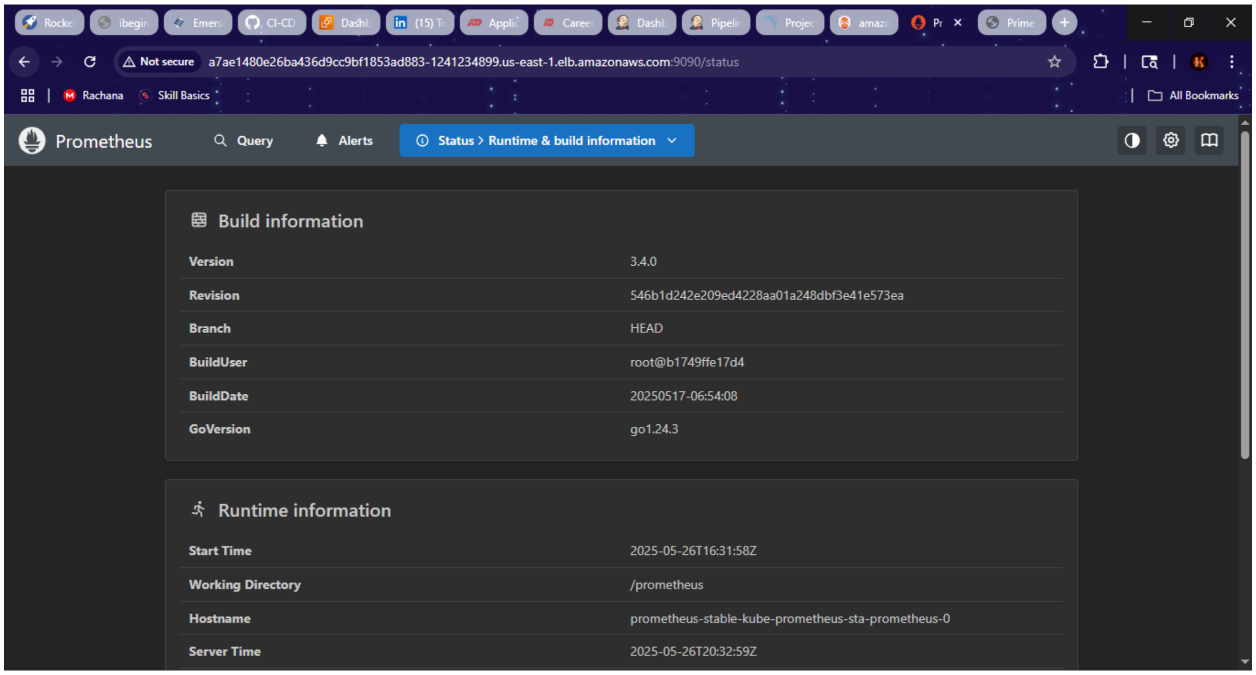


APP HEALTH
Healthy

SYNC STATUS
Synced to HEAD (b27574d)
Auto sync is enabled.
Author: Kethan Pilla <90024966+KethanPilla@users...
Comment: Update deployment.yaml

LAST SYNC
Sync OK to b27574d
Succeeded 3 hours ago (Mon May 26 2025 13:26:55 GMT-0400)
Author: Kethan Pilla <90024966+KethanPilla@users...
Comment: Update deployment.yaml

Prometheus


a7ae1480e26ba436d9cc9bf1853ad883-1241234899.us-east-1.elb.amazonaws.com:9090/status
Prometheus
Query
Alerts
Status > Runtime & build information

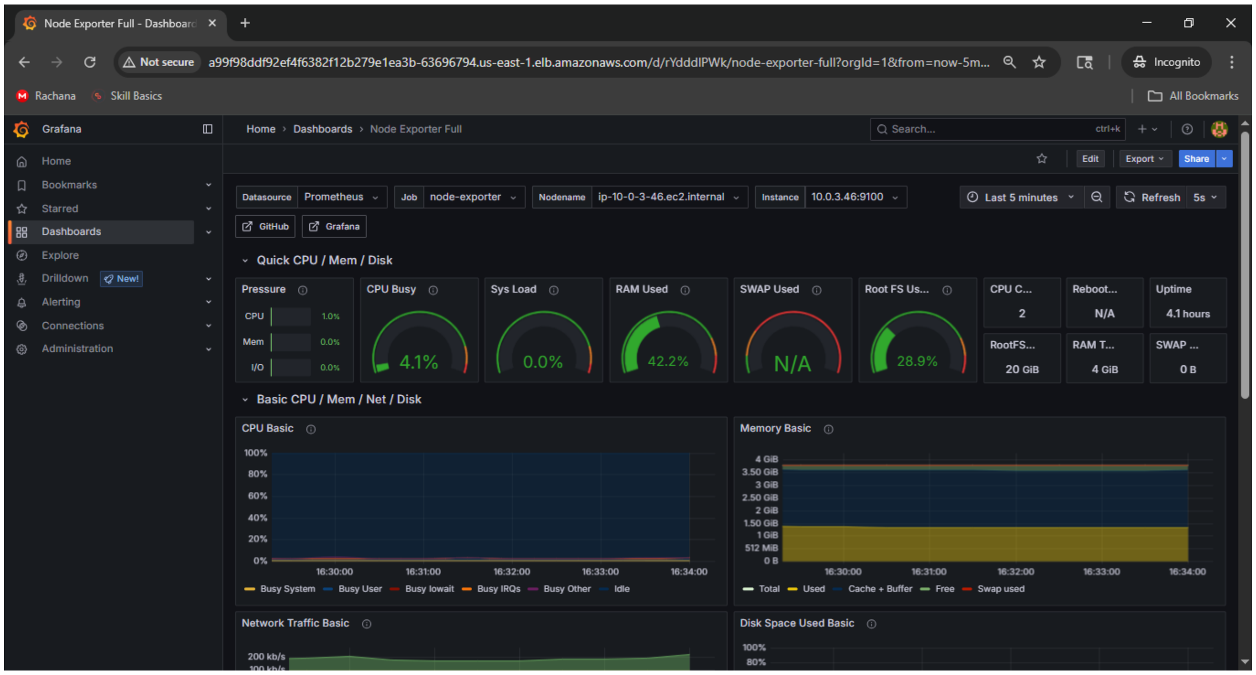
Build information

Version	3.4.0
Revision	546b1d242e209ed4228aa01a248dbf3e41e573ea
Branch	HEAD
BuildUser	root@b1749ffe17d4
BuildDate	20250517-06:54:08
GoVersion	go1.24.3

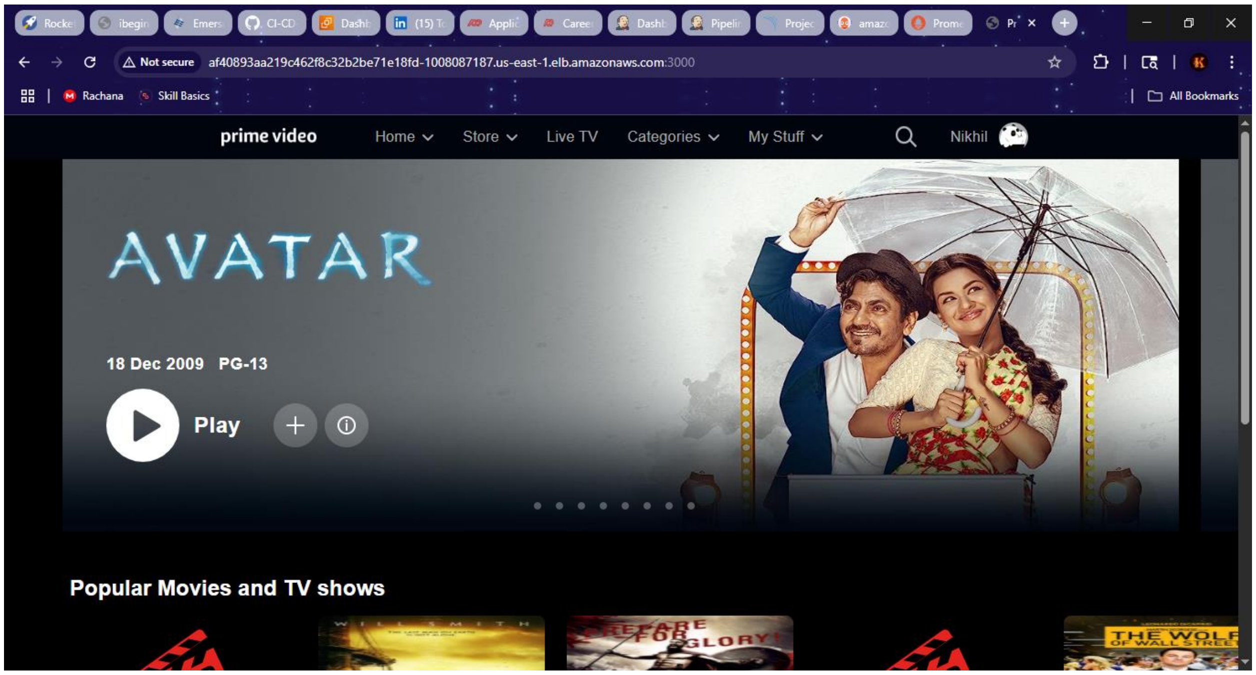
Runtime information

Start Time	2025-05-26T16:31:58Z
Working Directory	/prometheus
Hostname	prometheus-stable-kube-prometheus-sta-prometheus-0
Server Time	2025-05-26T20:32:59Z

Grafana dashboard



Deployed app

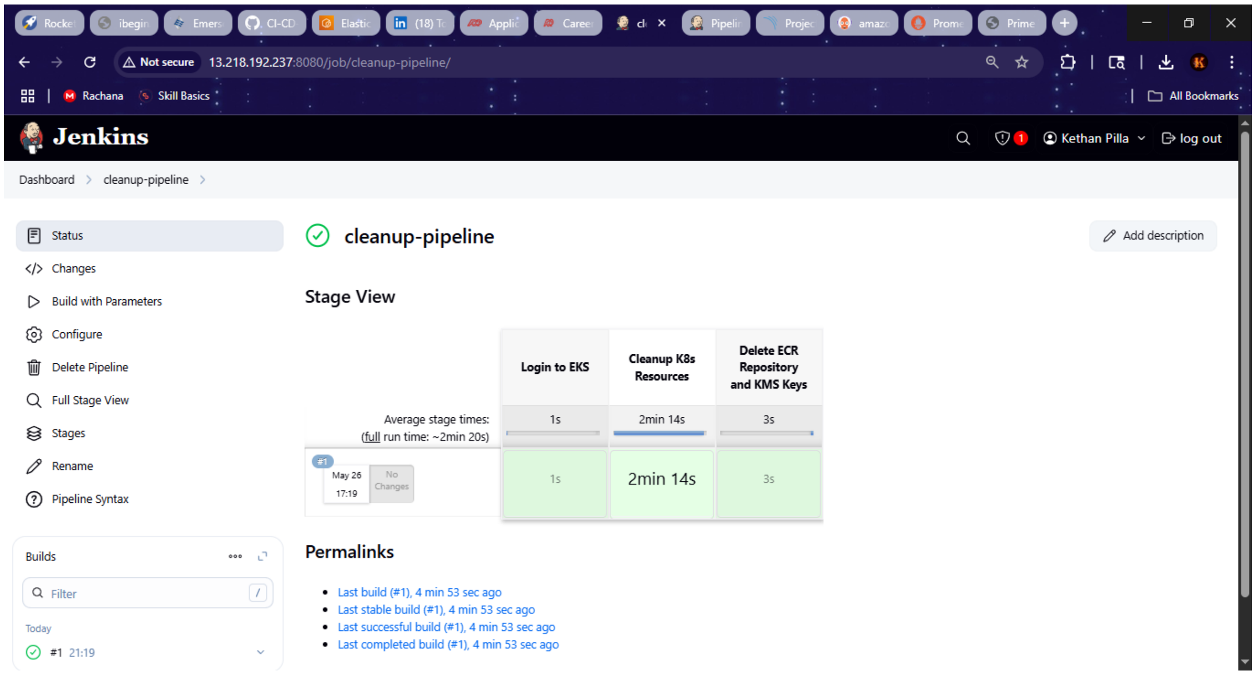

Not secure af40893aa219c462f8c32b2be71e18fd-1008087187.us-east-1.elb.amazonaws.com:3000
prime video
Home
Store
Live TV
Categories
My Stuff
Nikhil
AVATAR
18 Dec 2009 PG-13
Play
Popular Movies and TV shows

Pipeline-cleanup-code

pipeline {

```
  agent any

  environment {
     KUBECTL = '/usr/local/bin/kubectl'
  }

  parameters {
     string(name: 'CLUSTER_NAME', defaultValue: 'amazon-prime-cluster', description:
'Enter your EKS cluster name')
  }

  stages {

     stage("Login to EKS") {
        steps {
           script {
              withCredentials([string(credentialsId: 'access-key', variable: 'AWS_ACCESS_KEY'),
                          string(credentialsId: 'secret-key', variable: 'AWS_SECRET_KEY')]) {
                 sh "aws eks --region us-east-1 update-kubeconfig --name
${params.CLUSTER_NAME}"
              }
           }
        }
     }

     stage('Cleanup K8s Resources') {
        steps {
           script {
              // Step 1: Delete services and deployments
              sh 'kubectl delete svc kubernetes || true'
              sh 'kubectl delete deploy pandacloud-app || true'
              sh 'kubectl delete svc pandacloud-app || true'

              // Step 2: Delete ArgoCD installation and namespace
              sh 'kubectl delete -n argocd -f https://raw.githubusercontent.com/argoproj/argo-
cd/stable/manifests/install.yaml || true'
              sh 'kubectl delete namespace argocd || true'

              // Step 3: List and uninstall Helm releases in prometheus namespace
              sh 'helm list -n prometheus || true'
              sh 'helm uninstall kube-stack -n prometheus || true'

              // Step 4: Delete prometheus namespace
              sh 'kubectl delete namespace prometheus || true'

              // Step 5: Remove Helm repositories
              sh 'helm repo remove stable || true'
              sh 'helm repo remove prometheus-community || true'
           }
        }
     }

     stage('Delete ECR Repository and KMS Keys') {
           steps {
              script {
                  // Step 1: Delete ECR Repository
                  sh '''
                  aws ecr delete-repository --repository-name amazon-prime --region us-east-1 --
force
                  '''

                  // Step 2: Delete KMS Keys
                  sh '''
                  for key in $(aws kms list-keys --region us-east-1 --query "Keys[*].KeyId" --output
text); do
                      aws kms disable-key --key-id $key --region us-east-1
                      aws kms schedule-key-deletion --key-id $key --pending-window-in-days 7 --
region us-east-1
                  done
                  '''
              }
           }
        }

     }
}
```

Cleanup-pipeline



Dashboard > cleanup-pipeline >

cleanup-pipeline

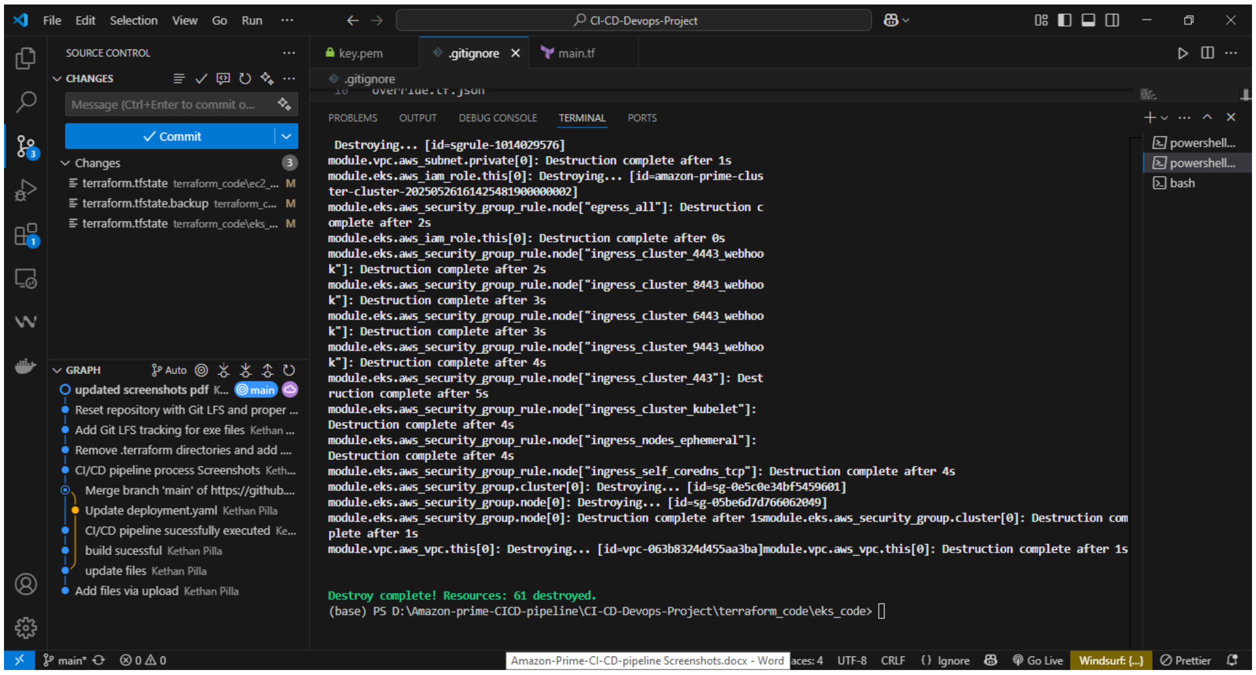
Stage View

	Login to EKS	Cleanup K8s Resources	Delete ECR Repository and KMS Keys
Average stage times: (full run time: ~2min 20s)	1s	2min 14s	3s
#1 May 26 17:19 No Changes	1s	2min 14s	3s

Permalinks

- Last build (#1), 4 min 53 sec ago
- Last stable build (#1), 4 min 53 sec ago
- Last successful build (#1), 4 min 53 sec ago
- Last completed build (#1), 4 min 53 sec ago

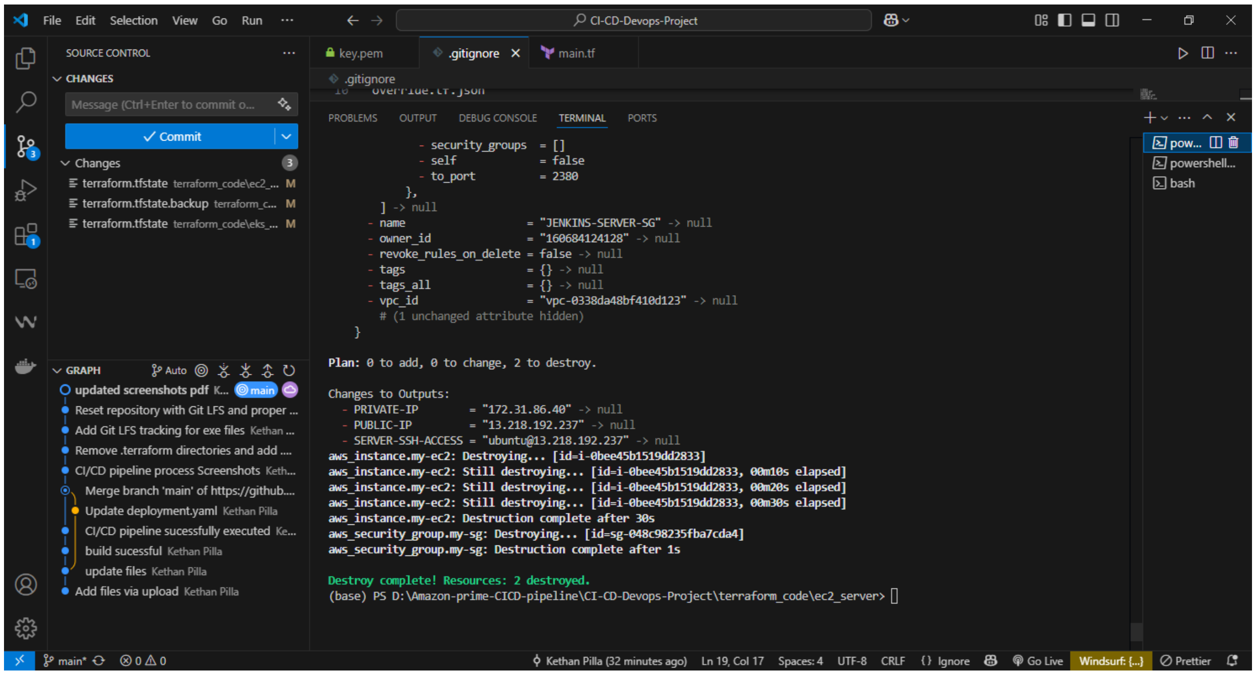
Clean Terraform



```
 Destroying... [id=sgrule-1014029576]
module.vpc.aws_subnet.private[0]: Destruction complete after 1s
module.eks.aws_iam_role.this[0]: Destroying... [id=amazon-prime-clus
ter-cluster-20250526161425481900000002]
module.eks.aws_security_group_rule.node["egress_all"]: Destruction c
omplete after 2s
module.eks.aws_iam_role.this[0]: Destruction complete after 0s
module.eks.aws_security_group_rule.node["ingress_cluster_4443_webhoo
k"]: Destruction complete after 2s
module.eks.aws_security_group_rule.node["ingress_cluster_8443_webhoo
k"]: Destruction complete after 3s
module.eks.aws_security_group_rule.node["ingress_cluster_6443_webhoo
k"]: Destruction complete after 3s
module.eks.aws_security_group_rule.node["ingress_cluster_9443_webhoo
k"]: Destruction complete after 4s
module.eks.aws_security_group_rule.node["ingress_cluster_443"]: Dest
ruction complete after 5s
module.eks.aws_security_group_rule.node["ingress_cluster_kubelet"]:
Destruction complete after 4s
module.eks.aws_security_group_rule.node["ingress_nodes_ephemeral"]:
Destruction complete after 4s
module.eks.aws_security_group_rule.node["ingress_self_coredns_tcp"]: Destruction complete after 4s
module.eks.aws_security_group.cluster[0]: Destroying... [id=sg-0e5c0e34bf5459601]
module.eks.aws_security_group.node[0]: Destroying... [id=sg-05be6d7d766062049]
module.eks.aws_security_group.node[0]: Destruction complete after 1smodule.eks.aws_security_group.cluster[0]: Destruction com
plete after 1s
module.vpc.aws_vpc.this[0]: Destroying... [id=vpc-063b8324d455aa3ba]module.vpc.aws_vpc.this[0]: Destruction complete after 1s


Destroy complete! Resources: 61 destroyed.
(base) PS D:\Amazon-prime-CICD-pipeline\CI-CD-Devops-Project\terraform_code\eks_code>
```

SOURCE CONTROL — Changes: terraform.tfstate, terraform.tfstate.backup, terraform.tfstate (M)

GRAPH:
- updated screenshots pdf
- Reset repository with Git LFS and proper ...
- Add Git LFS tracking for exe files
- Remove .terraform directories and add ...
- CI/CD pipeline process Screenshots
- Merge branch 'main' of https://github...
- Update deployment.yaml
- CI/CD pipeline sucessfully executed
- build sucessful
- update files
- Add files via upload


CI-CD-Devops-Project
SOURCE CONTROL
CHANGES
Commit
Changes 3
terraform.tfstate terraform_code\ec2_... M
terraform.tfstate.backup terraform_c... M
terraform.tfstate terraform_code\eks_... M
GRAPH
updated screenshots pdf K... main
Reset repository with Git LFS and proper ...
Add Git LFS tracking for exe files Kethan ...
Remove .terraform directories and add ...
CI/CD pipeline process Screenshots Keth...
Merge branch 'main' of https://github....
Update deployment.yaml Kethan Pilla
CI/CD pipeline sucessfully executed Ke...
build sucessful Kethan Pilla
update files Kethan Pilla
Add files via upload Kethan Pilla
key.pem
.gitignore
main.tf
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
pow...
powershell...
bash
main*
Kethan Pilla (32 minutes ago)
Ln 19, Col 17
Spaces: 4
UTF-8
CRLF
Ignore
Go Live
Windsurf: {...}
Prettier

```
              - security_groups  = []
              - self             = false
              - to_port          = 2380
            },
        ] -> null
      - name                   = "JENKINS-SERVER-SG" -> null
      - owner_id               = "160684124128" -> null
      - revoke_rules_on_delete = false -> null
      - tags                   = {} -> null
      - tags_all               = {} -> null
      - vpc_id                 = "vpc-0338da48bf410d123" -> null
        # (1 unchanged attribute hidden)
    }

Plan: 0 to add, 0 to change, 2 to destroy.

Changes to Outputs:
  - PRIVATE-IP        = "172.31.86.40" -> null
  - PUBLIC-IP         = "13.218.192.237" -> null
  - SERVER-SSH-ACCESS = "ubuntu@13.218.192.237" -> null
aws_instance.my-ec2: Destroying... [id=i-0bee45b1519dd2833]
aws_instance.my-ec2: Still destroying... [id=i-0bee45b1519dd2833, 00m10s elapsed]
aws_instance.my-ec2: Still destroying... [id=i-0bee45b1519dd2833, 00m20s elapsed]
aws_instance.my-ec2: Still destroying... [id=i-0bee45b1519dd2833, 00m30s elapsed]
aws_instance.my-ec2: Destruction complete after 30s
aws_security_group.my-sg: Destroying... [id=sg-048c98235fba7cda4]
aws_security_group.my-sg: Destruction complete after 1s

Destroy complete! Resources: 2 destroyed.
(base) PS D:\Amazon-prime-CICD-pipeline\CI-CD-Devops-Project\terraform_code\ec2_server>
```