TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG

BÁO CÁO THỰC HÀNH

Bài thực hành số 02: Quản lý và triển khai hạ tầng AWS và ứng dụng microservices với Terraform, CloudFormation, GitHub Actions, AWS CodePipeline và Jenkins

Môn học: NT548

Lớp: NT548.P11.MMCL

THÀNH VIÊN THỰC HIỆN (Nhóm 21):

STT	Họ và tên	MSSV
1	Lê Triệu Vỉ	21522785
2	Lý Thế Nguyên	21552389
3		

Điểm tự đánh giá
9

ĐÁNH GIÁ KHÁC:

Tổng thời gian thực hiện	
Phân chia công việc	
Ý kiến *(nếu có)* + Khó khăn + Đề xuất, kiến nghị	

Phần bên dưới của báo cáo này là báo cáo chi tiết của nhóm thực hiện

A. BÁO CÁO CHI TIẾT

Source code câu 1: https://github.com/just-vile/NT548-Lab2.git

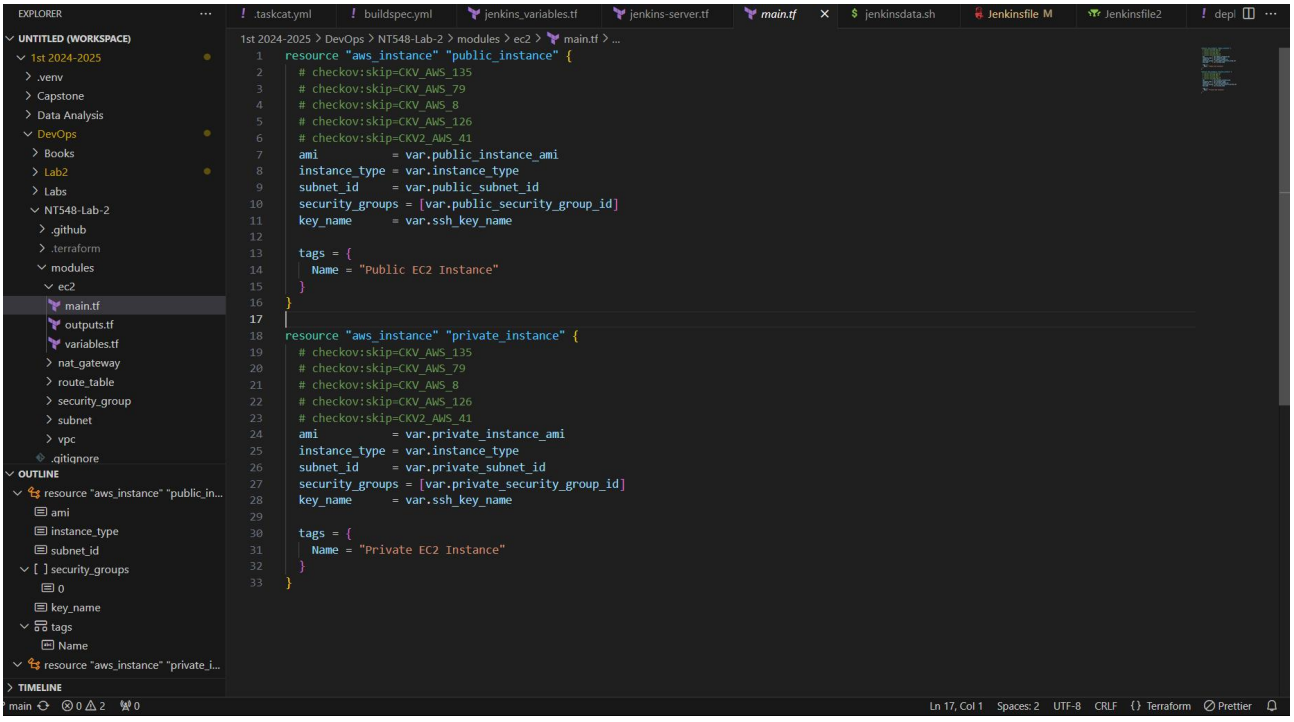
Source code câu 2: https://github.com/just-vile/NT548-Lab2-Cloudformation.git

Source code câu 3: https://github.com/just-vile/Multi-Tier-MicroService-Appliction-Deployment.git. branch: latest

1. Câu 1

a. Mục 1

Sử dụng Checkov để scan từng module ở local:




```
D:\Learning Space\1st 2024-2025\DevOps\NT548-Lab-2>c:\users\asus\appdata\roaming\python\python312\scripts\checkov.cmd -d modules\ec2
2024-11-09 17:02:51,767 [ThreadPoolEx] [WARNI]  [ArmLocalGraph] created 0 vertices
2024-11-09 17:02:51,767 [ThreadPoolEx] [WARNI]  [ArmLocalGraph] created 0 edges
[ terraform framework ]: 100%|[3/3], Current File Scanned=modules\ec2\variables.tf
[ secrets framework ]: 100%|[3/3], Current File Scanned=modules\ec2\variables.tftf

       _               _
   ___| |__   ___  ___| | _______   __
  / __| '_ \ / _ \/ __| |/ / _ \ \ / /
 | (__| | | |  __/ (__|   < (_) \ V /
  \___|_| |_|\___|\___|_|\_\___/ \_/

By Prisma Cloud | version: 3.2.281
Update available         -> 3.2.283
Run pip3 install -U checkov to update


terraform scan results:

Passed checks: 4, Failed checks: 0, Skipped checks: 10
```

Ở module EC2, Checkov detect một vài lỗi, tuy nhiên, phần lớn đều không liên quan đến yêu cầu ở Lab 1, nên em đã comment lại trong code để Checkov skip qua.

Các lỗi mà nhóm đã skip:

```
Check: CKV_AWS_126: "Ensure that detailed monitoring is enabled for EC2 instances"
        SKIPPED for resource: aws_instance.public_instance
        Suppress comment: No comment provided
        File: \main.tf:1-16
        Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/aws-logging-policies/ensure-that-detailed-monitoring-is-enabled-for-ec2-instances
Check: CKV_AWS_135: "Ensure that EC2 is EBS optimized"
        SKIPPED for resource: aws_instance.public_instance
        Suppress comment: No comment provided
        File: \main.tf:1-16
        Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/aws-general-policies/ensure-that-ec2-is-ebs-optimized
Check: CKV_AWS_79: "Ensure Instance Metadata Service Version 1 is not enabled"
        SKIPPED for resource: aws_instance.public_instance
        Suppress comment: No comment provided
        File: \main.tf:1-16
        Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/aws-general-policies/bc-aws-general-31
Check: CKV_AWS_8: "Ensure all data stored in the Launch configuration or instance Elastic Blocks Store is securely encrypted"
        SKIPPED for resource: aws_instance.public_instance
        Suppress comment: No comment provided
        File: \main.tf:1-16
        Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/aws-general-policies/general-13
Check: CKV_AWS_126: "Ensure that detailed monitoring is enabled for EC2 instances"
        SKIPPED for resource: aws_instance.private_instance
        Suppress comment: No comment provided
        File: \main.tf:18-33
        Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/aws-logging-policies/ensure-that-detailed-monitoring-is-enabled-for-ec2-instances
Check: CKV_AWS_135: "Ensure that EC2 is EBS optimized"
        SKIPPED for resource: aws_instance.private_instance
        Suppress comment: No comment provided
        File: \main.tf:18-33
        Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/aws-general-policies/ensure-that-ec2-is-ebs-optimized
Check: CKV_AWS_79: "Ensure Instance Metadata Service Version 1 is not enabled"
        SKIPPED for resource: aws_instance.private_instance
        Suppress comment: No comment provided
        File: \main.tf:18-33
        Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/aws-general-policies/bc-aws-general-31
Check: CKV_AWS_8: "Ensure all data stored in the Launch configuration or instance Elastic Blocks Store is securely encrypted"
        SKIPPED for resource: aws_instance.private_instance
        Suppress comment: No comment provided
        File: \main.tf:18-33
        Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/aws-general-policies/general-13
Check: CKV2_AWS_41: "Ensure an IAM role is attached to EC2 instance"
        SKIPPED for resource: aws_instance.public_instance
```

Tương tự với các module khác …

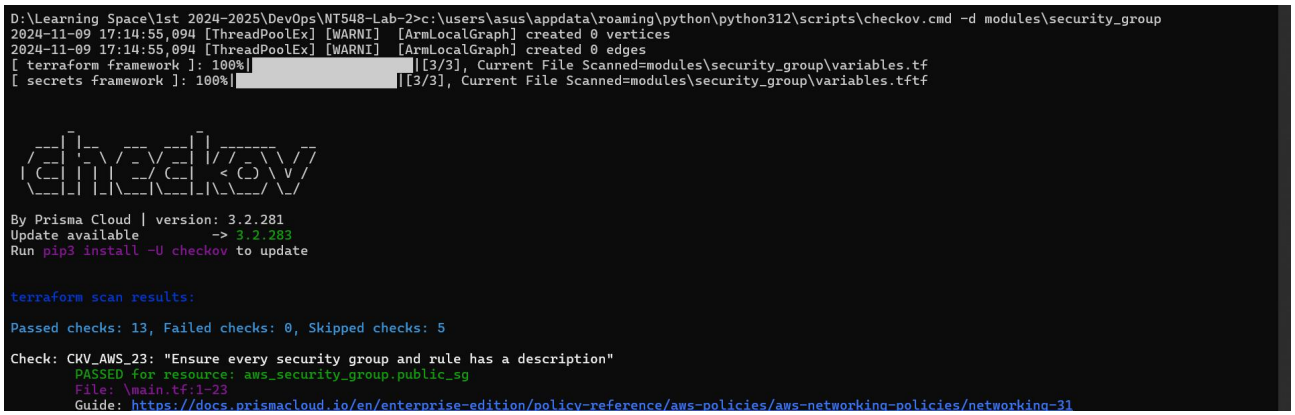
Module nat_gateway

```
D:\Learning Space\1st 2024-2025\DevOps\NT548-Lab-2>c:\users\asus\appdata\roaming\python\python312\scripts\checkov.cmd -d modules\nat_gateway
2024-11-09 17:03:57,940 [ThreadPoolEx] [WARNI]  [ArmLocalGraph] created 0 vertices
2024-11-09 17:03:57,942 [ThreadPoolEx] [WARNI]  [ArmLocalGraph] created 0 edges
[ terraform framework ]: 100%|[3/3], Current File Scanned=modules\nat_gateway\variables.tf
[ secrets framework ]: 100%|[3/3], Current File Scanned=modules\nat_gateway\variables.tf
[ secrets framework ]:  67%|[2/3], Current File Scanned=modules\nat_gateway\variables.tf

       _               _
   ___| |__   ___  ___| | _______   __
  / __| '_ \ / _ \/ __| |/ / _ \ \ / /
 | (__| | | |  __/ (__|   < (_) \ V /
  \___|_| |_|\___|\___|_|\_\___/ \_/

By Prisma Cloud | version: 3.2.281
Update available         -> 3.2.283
Run pip3 install -U checkov to update


terraform scan results:

Passed checks: 1, Failed checks: 0, Skipped checks: 0

Check: CKV2_AWS_19: "Ensure that all EIP addresses allocated to a VPC are attached to EC2 instances"
        PASSED for resource: aws_eip.nat_eip
        File: \main.tf:1-3
        Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/aws-networking-policies/ensure-that-all-eip-addresses-allocated-to-a-vpc-are-attached-to-ec2-instances
```

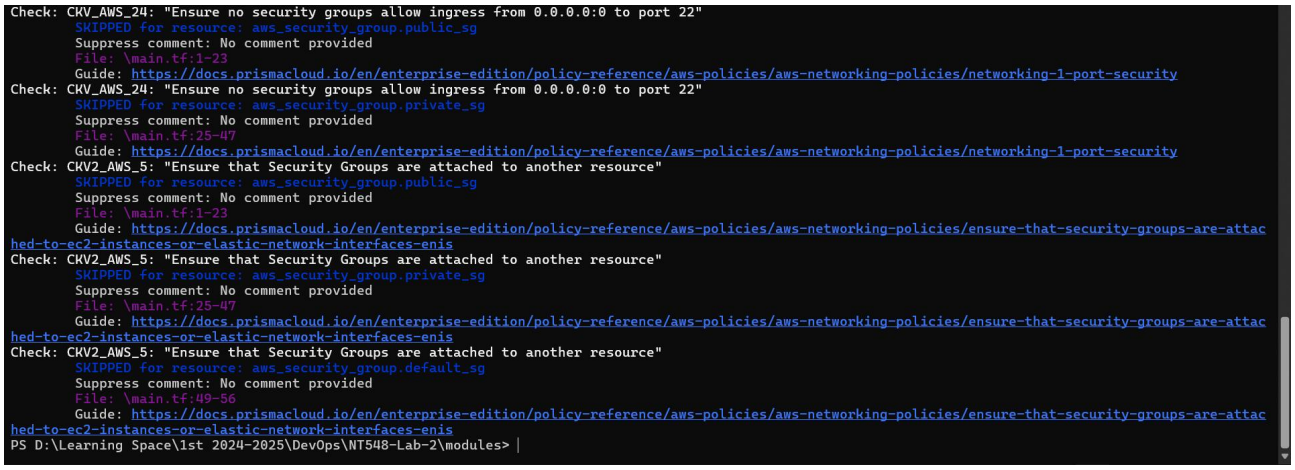
Module route_table

```
D:\Learning Space\1st 2024-2025\DevOps\NT548-Lab-2>c:\users\asus\appdata\roaming\python\python312\scripts\checkov.cmd -d modules\route_table
2024-11-09 17:07:23,911 [ThreadPoolEx] [WARNI]  [ArmLocalGraph] created 0 vertices
2024-11-09 17:07:23,911 [ThreadPoolEx] [WARNI]  [ArmLocalGraph] created 0 edges
[ terraform framework ]: 100%|[3/3], Current File Scanned=modules\route_table\variables.tf
[ secrets framework ]: 100%|[3/3], Current File Scanned=modules\route_table\variables.tf
[ secrets framework ]:  67%|[2/3], Current File Scanned=modules\route_table\variables.tf

       _               _
   ___| |__   ___  ___| | _______   __
  / __| '_ \ / _ \/ __| |/ / _ \ \ / /
 | (__| | | |  __/ (__|   < (_) \ V /
  \___|_| |_|\___|\___|_|\_\___/ \_/

By Prisma Cloud | version: 3.2.281
Update available         -> 3.2.283
Run pip3 install -U checkov to update


terraform scan results:

Passed checks: 4, Failed checks: 0, Skipped checks: 0
```

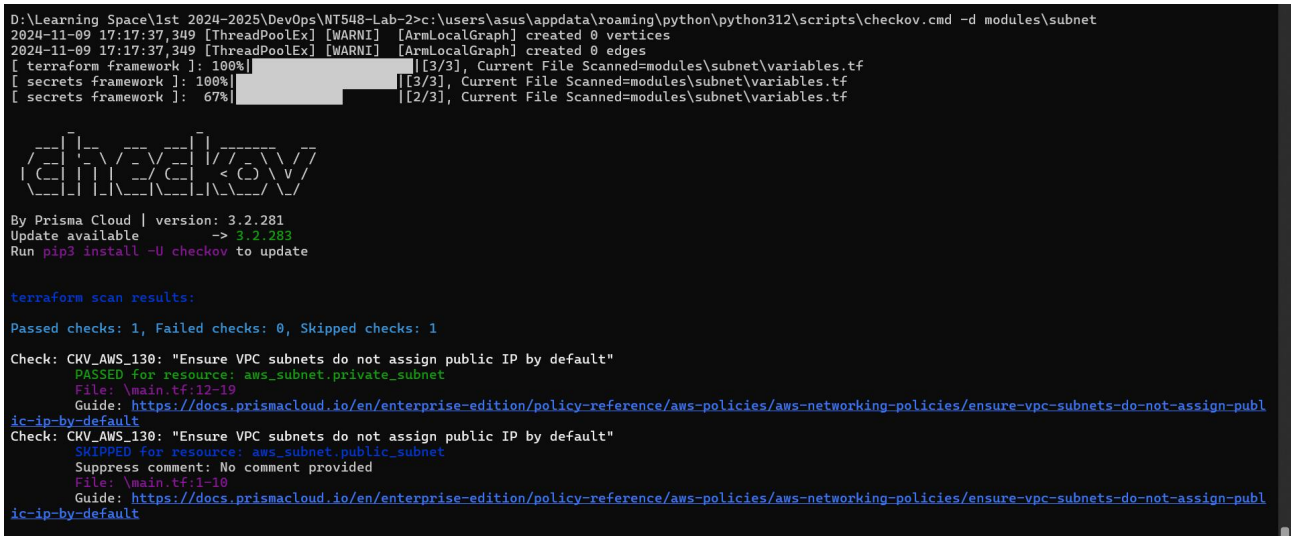
Module security_group



Các lỗi đã skip ở module Security Group



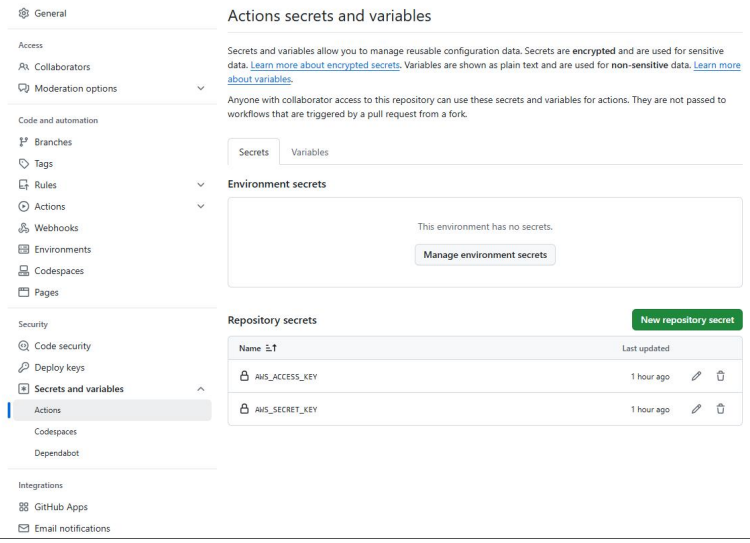
Module subnet:



Tạo thư mục .github/workflow. Sau đó tạo file deploy.yaml để tự động quá trình triển khai terraform với github actions.

```yaml
name: 'Terraform Infastrucutre Deployment with Checkov Scan'

on:
  push:
    branches:
      - main
  pull_request:
    branches:
      - main

env:
  AWS_ACCESS_KEY_ID: ${{ secrets.aws_access_key }}
  AWS_SECRET_ACCESS_KEY: ${{ secrets.aws_secret_key }}
  AWS_REGION: us-east-1
  TF_VAR_region: us-east-1
```

Triển khai trên branch main, cập nhật AWS key trong section secret trên Github.



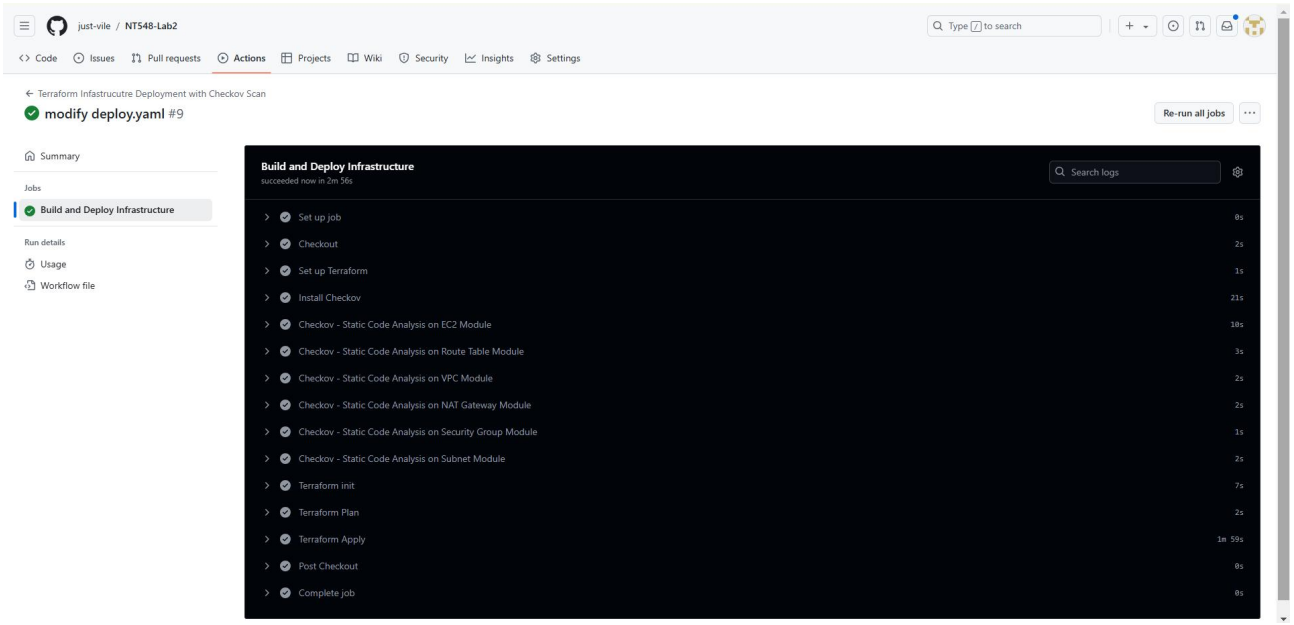
File deploy.yaml sẽ setup Terraform, tải Checkov và thực hiện scan trên từng modules.

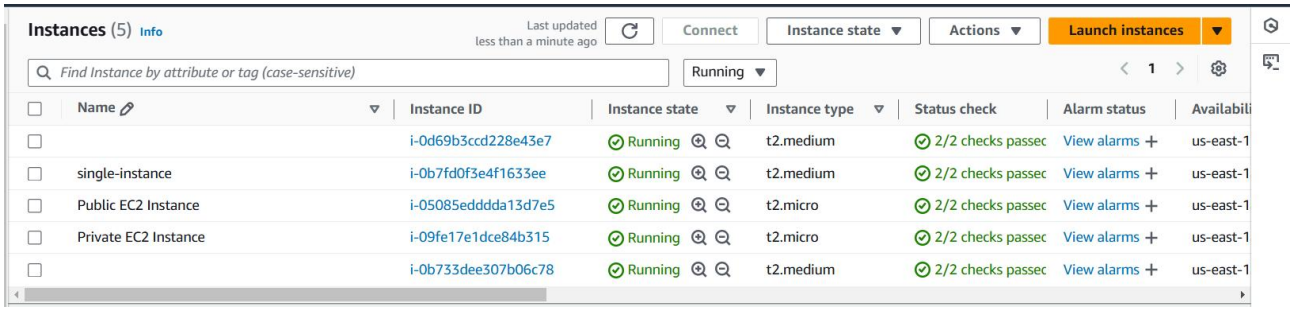
```yaml
jobs:
  build:
    name: Build and Deploy Infrastructure
    runs-on: ubuntu-latest
    steps:
      - name: Checkout
        uses: actions/checkout@v2

      - name: Set up Terraform
        uses: hashicorp/setup-terraform@v1

      - name: Install Checkov
        run: |
          python3 -m pip install --upgrade pip
          pip install checkov

      # Run Checkov on each module directory
      - name: Checkov - Static Code Analysis on EC2 Module
        run: checkov -d modules/ec2

      - name: Checkov - Static Code Analysis on Route Table Module
        run: checkov -d modules/route_table

      - name: Checkov - Static Code Analysis on VPC Module
        run: checkov -d modules/vpc

      - name: Checkov - Static Code Analysis on NAT Gateway Module
        run: checkov -d modules/nat_gateway

      - name: Checkov - Static Code Analysis on Security Group Module
        run: checkov -d modules/security_group

      - name: Checkov - Static Code Analysis on Subnet Module
        run: checkov -d modules/subnet
```

Sau đó sẽ khởi tạo và tự động deploy lên AWS.

```yaml
    # Initialize Terraform in the root directory
    - name: Terraform init
      id: init
      run: terraform init

    - name: Terraform Plan
      id: plan
      run: terraform plan -out=tfplan

    - name: Terraform Apply
      id: apply
      run: terraform apply --auto-approve tfplan

```

Kết quả:





Các cấu hình vẫn y như Lab 1.

2. Câu 2

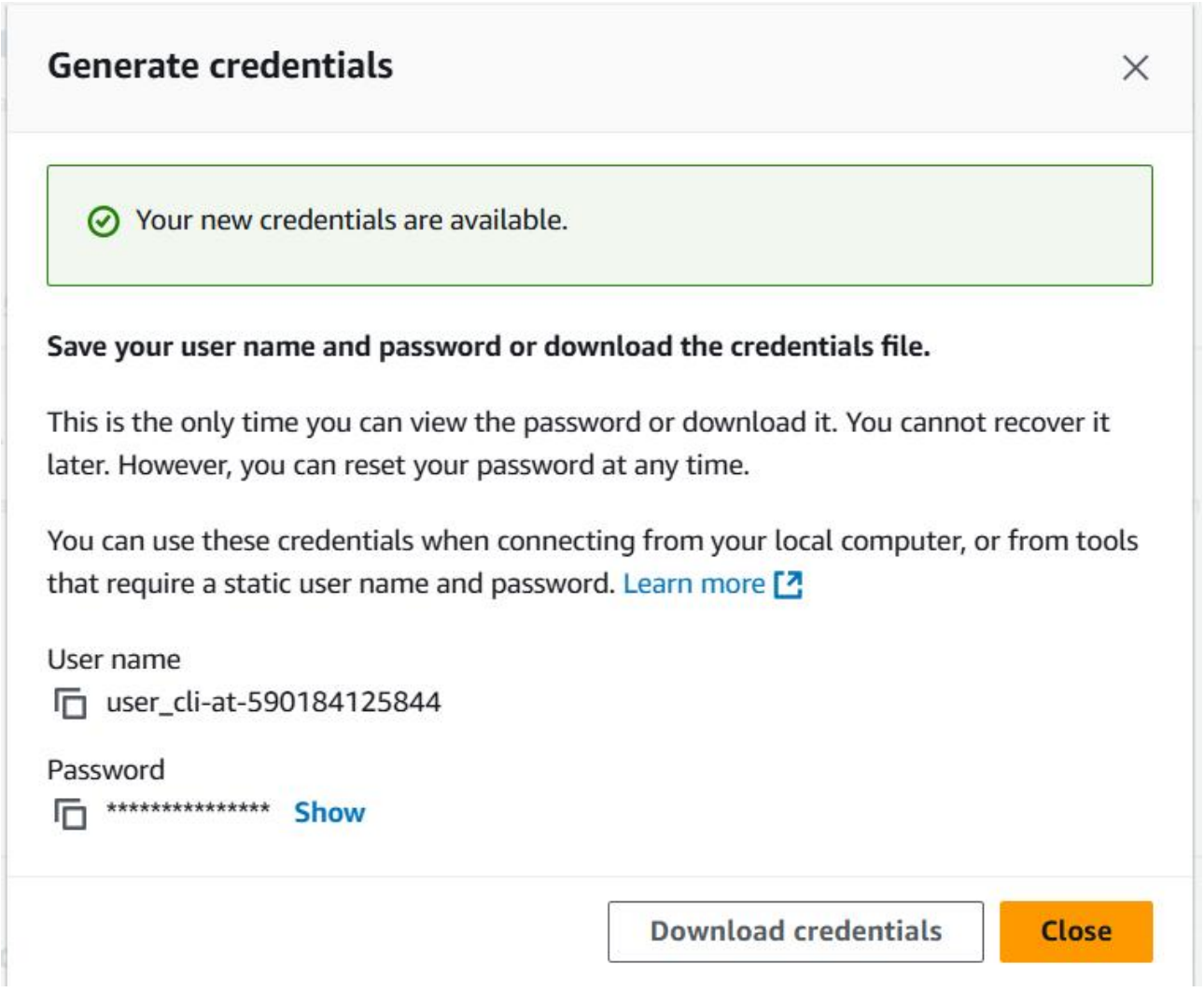
a. Mục 1

Trước tiên, ta tạo một role để tránh xảy ra xung đột khi Pipeline chạy:

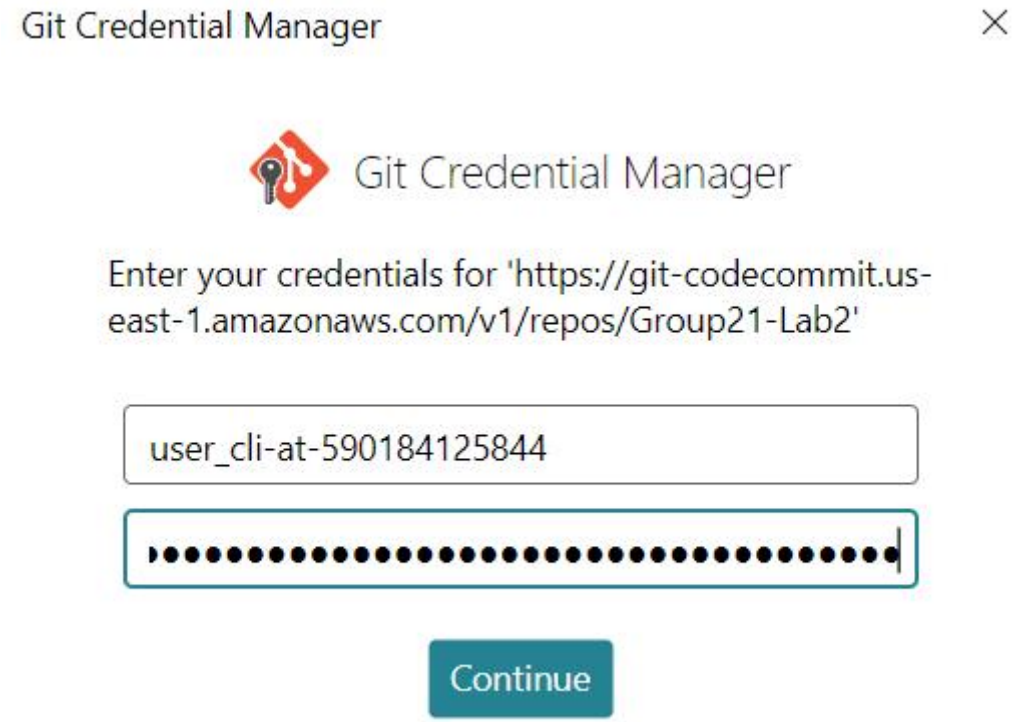
Cloud Control API	Full access	All resources	None
CloudFormation	Full access	All resources	None
CloudWatch Logs	Limited: Write	Multiple	None
CodeBuild	Full access	All resources	None
CodeDeploy	Full access	All resources	None
CodePipeline	Full access	All resources	None
EC2	Full access	All resources	None
IAM	Full access	All resources	None
S3	Full access	All resources	None

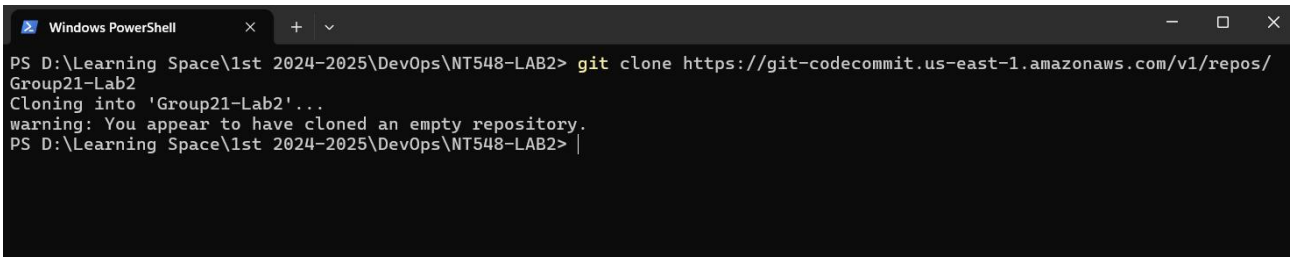
Đặt tên là 'Group21-CodeBuild-Role'

Vào IAM, sau đó tạo một Git Credential để truy cập CodeCommit.



Khi clone Repo trên CodeCommit về, sẽ bắt ta đăng nhập bằng Git Credentials trên.

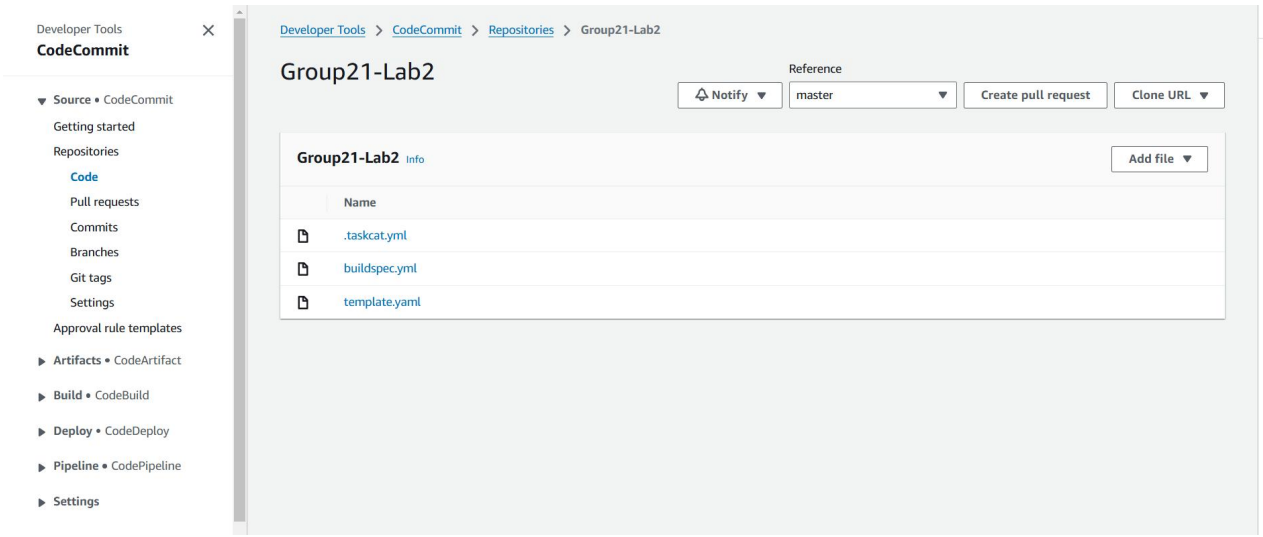




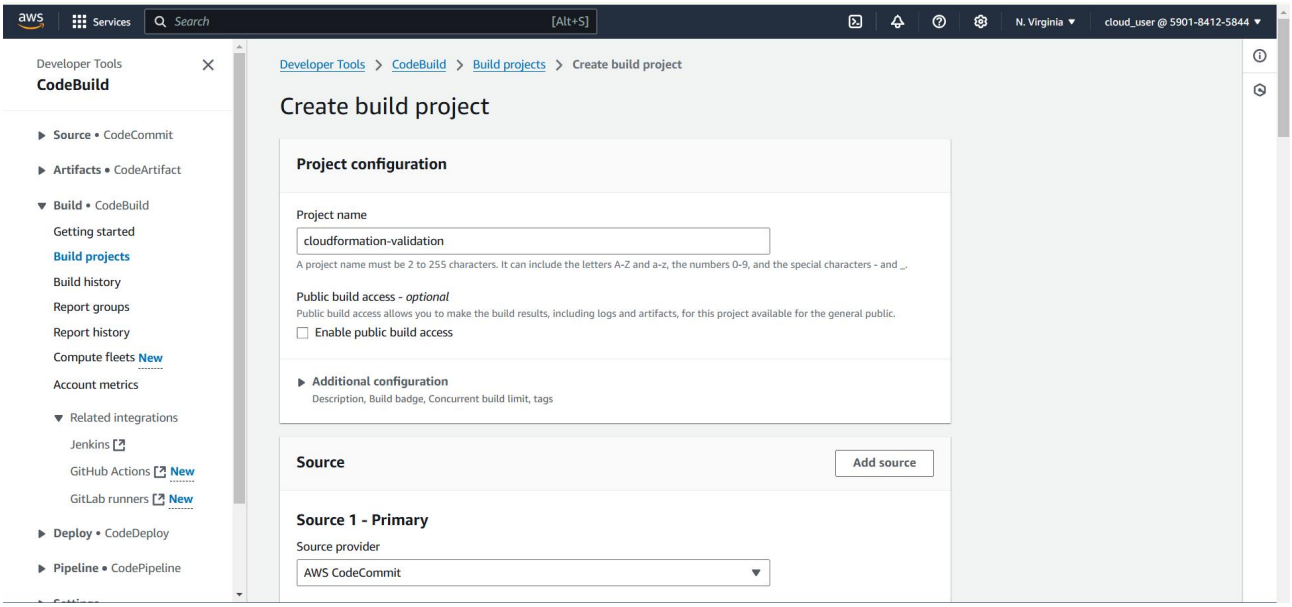
Push code từ Local lên CodeCommit.

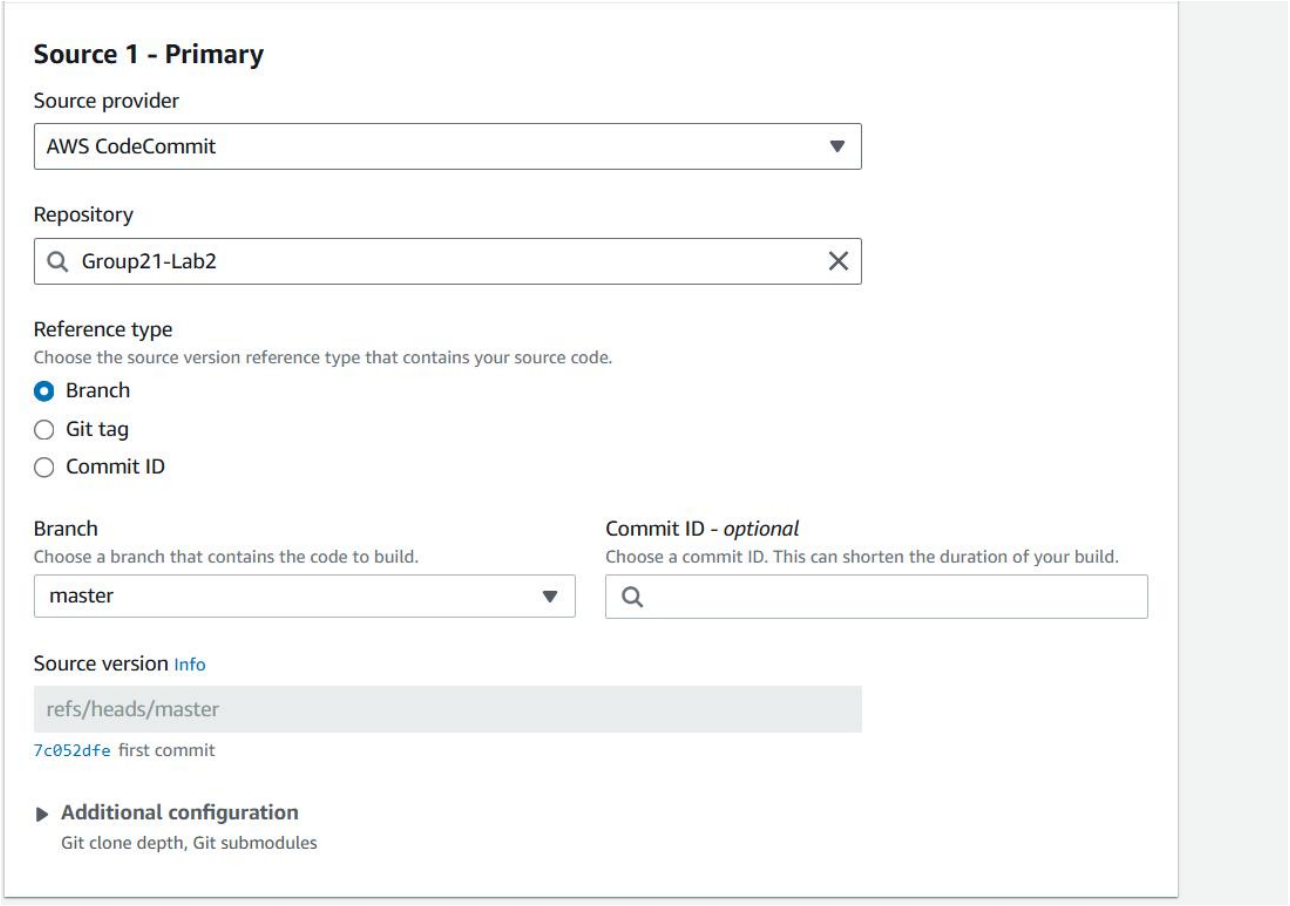
```
PS D:\Learning Space\1st 2024-2025\DevOps\NT548-LAB2\Group21-Lab2> git add .
PS D:\Learning Space\1st 2024-2025\DevOps\NT548-LAB2\Group21-Lab2> git commit -m 'first commit'
[master (root-commit) 7c052df] first commit
 3 files changed, 229 insertions(+)
 create mode 100644 .taskcat.yml
 create mode 100644 buildspec.yml
 create mode 100644 template.yaml
PS D:\Learning Space\1st 2024-2025\DevOps\NT548-LAB2\Group21-Lab2> git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 1.57 KiB | 1.57 MiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Validating objects: 100%
To https://git-codecommit.us-east-1.amazonaws.com/v1/repos/Group21-Lab2
 * [new branch]      master -> master
PS D:\Learning Space\1st 2024-2025\DevOps\NT548-LAB2\Group21-Lab2>
```

Đã push thành công lên CodeCommit.

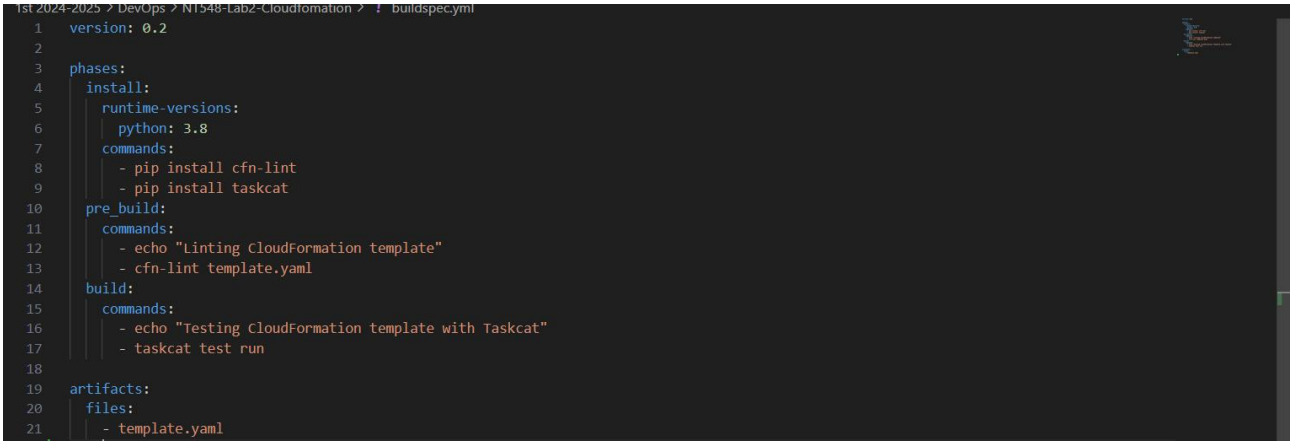


Kế tiếp, ta sẽ tạo một Build Project bằng CodeBuild, tên project là 'cloudformation-validation', source là từ CodeCommit khi nãy.



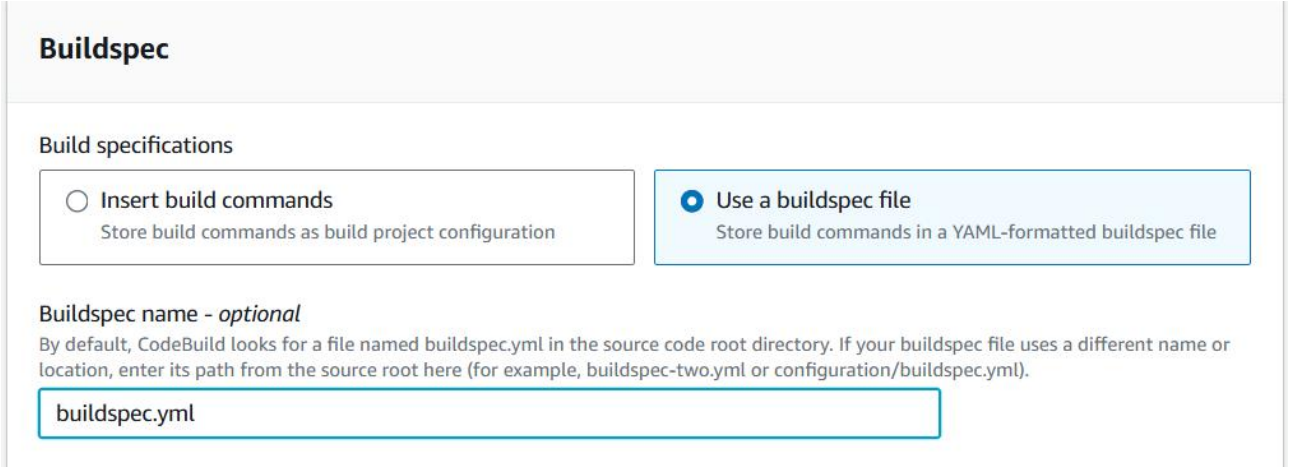


File build là 'buildspec.yml': thực hiện tải về cfn-lint và taskcat, sau đó scan file template để kiểm tra tính đúng đắn của nó, cuối cùng nó sẽ output ra file template.yaml:

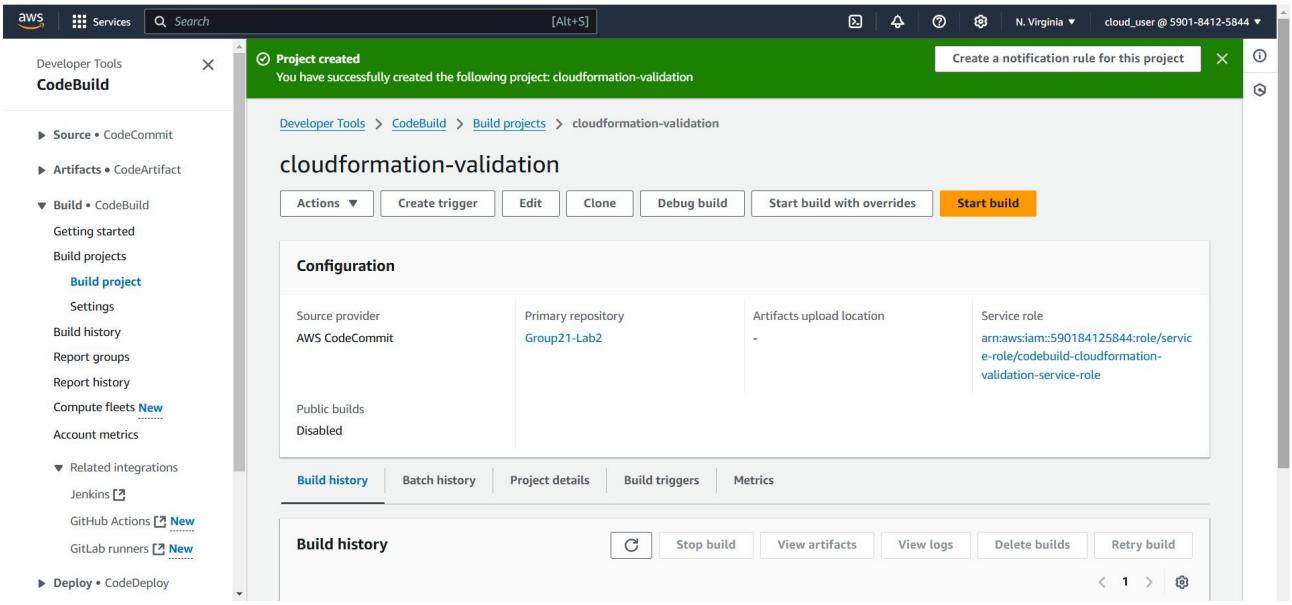


```yaml
version: 0.2

phases:
  install:
    runtime-versions:
      python: 3.8
    commands:
      - pip install cfn-lint
      - pip install taskcat
  pre_build:
    commands:
      - echo "Linting CloudFormation template"
      - cfn-lint template.yaml
  build:
    commands:
      - echo "Testing CloudFormation template with Taskcat"
      - taskcat test run

artifacts:
  files:
    - template.yaml
```

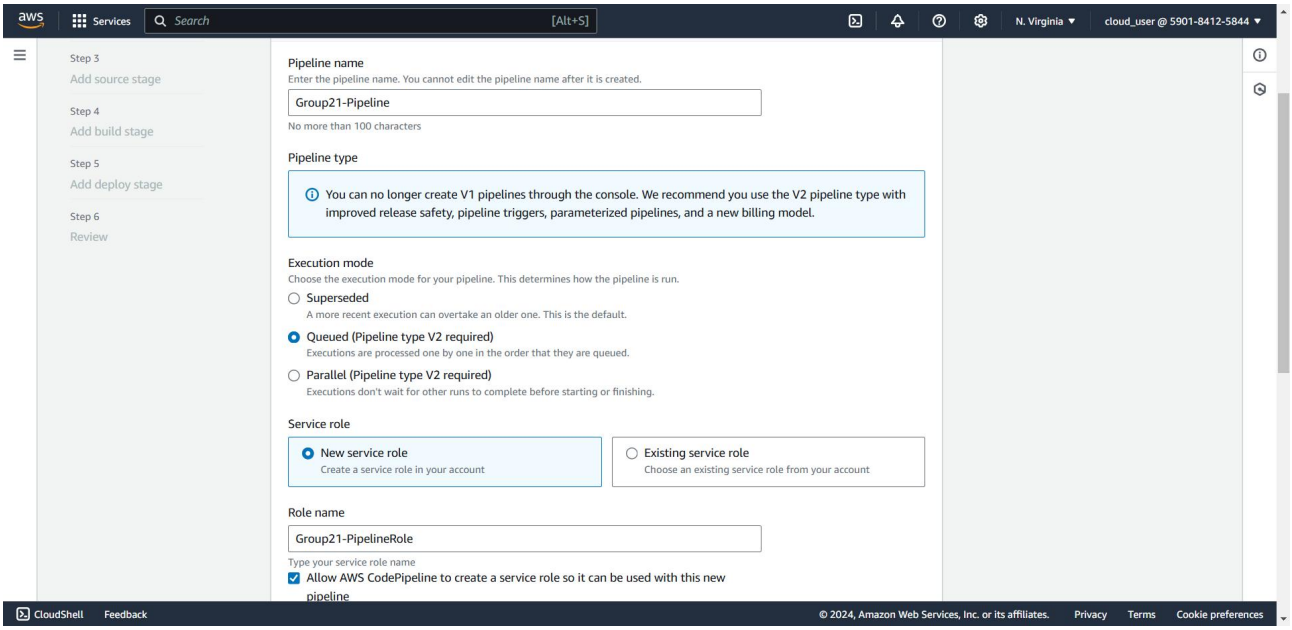
Chọn 'Use a buildspec file', và nhập đừng dẫn là file 'buildspec.yml'



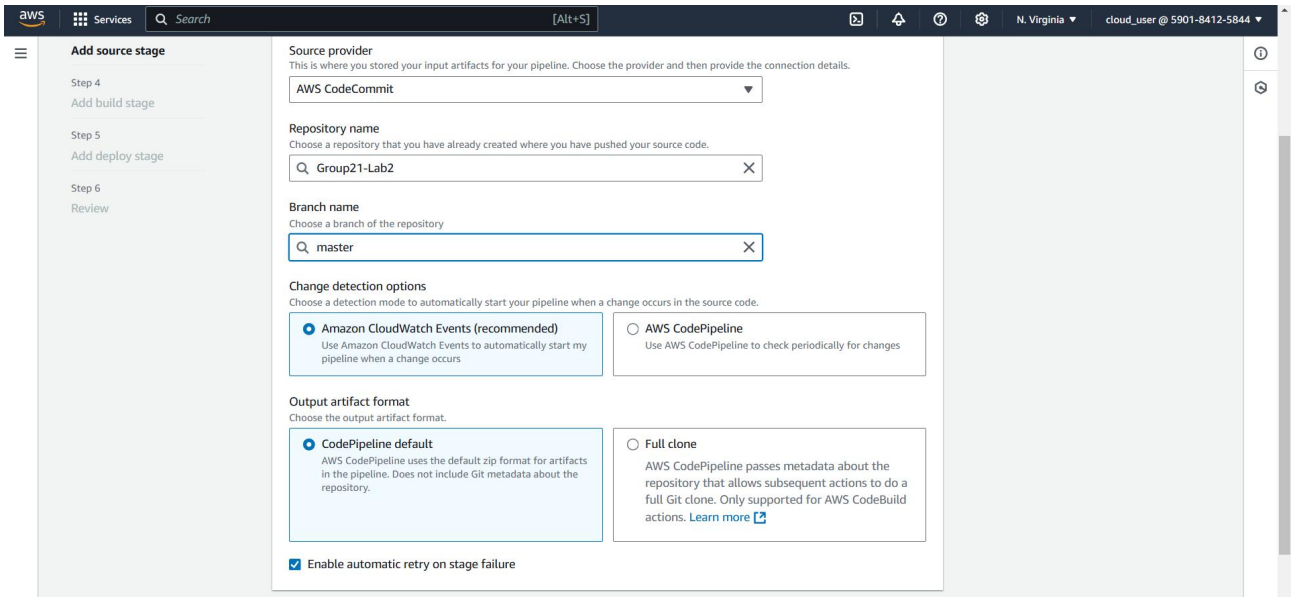
Khởi tạo Build Project thành công.



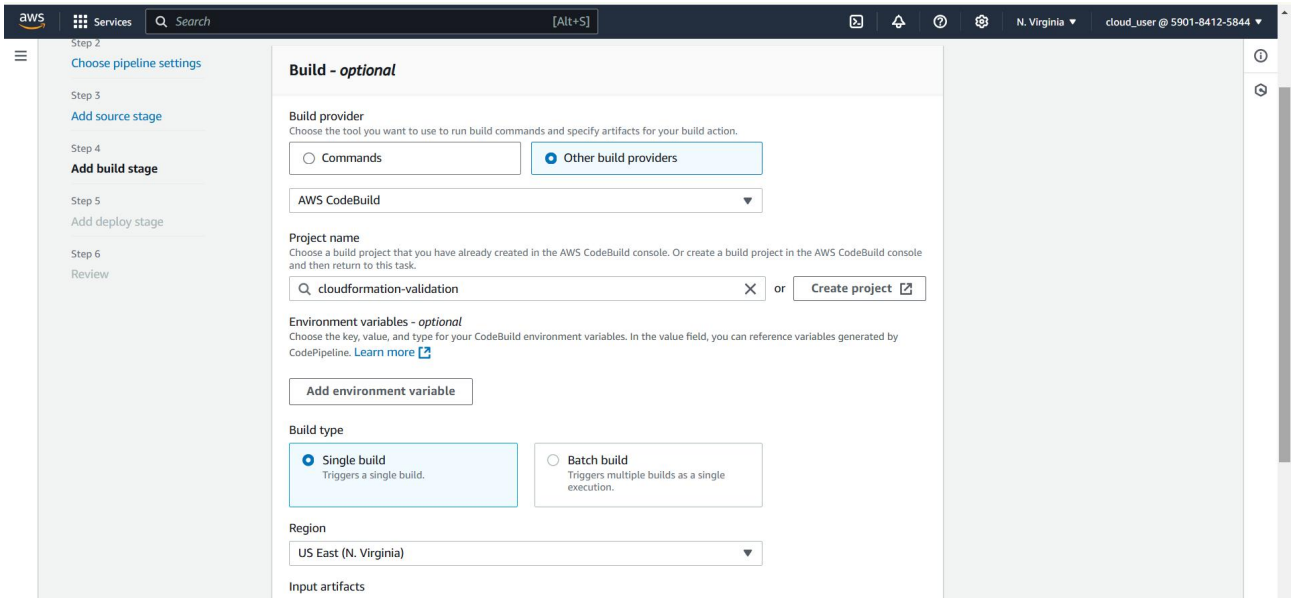
Tiếp đến ta sẽ tạo một Pipeline, tên là 'Group21-Pipeline', tạo một role là 'Group21-PipelineRole' (để mặc định cũng được).



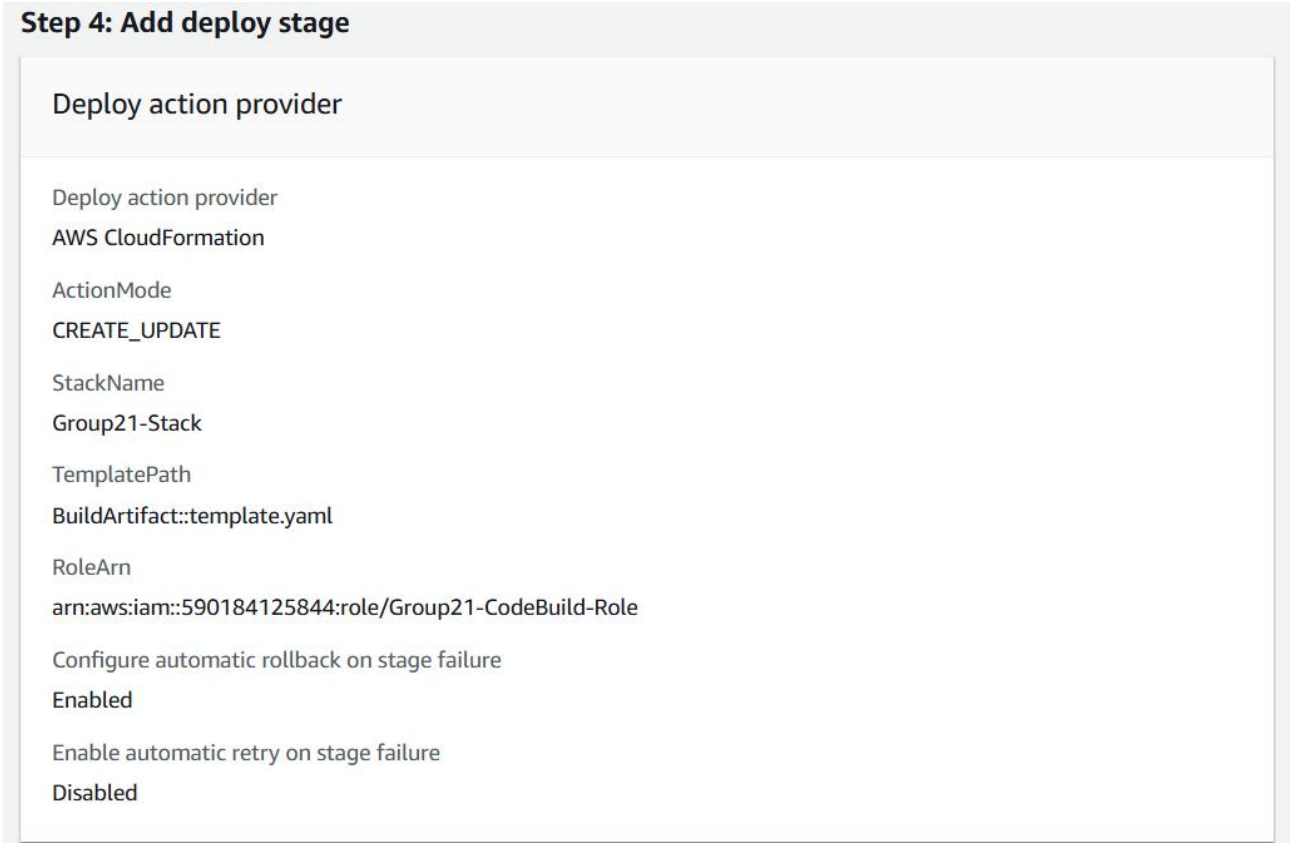
Source là từ CodeCommit, branch master.

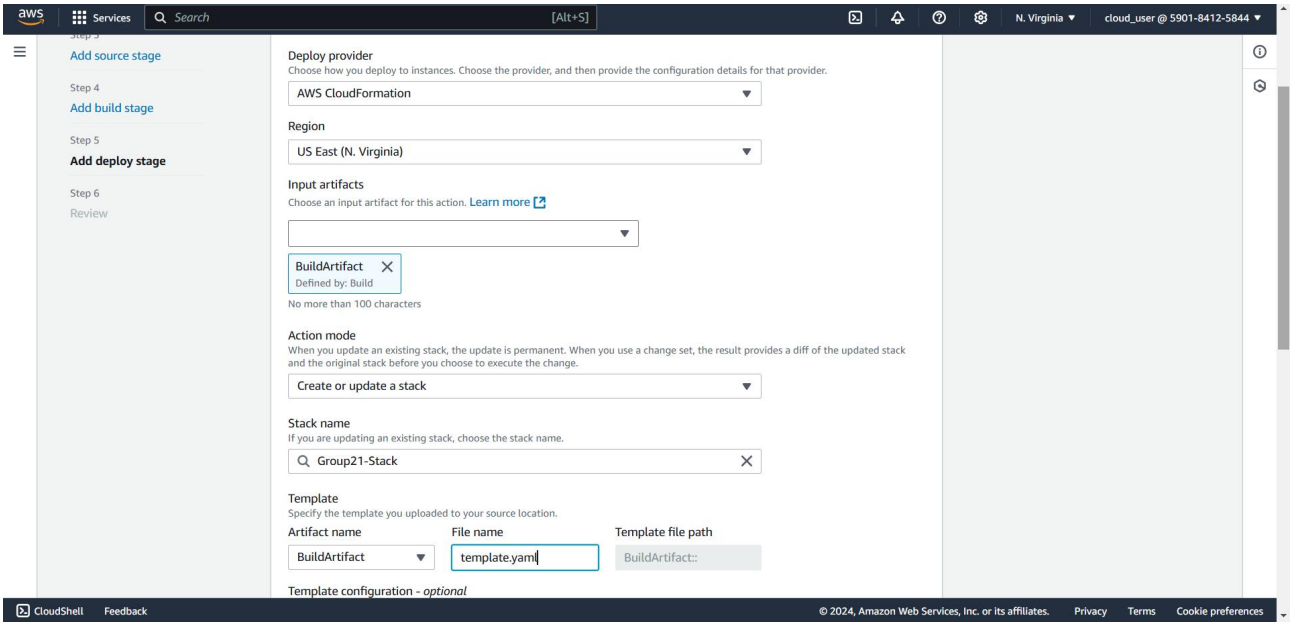


Sử dụng Build Project khi nãy

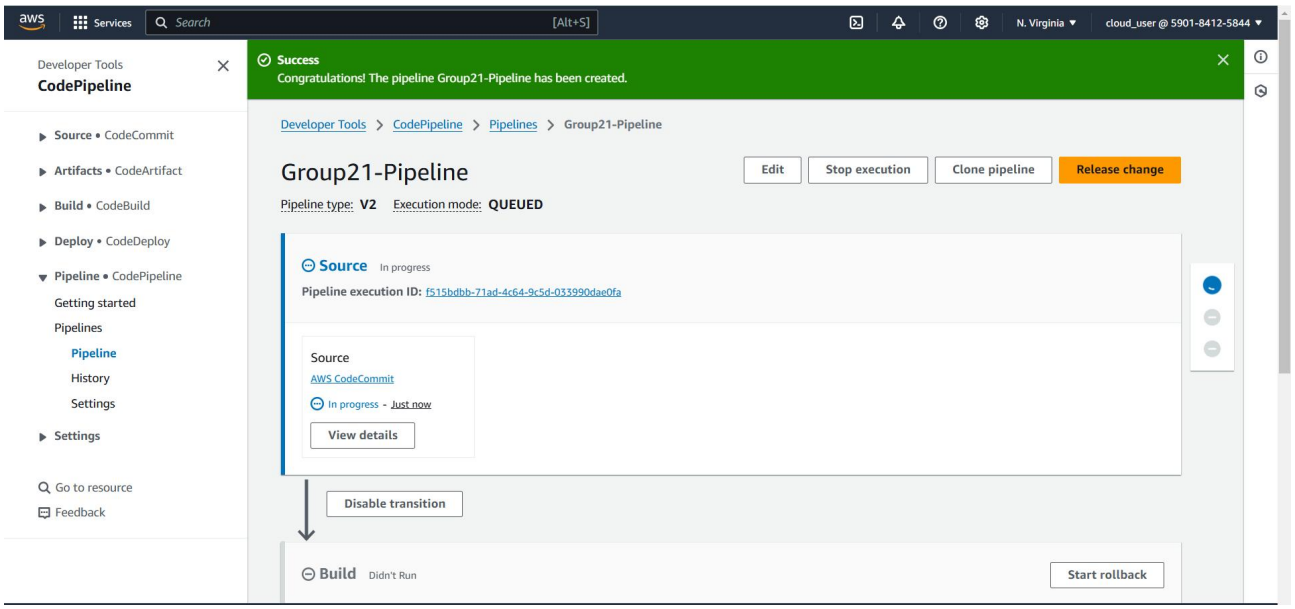


Chọn 'Deploy action provider' là AWS Cloudformation, CREATE_UPDATE, tên stack là 'Group21-Stack', TemplatePath là file template.yaml từ BuildArtifact, sử dụng Role 'Group21-CodeBuild-Role' vừa nãy tạo.

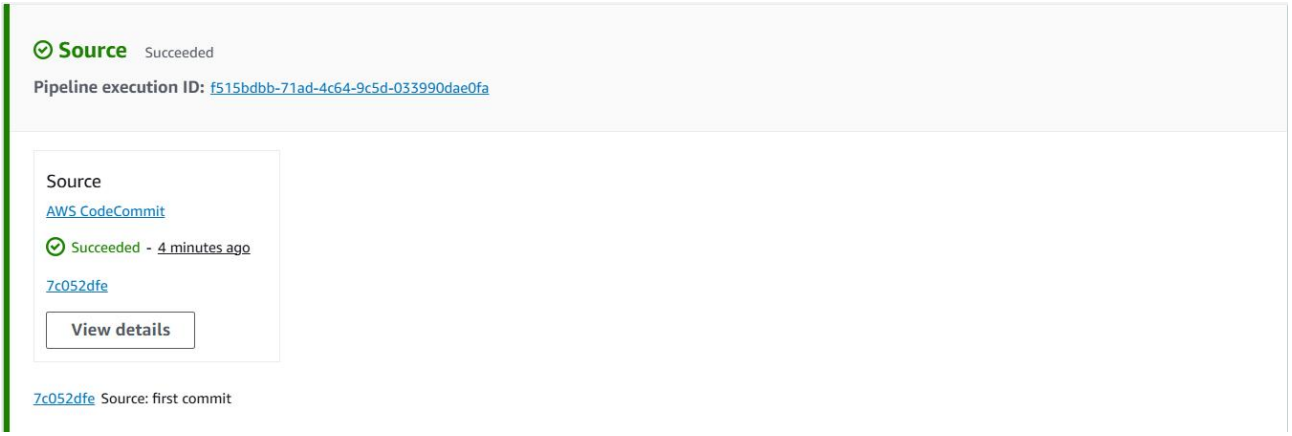




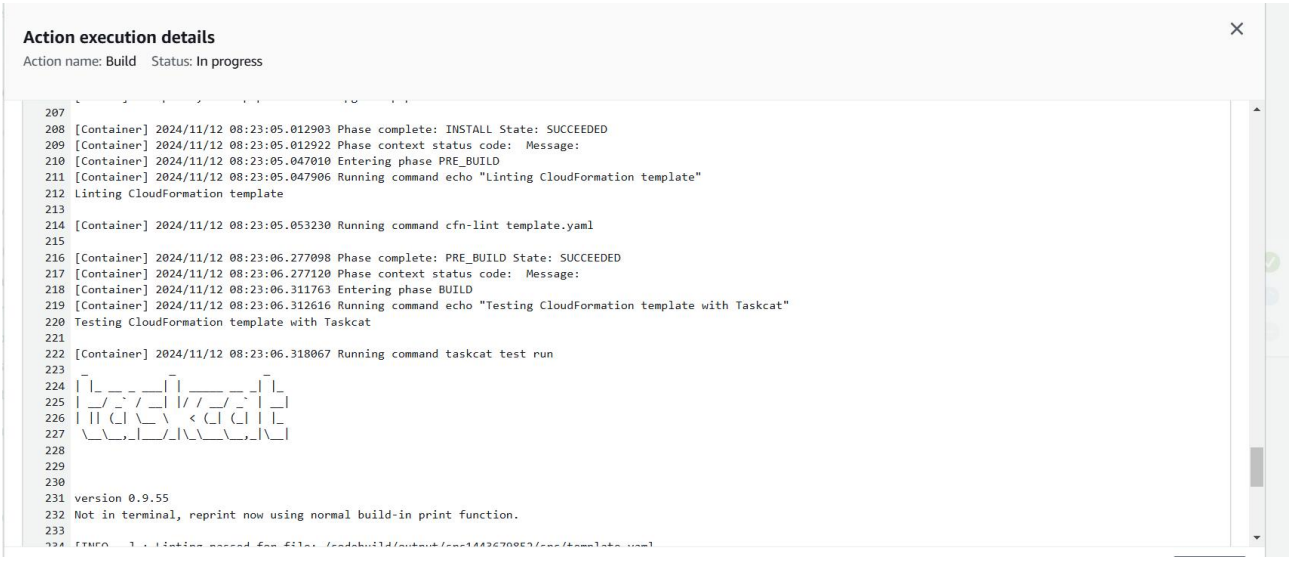
Chạy Pipeline vừa tạo.



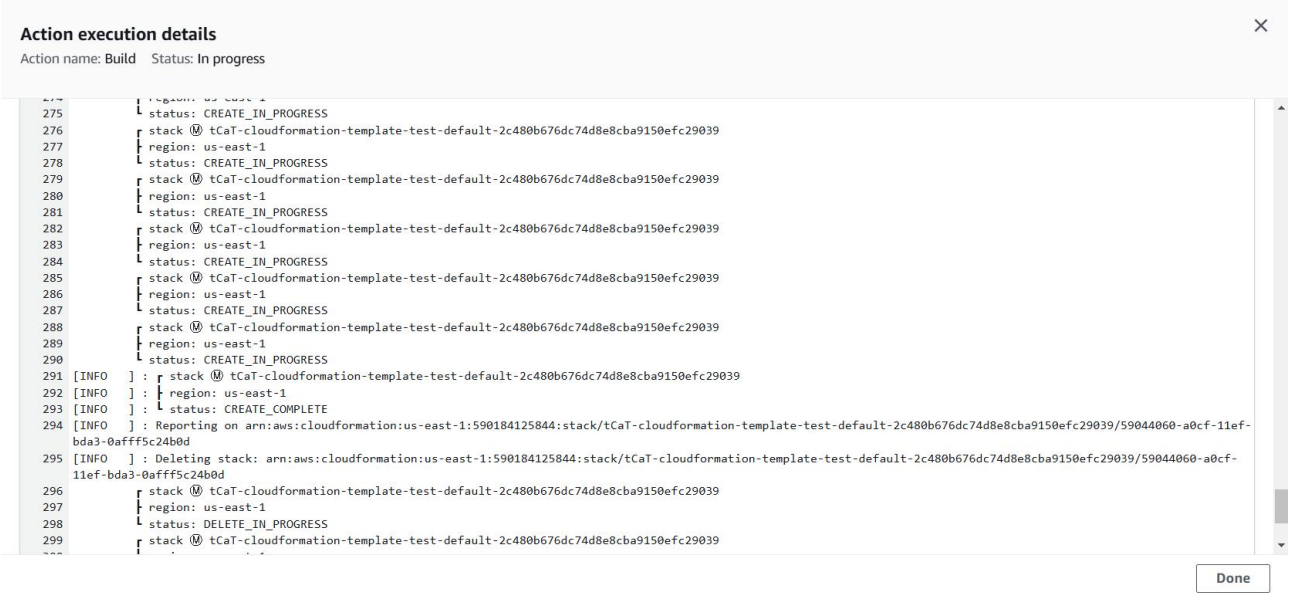
Source stage chạy thành công.

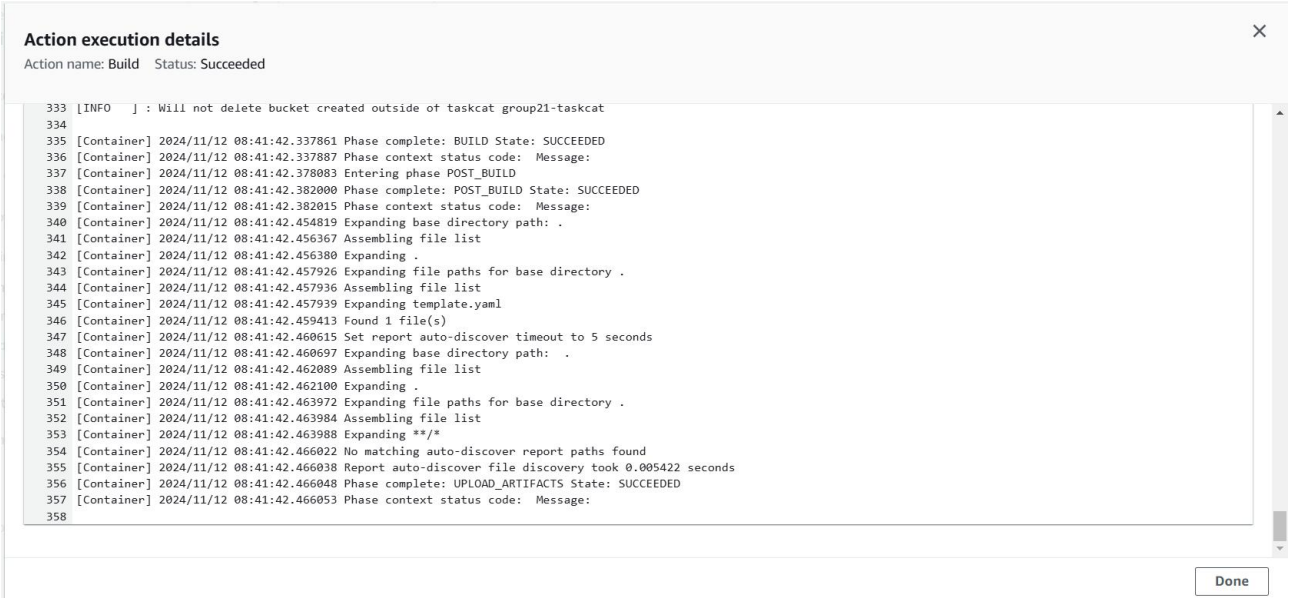


Ở Build Stage, cfn-lint đã scan file template.yaml và output 'Success'

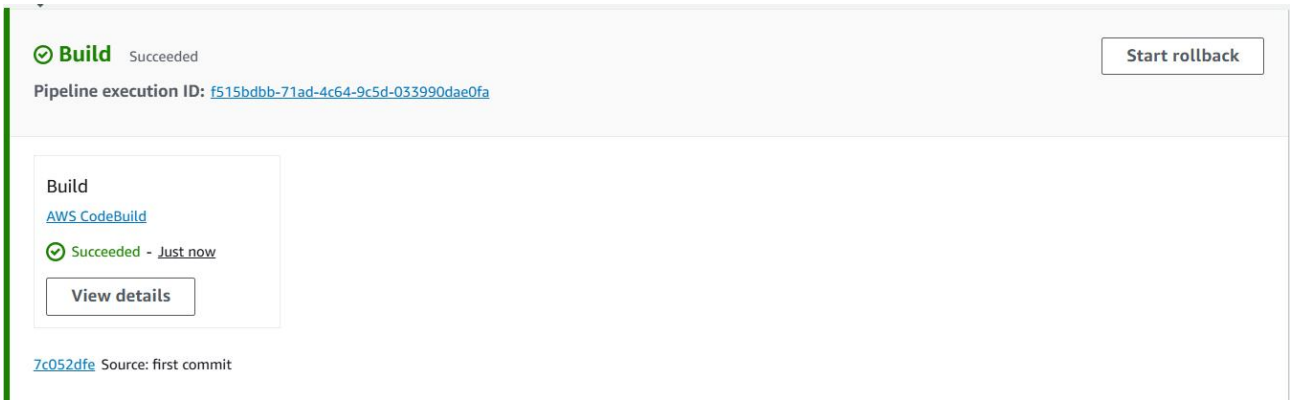


Tương tự đối với Taskcat, chạy lên 'taskcat test run' và output 'Success'

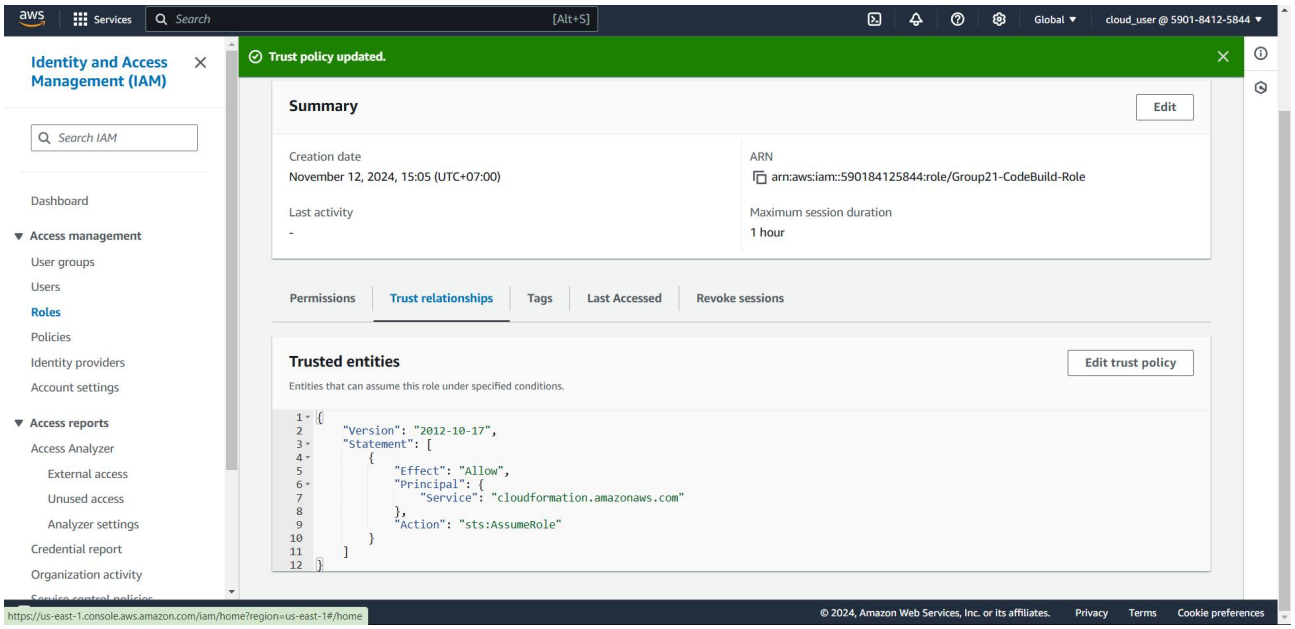




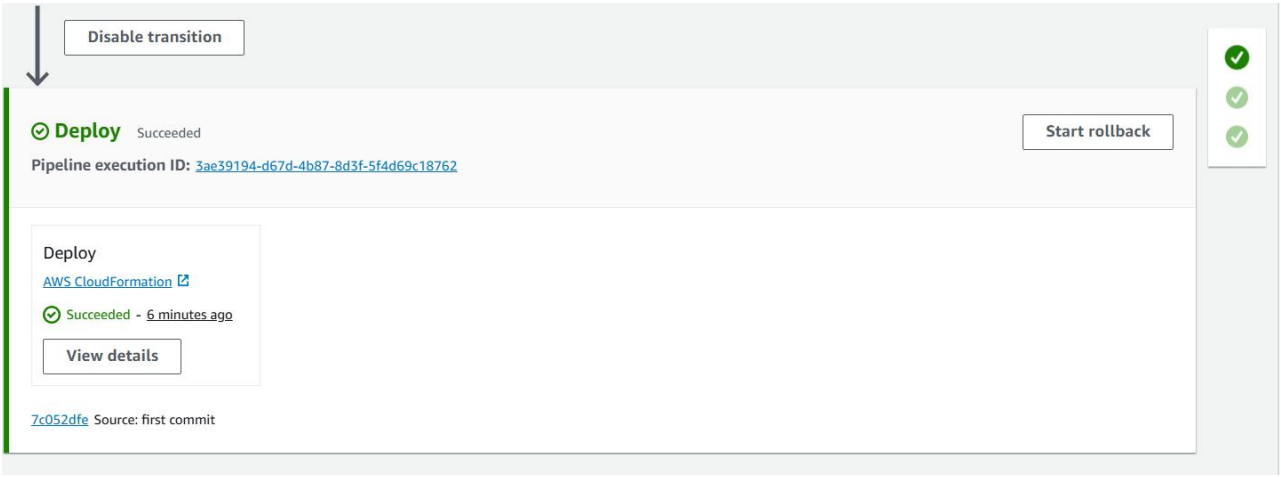
Build stage chạy thành công.



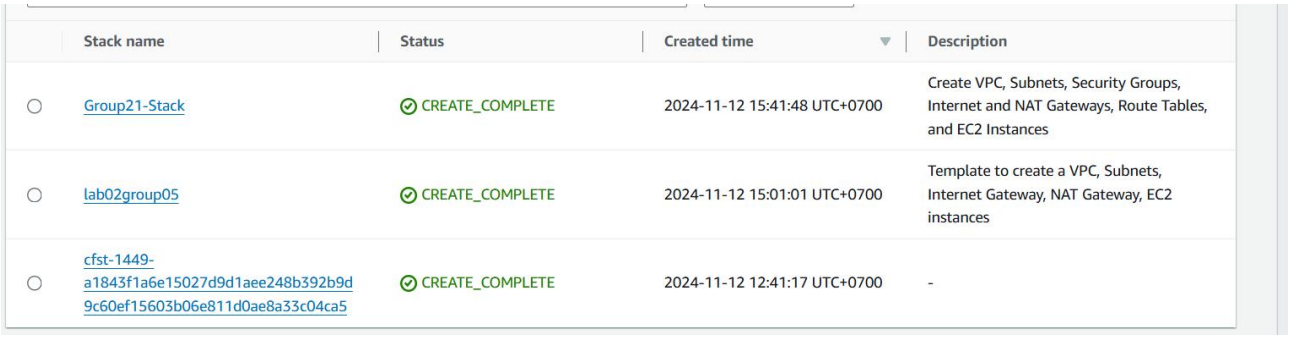
Khi chạy đến Deploy stage thì phải update lại trust policy:



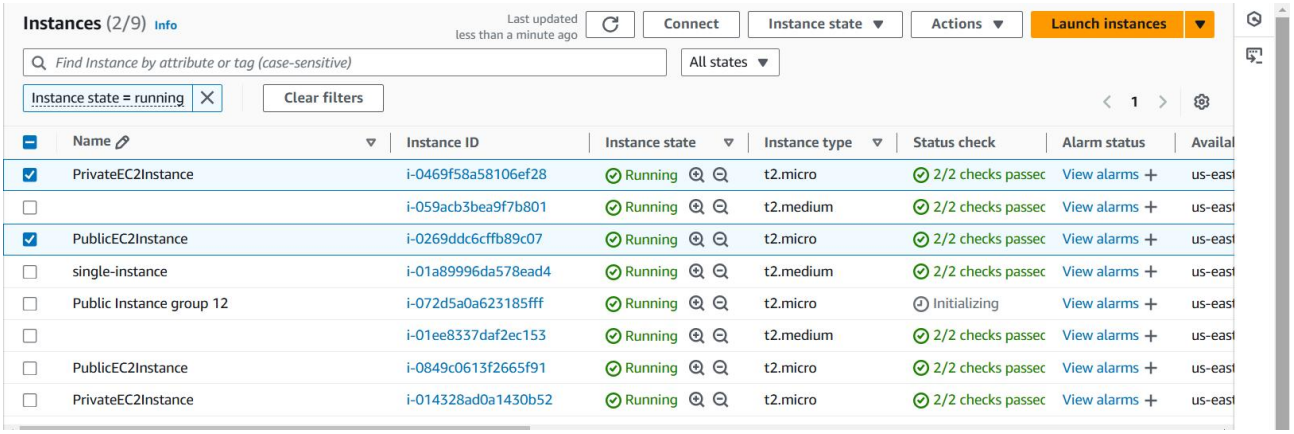
Deploy stage chạy thành công!



Các Cloudformation stack đã được khởi tạo.

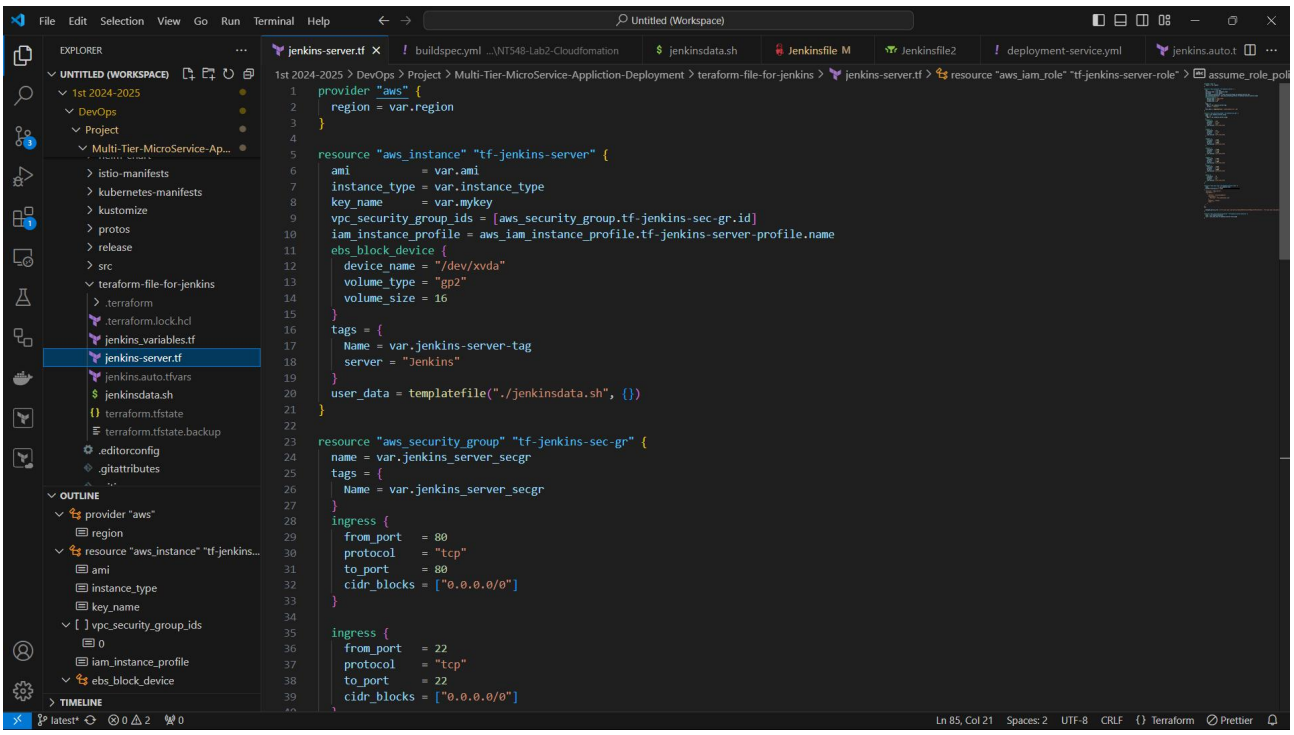


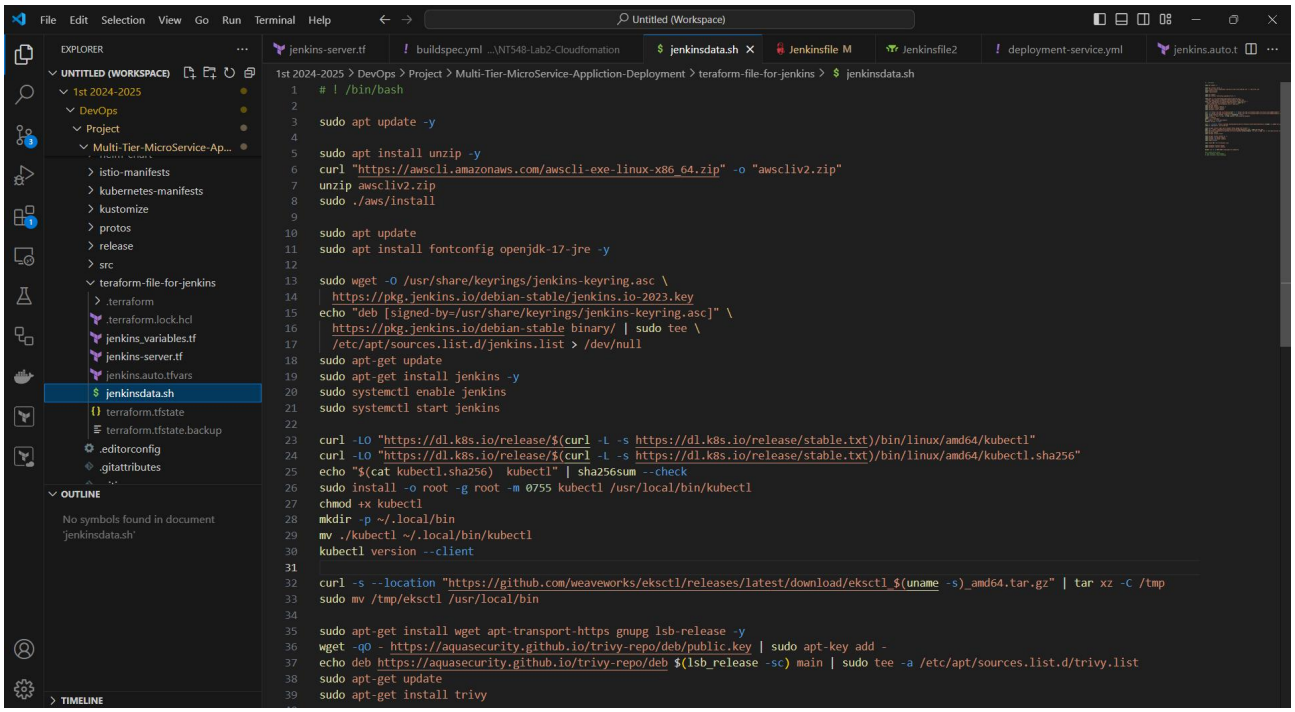
Stack name	Status	Created time	Description
Group21-Stack	CREATE_COMPLETE	2024-11-12 15:41:48 UTC+0700	Create VPC, Subnets, Security Groups, Internet and NAT Gateways, Route Tables, and EC2 Instances
lab02group05	CREATE_COMPLETE	2024-11-12 15:01:01 UTC+0700	Template to create a VPC, Subnets, Internet Gateway, NAT Gateway, EC2 instances
cfst-1449-a1843f1a6e15027d9d1aee248b392b9d9c60ef15603b06e811d0ae8a33c04ca5	CREATE_COMPLETE	2024-11-12 12:41:17 UTC+0700	-

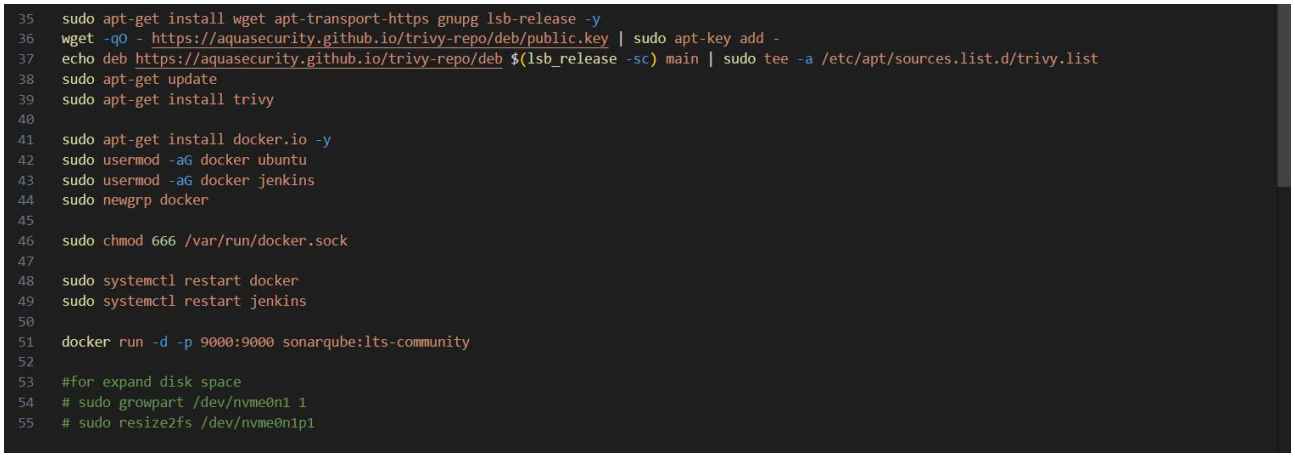


3. Câu 3

Đầu tiên, tạo một Jenkins Server bằng terraform, sau đó tải các package cần thiết để triển khai Docker, Kubernetes, EKS, Jenkins, Trivy và SonarQube.

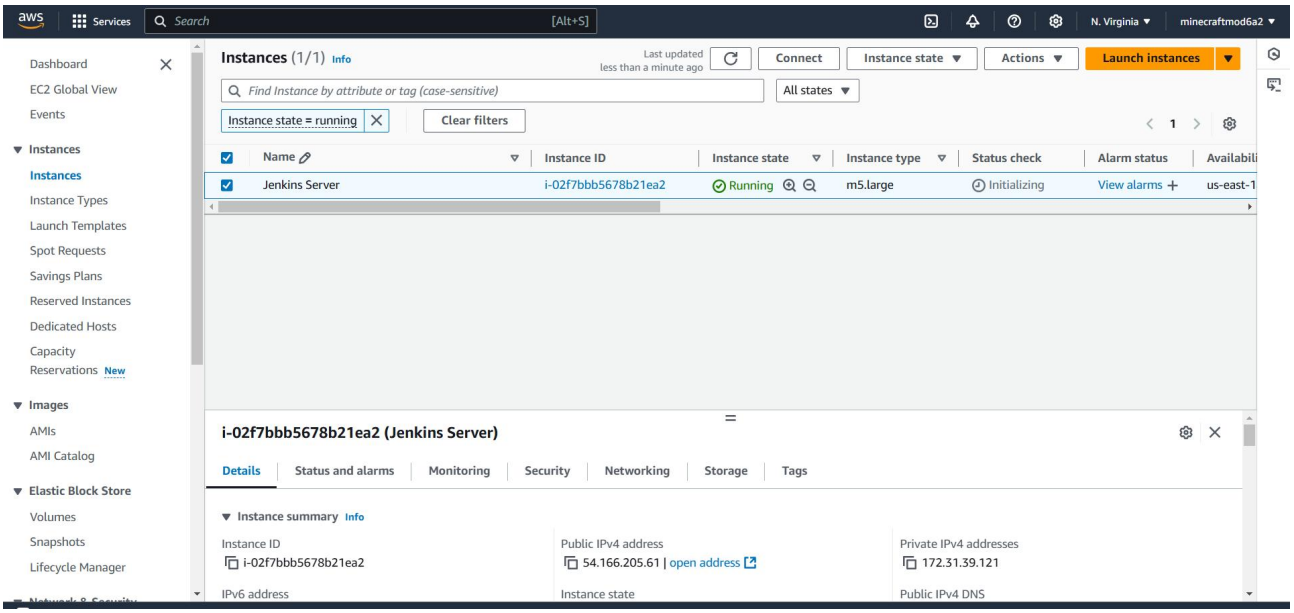




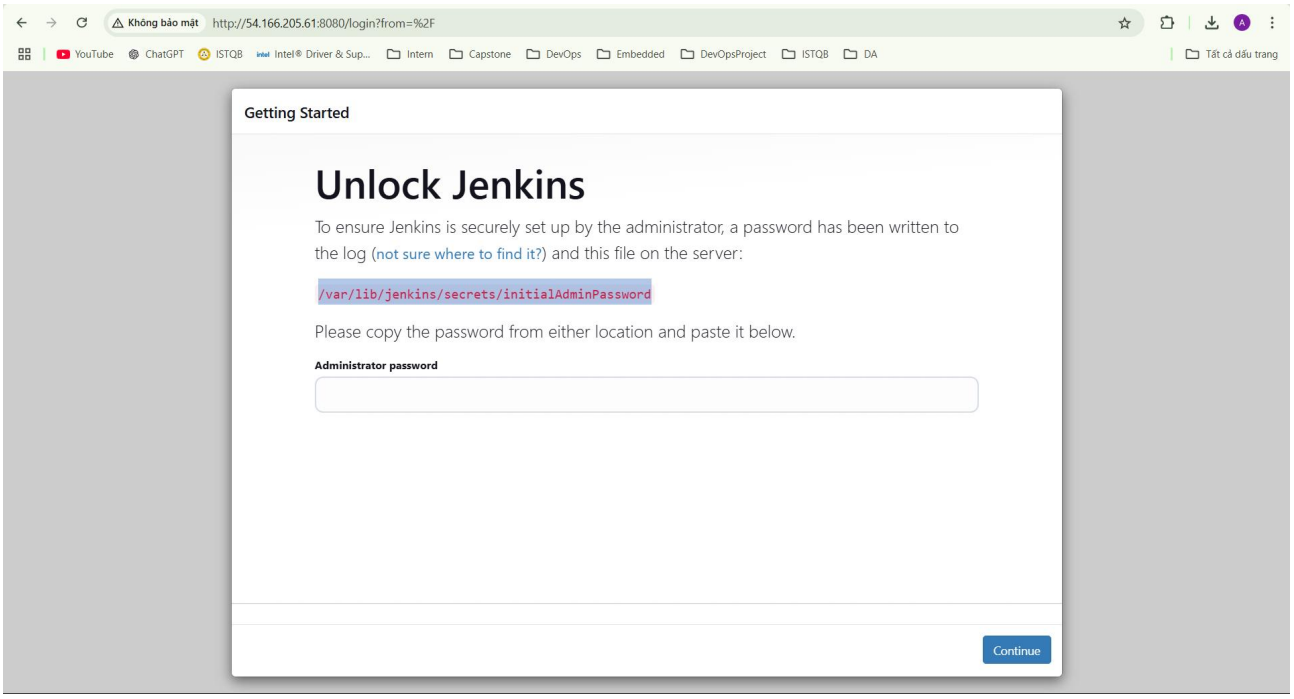


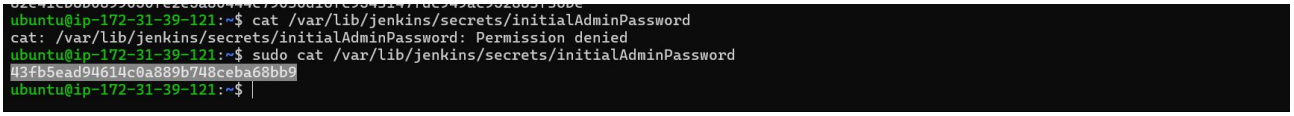
Lần lượt gõ terraform init, terraform plan và terraform apply.

Triển khai Jenkins Server thành công, IP public là 54.166.205.61.

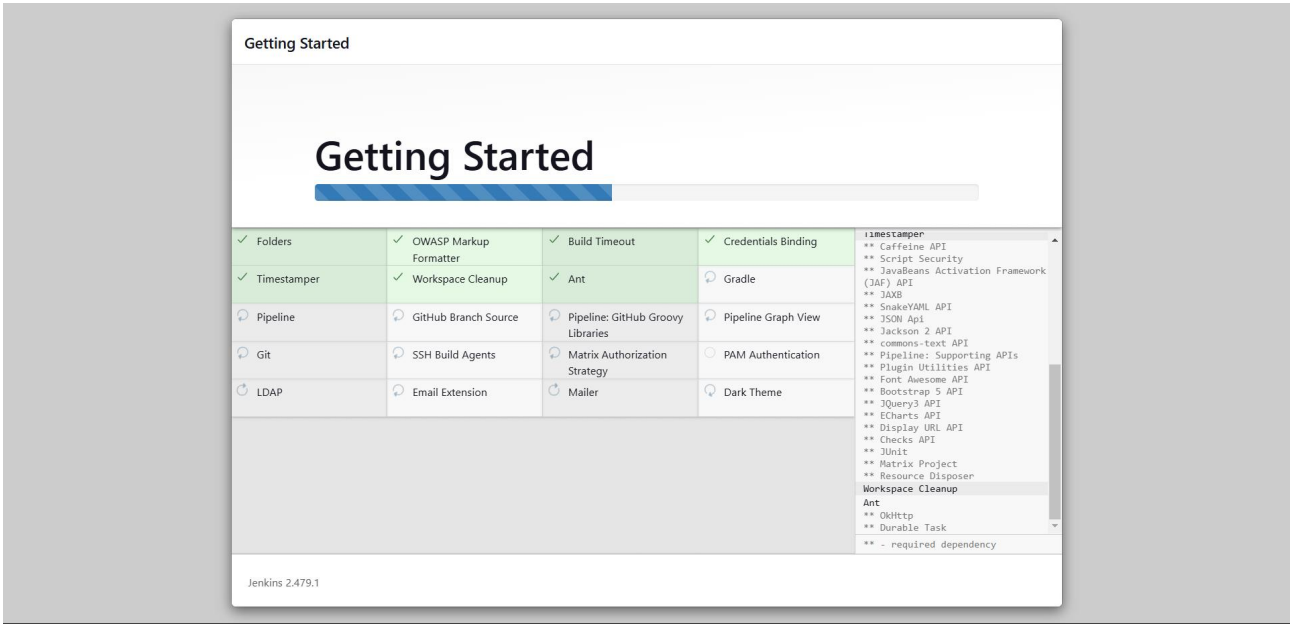


Truy cập vào Jenkins bằng địa chỉ <PublicIP>:8080, và vào path bên dưới để lấy mật khẩu.

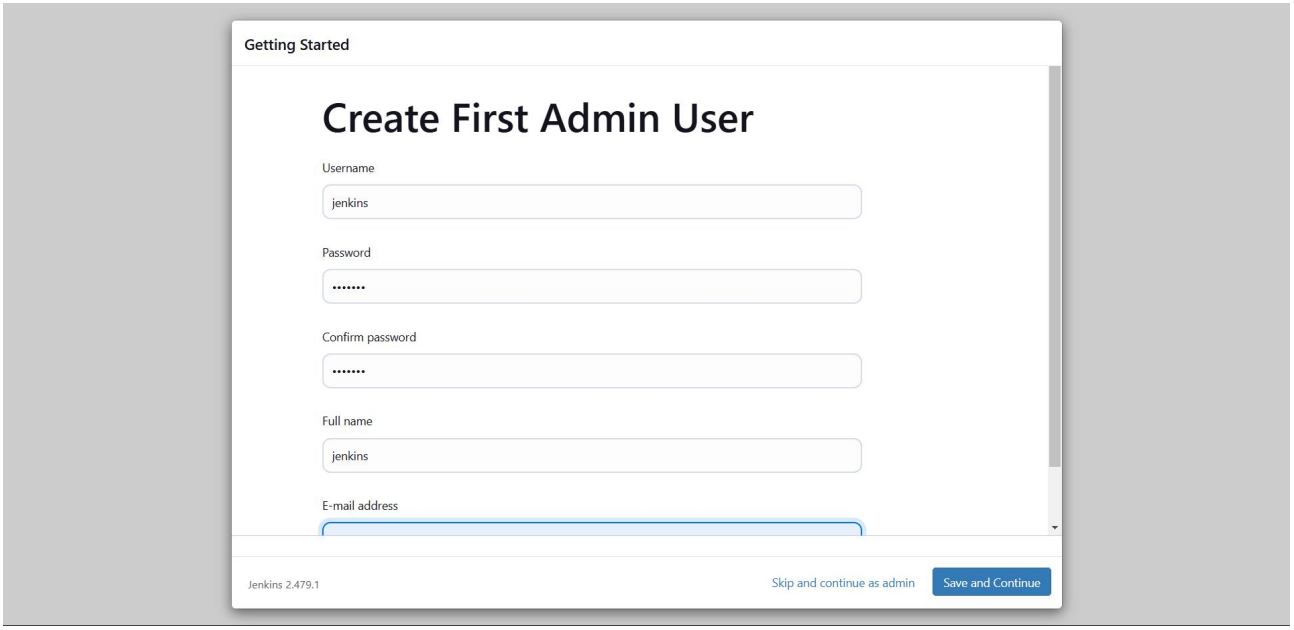


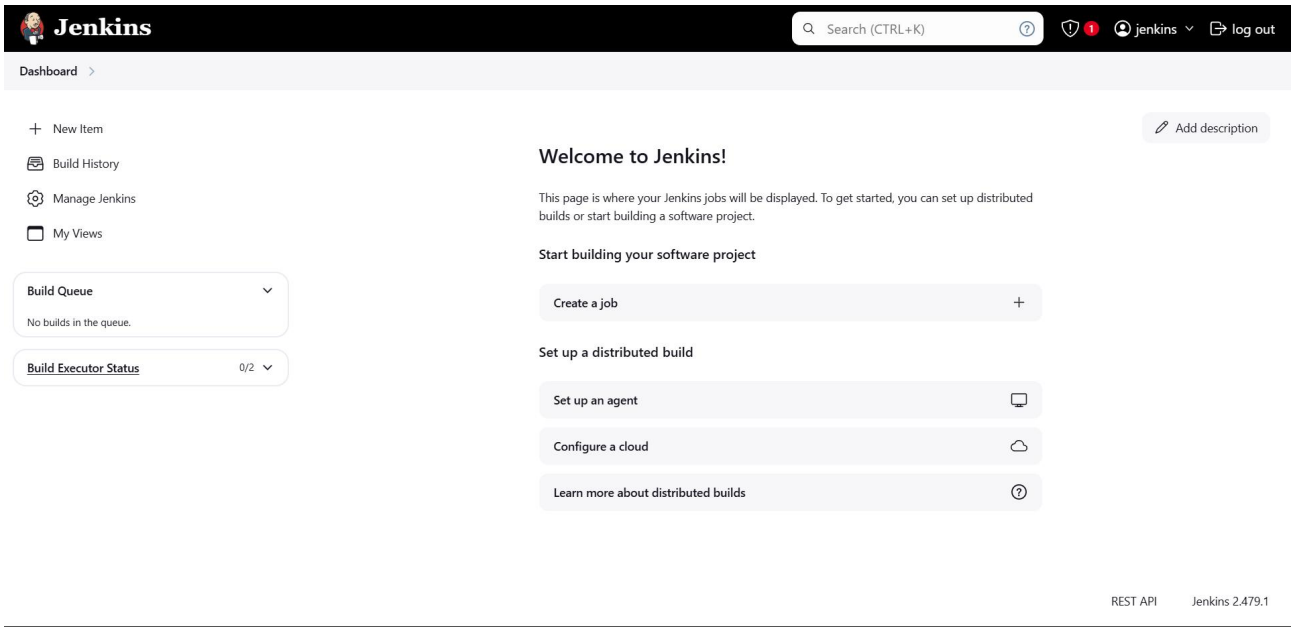


Tải về một số plugins.



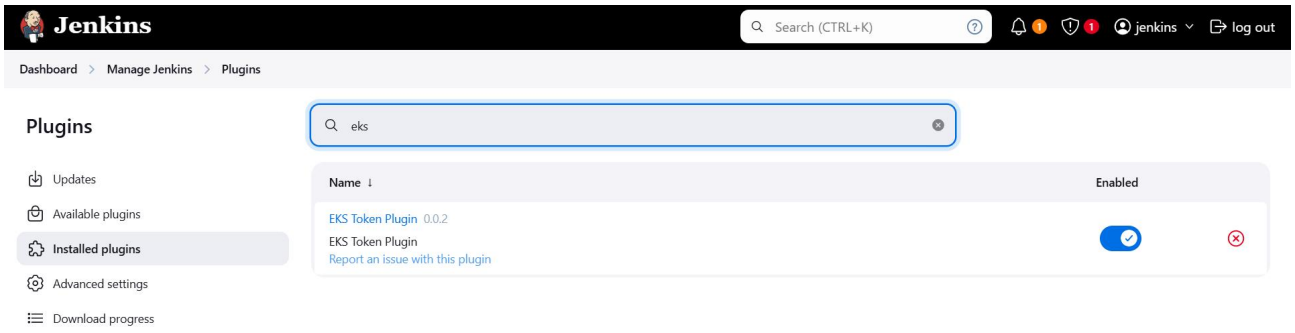
Tạo Admin User.



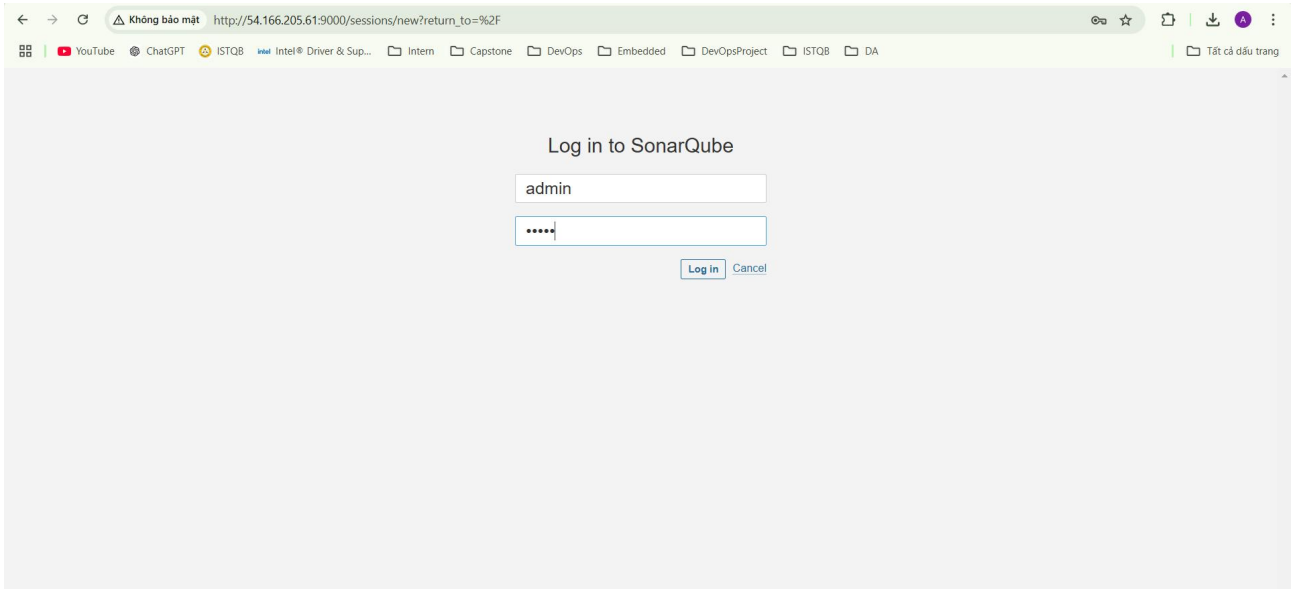


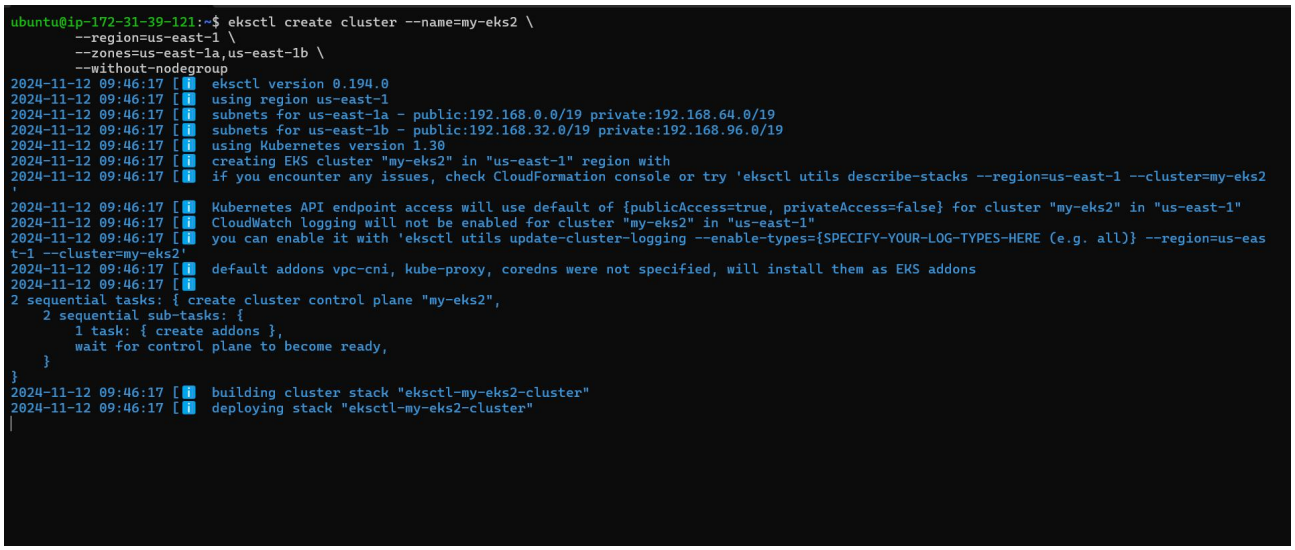
Tải về một số Plugins.

☑	SonarQube Scanner 2.17.2 External Site/Tool Integrations · Build Reports This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.	8 mo 26 days ago
☑	Docker 1.7.0 Cloud Providers · Cluster Management · docker This plugin integrates Jenkins with Docker	28 days ago
☑	Docker Commons 445.v6b_646c962a_94 Library plugins (for use by other plugins) · docker Provides the common shared functionality for various Docker-related plugins.	4 days 17 hr ago
☑	Docker Pipeline 580.vc0c340686b_54 pipeline · DevOps · Deployment · docker Build and use Docker containers from pipelines.	5 mo 24 days ago
☑	CloudBees Docker Build and Publish 1.4.0 Build Tools · docker This plugin enables building Dockerfile based projects, as well as publishing of the built images/repos to the docker registry.	2 yr 2 mo ago
☑	Kubernetes Client API 6.10.0-240.v57880ce8b_0b_2 kubernetes · Library plugins (for use by other plugins) Kubernetes Client API plugin for use by other Jenkins plugins.	9 mo 20 days ago
☑	Kubernetes Credentials 190.v03c305394deb_ kubernetes · credentials Common classes for Kubernetes credentials	1 mo 25 days ago
☑	Kubernetes 4296.v20a_7e4d77cf6 Cloud Providers · Cluster Management · kubernetes · Agent Management This plugin integrates Jenkins with Kubernetes	6 days 23 hr ago
☑	Kubernetes CLI 1.12.1 kubernetes Configure kubectl for Kubernetes	1 yr 2 mo ago
☑	Kubernetes Credentials Provider 1.262.v2670ef7ea_0c5 kubernetes · credentials Provides a read only credentials store backed by Kubernetes.	8 mo 16 days ago



Đăng nhập vào SonarQube với tài khoản admin:admin.

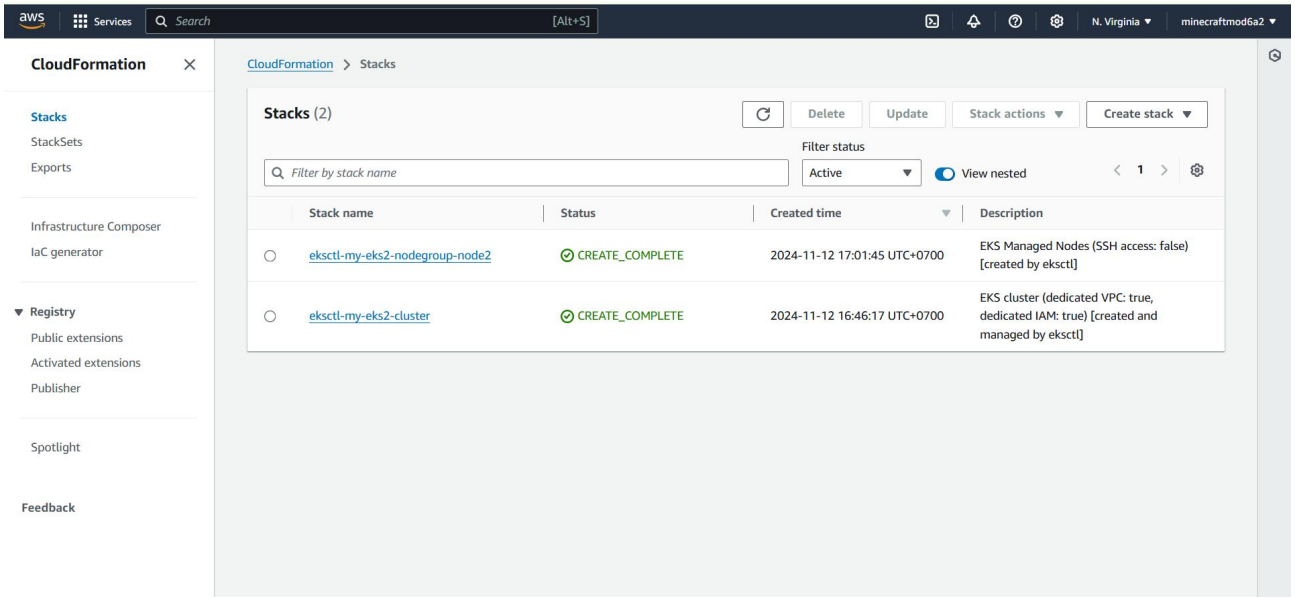


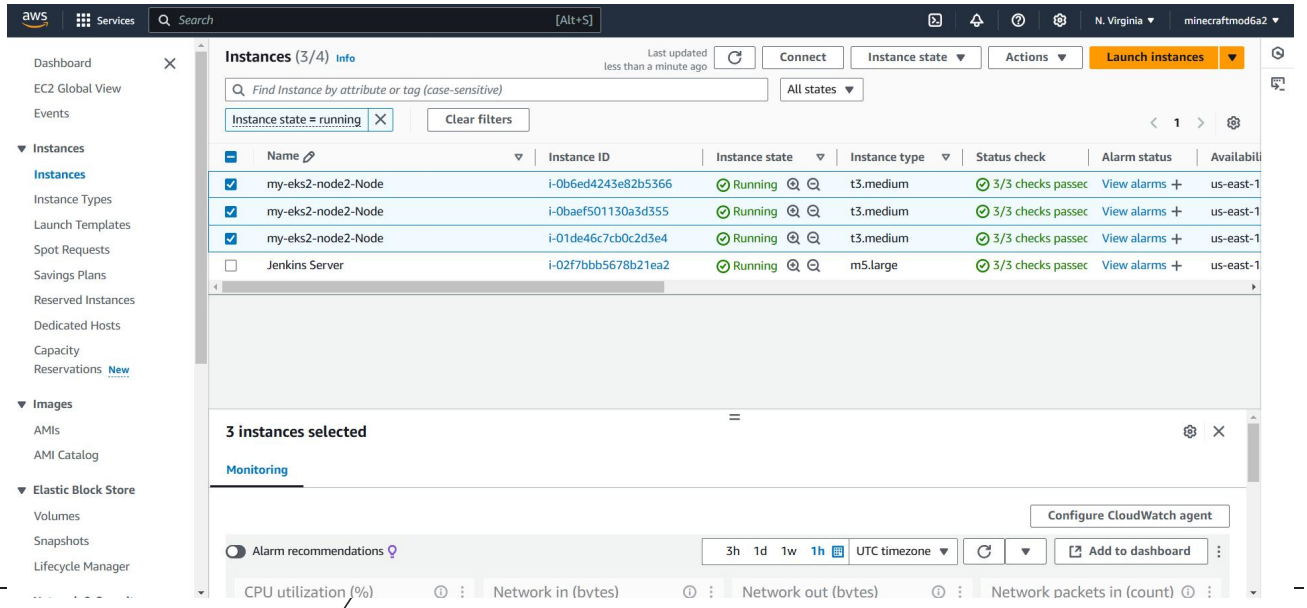


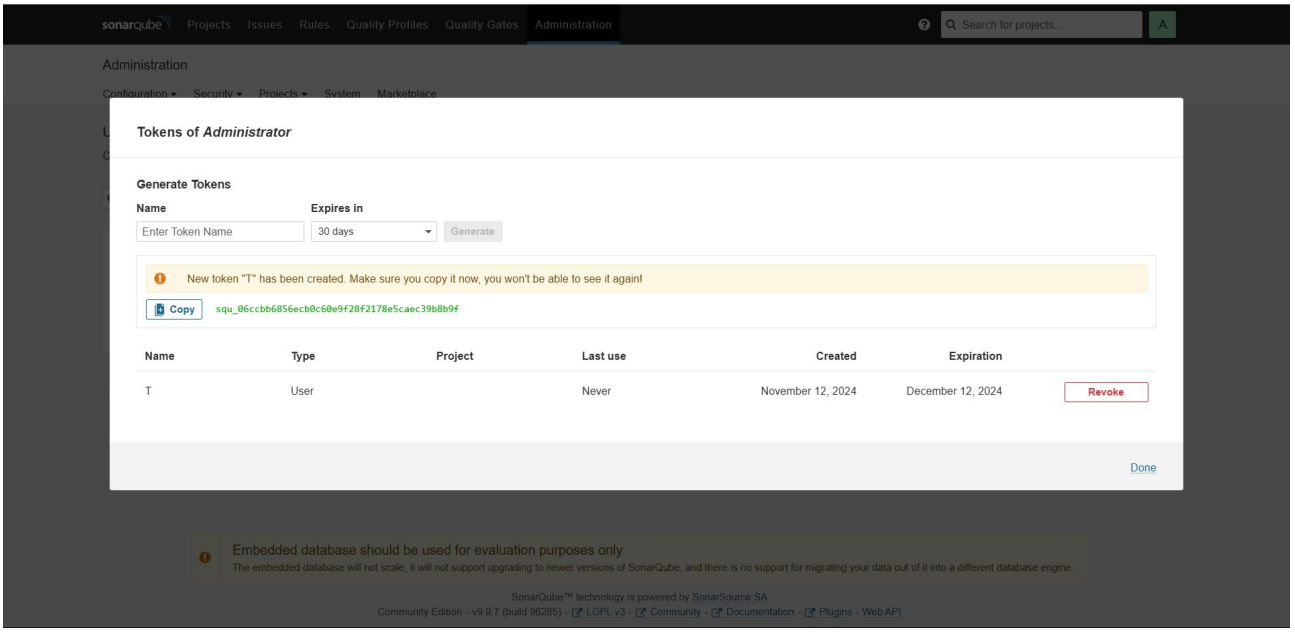
Tạo một Cluster tên ‘my-eks2’

```
ubuntu@ip-172-31-39-121:~$ eksctl utils associate-iam-oidc-provider \
 --region us-east-1 \
 --cluster my-eks2 \
 --approve
2024-11-12 10:01:37 [ℹ]  will create IAM Open ID Connect provider for cluster "my-eks2" in "us-east-1"
2024-11-12 10:01:37 [✔]  created IAM Open ID Connect provider for cluster "my-eks2" in "us-east-1"
ubuntu@ip-172-31-39-121:~$ eksctl create nodegroup --cluster=my-eks2 \
   --region=us-east-1 \
   --name=node2 \
   --node-type=t3.medium \
   --nodes=3 \
   --nodes-min=2 \
   --nodes-max=3 \
   --node-volume-size=20 \
   --ssh-public-key=mykey \
   --managed \
   --asg-access \
   --external-dns-access \
   --full-ecr-access \
   --appmesh-access \
   --alb-ingress-access
2024-11-12 10:01:44 [ℹ]  will use version 1.30 for new nodegroup(s) based on control plane version
2024-11-12 10:01:45 [ℹ]  nodegroup "node2" will use "" [AmazonLinux2/1.30]
2024-11-12 10:01:45 [ℹ]  1 nodegroup (node2) was included (based on the include/exclude rules)
2024-11-12 10:01:45 [ℹ]  will create a CloudFormation stack for each of 1 managed nodegroups in cluster "my-eks2"
2024-11-12 10:01:45 [ℹ]
2 sequential tasks: { fix cluster compatibility, 1 task: { 1 task: { create managed nodegroup "node2" } }
}
2024-11-12 10:01:45 [ℹ]  checking cluster stack for missing resources
2024-11-12 10:01:45 [ℹ]  cluster stack has all required resources
2024-11-12 10:01:45 [ℹ]  building managed nodegroup stack "eksctl-my-eks2-nodegroup-node2"
2024-11-12 10:01:45 [ℹ]  deploying stack "eksctl-my-eks2-nodegroup-node2"
2024-11-12 10:01:45 [ℹ]  waiting for CloudFormation stack "eksctl-my-eks2-nodegroup-node2"
2024-11-12 10:02:15 [ℹ]  waiting for CloudFormation stack "eksctl-my-eks2-nodegroup-node2"
2024-11-12 10:03:05 [ℹ]  waiting for CloudFormation stack "eksctl-my-eks2-nodegroup-node2"
2024-11-12 10:04:26 [ℹ]  waiting for CloudFormation stack "eksctl-my-eks2-nodegroup-node2"
2024-11-12 10:04:26 [ℹ]  no tasks
2024-11-12 10:04:26 [✔]  created 0 nodegroup(s) in cluster "my-eks2"
```

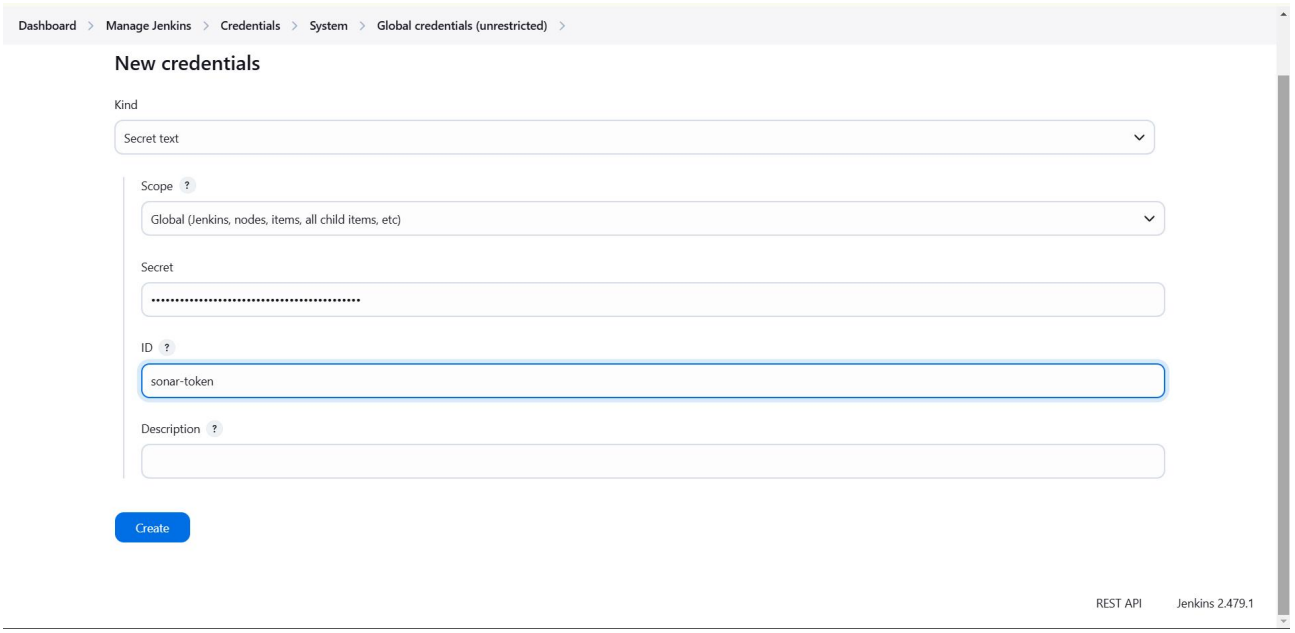
Liên kết OIDC với với Cluster ‘my-eks2’ và tạo một nodegroup cho cluster ‘my-eks2’, gồm 3 node EC2 (t3.medium) với cấu hình chi tiết bên trên.



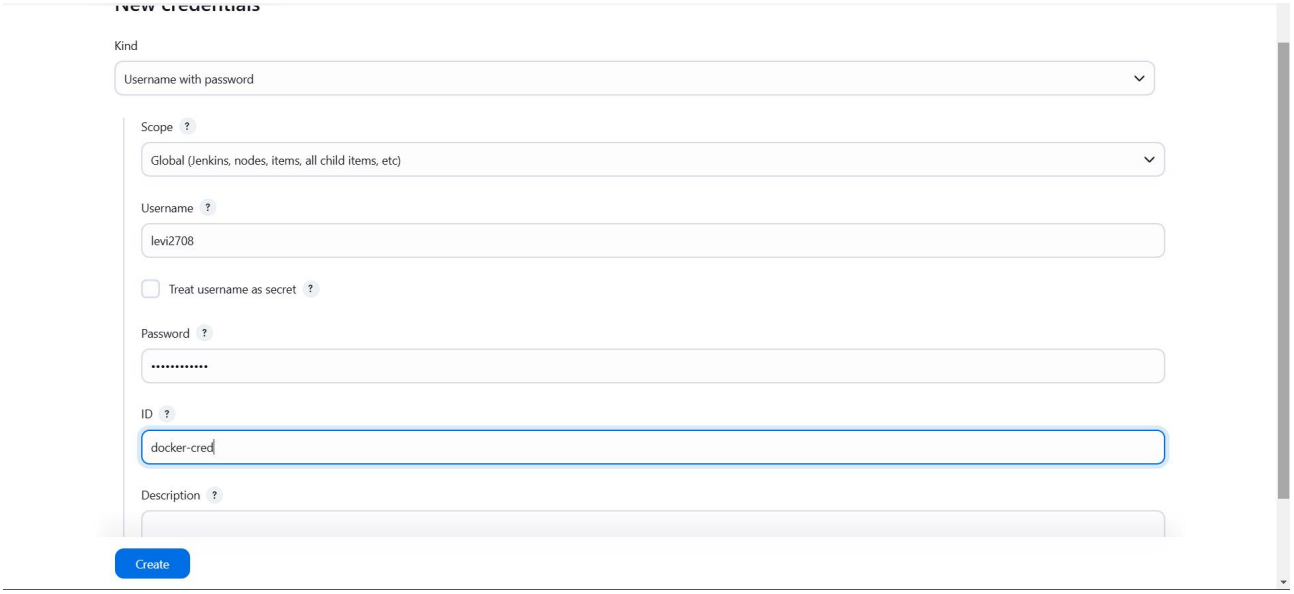




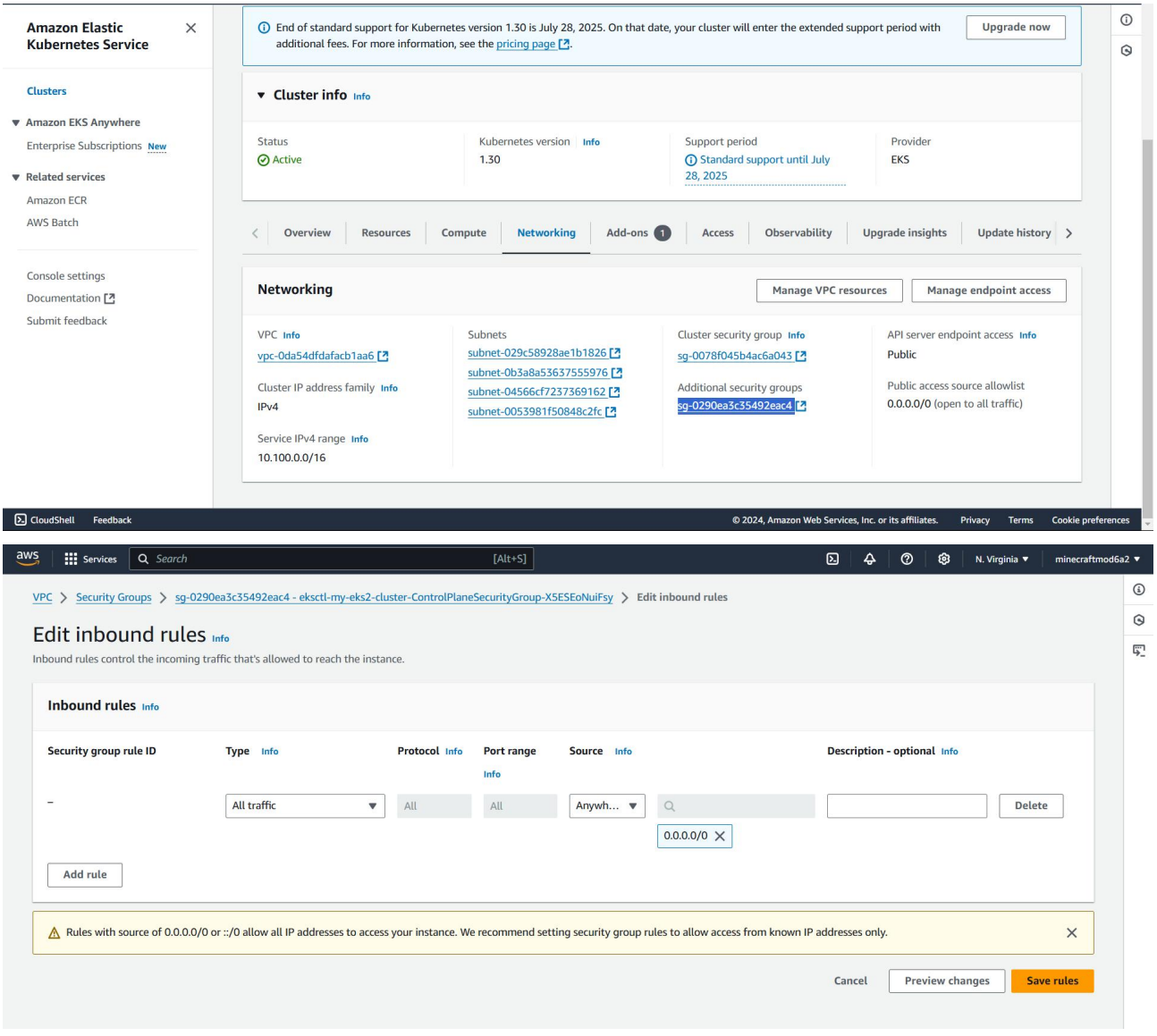
Quay lại với SonarQube, chọn vào Admininstration -> Token -> Generate Token.



Vào Jenkins, Manage Jenkins -> Credentials -> Global Credentials, tạo một secret text với token vừa tạo, đặt id là ‘sonar-token’.



Tương tự như trên, tạo một Username with password với tài khoản mật khẩu của Docker.



Trước khi vào bước tiếp theo, ta cần chỉnh lại inbound rules ở Additional security group của 'my-eks2'.

Tạo một namespace tên là 'webapps'

```
ubuntu@ip-172-31-39-121:~$ kubectl create namespace webapps
namespace/webapps created
ubuntu@ip-172-31-39-121:~$ kubectl get namespace
NAME              STATUS   AGE
default           Active   49m
kube-node-lease   Active   49m
kube-public       Active   49m
kube-system       Active   49m
webapps           Active   16s
ubuntu@ip-172-31-39-121:~$
```

Tạo một ServiceAccount tên là Jenkins cho namespace webapps.

```
ubuntu@ip-172-31-39-121:~$ nano sa.yaml
ubuntu@ip-172-31-39-121:~$ nano sa.yaml
ubuntu@ip-172-31-39-121:~$
```

ubuntu@ip-172-31-39-121: ~

```
  GNU nano 7.2
apiVersion: v1
kind: ServiceAccount
metadata:
  name: jenkins
  namespace: webapps
```

```
ubuntu@ip-172-31-39-121:~$ kubectl apply -f sa.yaml
serviceaccount/jenkins created
ubuntu@ip-172-31-39-121:~$
```

Tạo một role tên là ‘app-role’ cho namespace webapps.

Các rules bao gồm:

Các API mặc định của Kubernetes cùng với vài API khác như apps, autoscaling, …

Danh sách các tài nguyên mà Role có thể truy cập như pods, configmaps, …

Danh sách các hành động Role thực hiện như get, list, create, …

```
ubuntu@ip-172-31-39-121:~$ nano rol.yaml
ubuntu@ip-172-31-39-121:~$
```

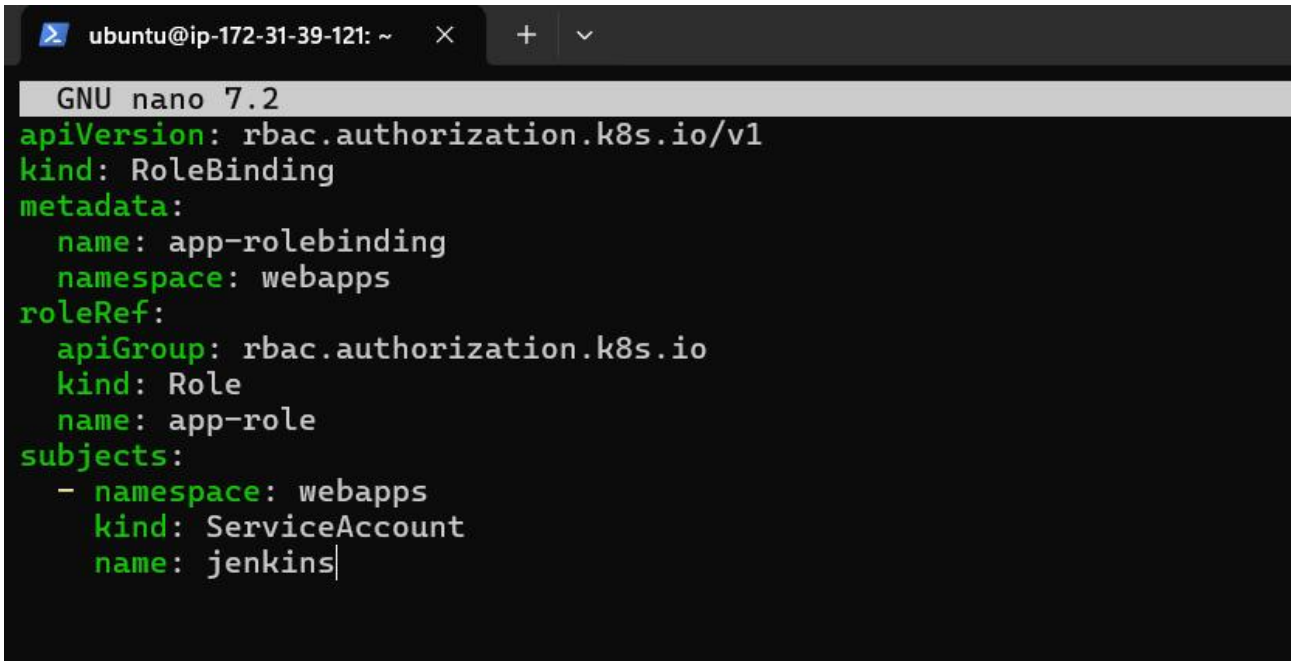
```
ubuntu@ip-172-31-39-121: ~
 GNU nano 7.2                                  rol.yaml *
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: app-role
  namespace: webapps
rules:
  - apiGroups:
    - ""
    - apps
    - autoscaling
    - batch
    - extensions
    - policy
    - rbac.authorization.k8s.io
    resources:
    - pods
    - configmaps
    - deployments
    - daemonsets
    - componentstatuses
    - events
    - endpoints
    - horizontalpodautoscalers
    - ingress
    - jobs
    - limitranges
    - namespaces
    - nodes
    - persistentvolumes
    - persistentvolumeclaims
    - resourcequotas
    - replicasets
    - replicationcontrollers
    - serviceaccounts
    - services
    verbs:
    - get
    - list
    - watch
    - create
^G Help      ^O Write Out  ^W Where Is   ^K Cut       ^T Execute   ^C Location  M-U Undo    M-A Set Mark  M-] To Bracket
^X Exit      ^R Read File  ^\ Replace    ^U Paste     ^J Justify   ^/ Go To Line M-E Redo    M-6 Copy      ^Q Where Was
```

```
ubuntu@ip-172-31-39-121:~$ kubectl apply -f rol.yaml
role.rbac.authorization.k8s.io/app-role created
ubuntu@ip-172-31-39-121:~$
```

Tạo một RoleBinding tên là app-rolebinding cho namespace webapps.

Gán quyền của Role ‘app-role’ cho ServiceAccount ‘jenkins’ trong namespace webapps, nghĩa là SA ‘jenkins’ sẽ có tất cả quyền được xác định trong ‘app-role’.

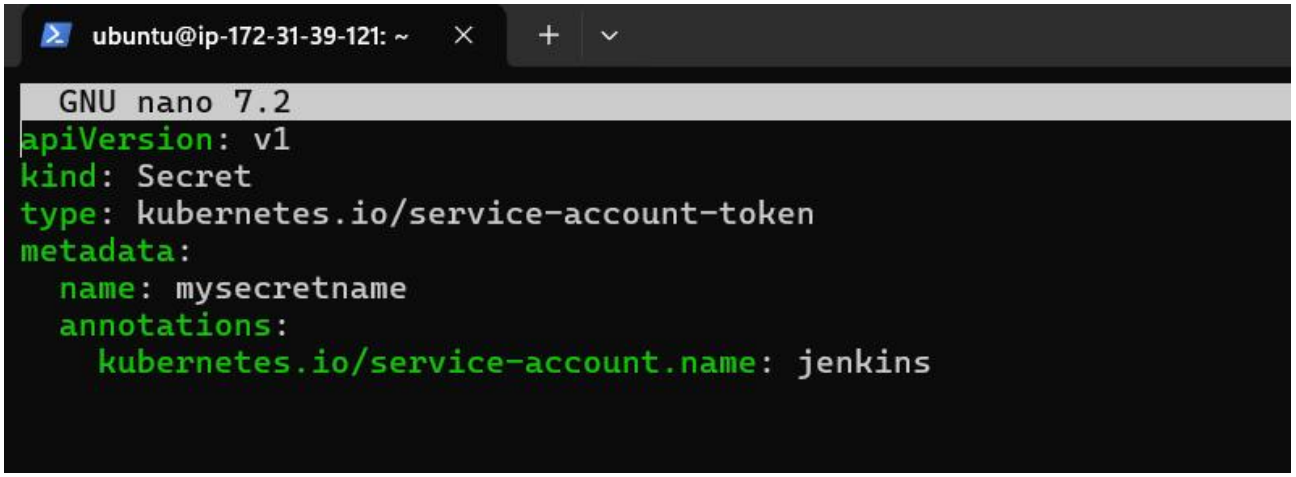
```
ubuntu@ip-172-31-39-121:~$ nano bind.yaml
```




```
ubuntu@ip-172-31-39-121:~$ kubectl apply -f bind.yaml
rolebinding.rbac.authorization.k8s.io/app-rolebinding created
ubuntu@ip-172-31-39-121:~$
```

Tạo một secret chứa token của ServiceAccount ‘jenkins’, tên là mysecretname.

```
ubuntu@ip-172-31-39-121:~$ nano sec.yaml
ubuntu@ip-172-31-39-121:~$
```

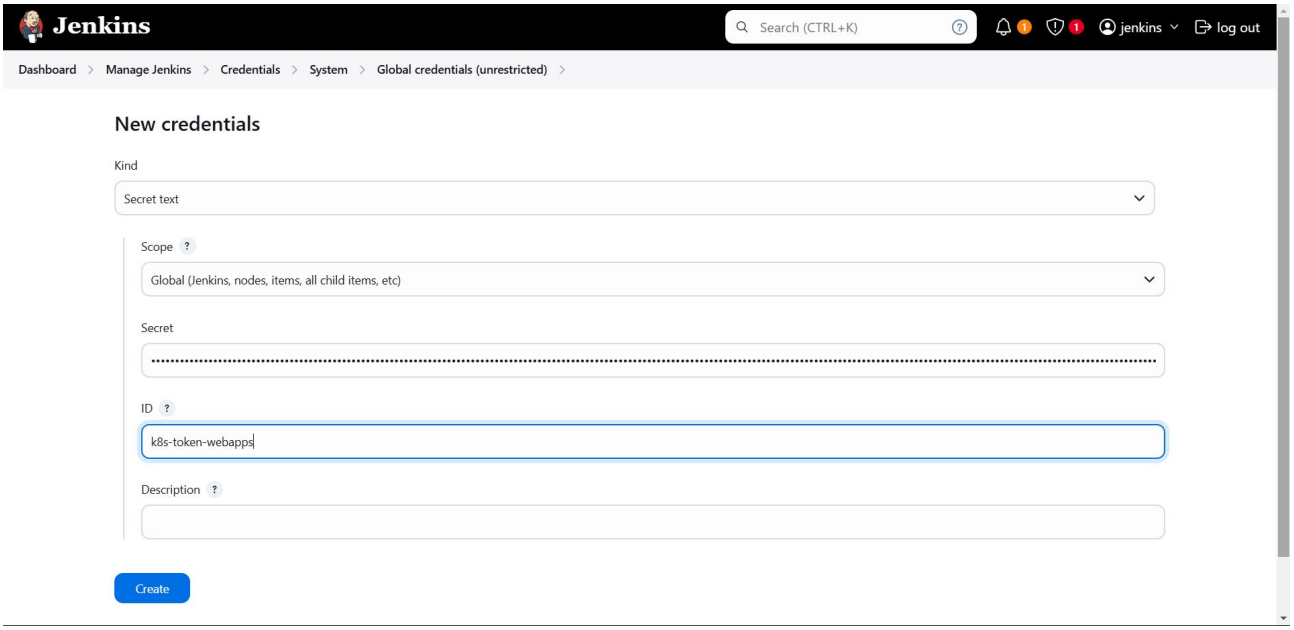



```
ubuntu@ip-172-31-39-121:~$ kubectl apply -f sec.yaml -n webapps
secret/mysecretname created
ubuntu@ip-172-31-39-121:~$
```

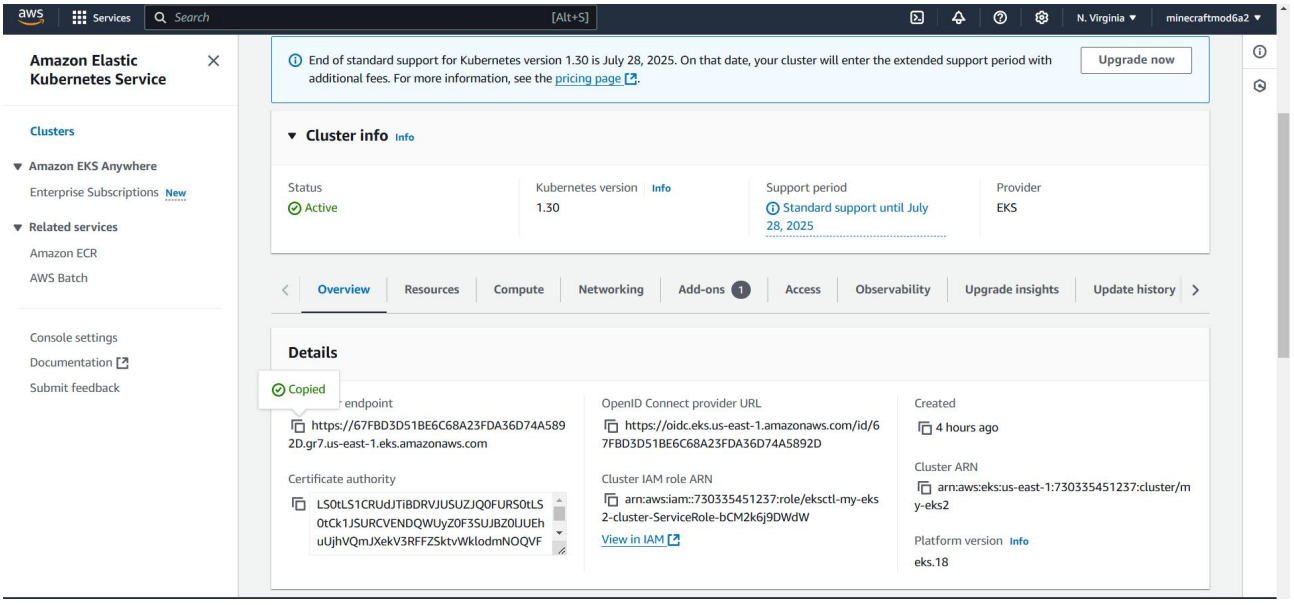
Ta có thể lấy token đó bằng cách dùng câu lệnh bên dưới:

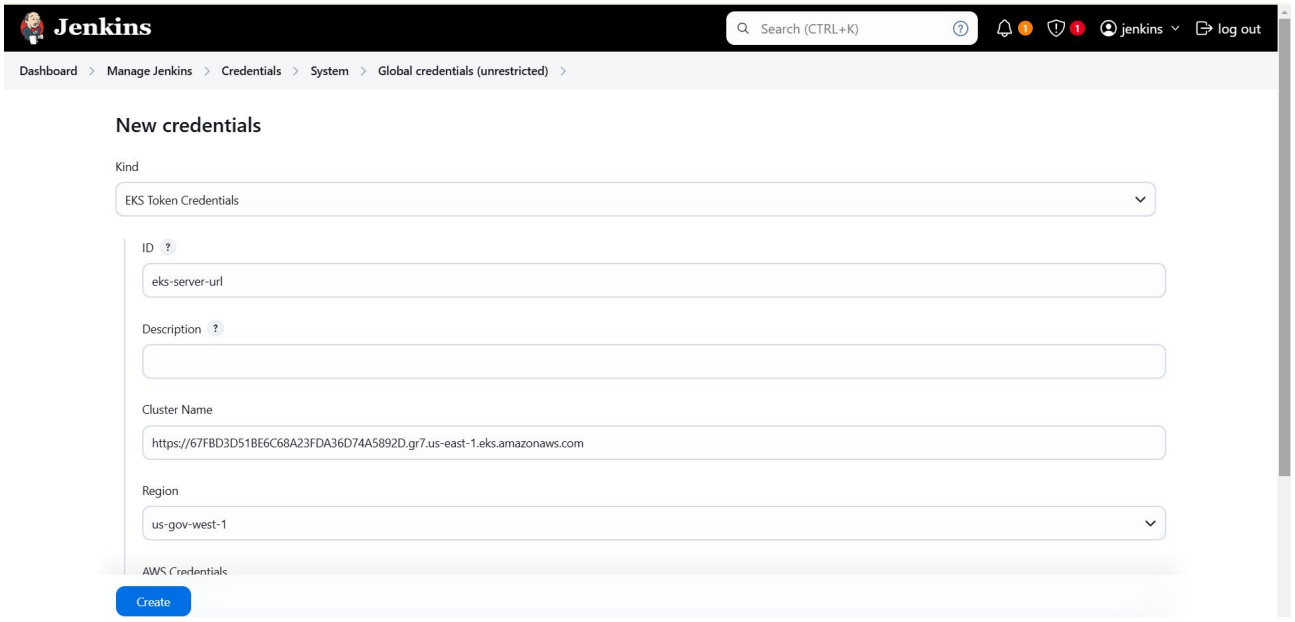
```
ubuntu@ip-172-31-39-121:~$ kubectl -n webapps describe secret mysecretname
Name:         mysecretname
Namespace:    webapps
Labels:       <none>
Annotations:  kubernetes.io/service-account.name: jenkins
              kubernetes.io/service-account.uid: e5cb9719-074f-4b76-bb0e-01e0ca4e6262

Type:  kubernetes.io/service-account-token

Data
====
ca.crt:     1107 bytes
namespace:  7 bytes
token:      eyJhbGciOiJSUzI1NiIsImtpZCI6IlA3cjRxT3AzSEhTRnYzV1g5aFBkYWo4Y1AzQVFZUGI2b0swSGpNUml1cXcifQ.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9uYW1lc3BhY2UiOiJ3ZWJhcHBzIiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZWNyZXQubmFtZSI6Im15c2VjcmV0bmFtZSIsImt1YmVybmV0ZXMuaW8vc2VydmljZWFjY291bnQvc2VydmljZS1hY2NvdW50Lm5hbWUiOiJqZW5raW5zIiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZXJ2aWNlLWFjY291bnQudWlkIjoiZTVjYjk3MTktMDc0Zi00Yjc2LWJiMGUtMDFlMGNhNGU2MjYyIiwic3ViIjoic3lzdGVtOnNlcnZpY2VhY2NvdW50OndlYmFwcHM6amVua2lucyJ9.hgSKrZZ6nl1Qf3QRcHZxih0HThQfdKkYgMhdFojBtr28ym8pUJ8iU-5EEodDVN2n8AS49WuVKf805QjV6-cyCygU-pvVpqv_zScHsJ4SliRsRnfweYiQmXlOTkw28XydcavS6q7Dd4Vhz82f4ZJizTVXEG4JTuzpFyAZy7JkUl7-B6BCVz1ZzdOf-MbgaTUIXGCMVu_EUWOmWMMlhDSc4x0eT0ZMupO0WWoODZFJaUNV3Z2KtUlfmAXt2FXWu56fkPl9w3cwbNp2KqulTFg6NP7tojTJvOrX-3aMtoQT3PXPFp337tUDfdy2aJJqYXk7RqWK-r7-fKnfV0hdz5ug3A
ubuntu@ip-172-31-39-121:~$
```

Lưu vào trong credentials của Jenkins, id là k8s-token-webapps.

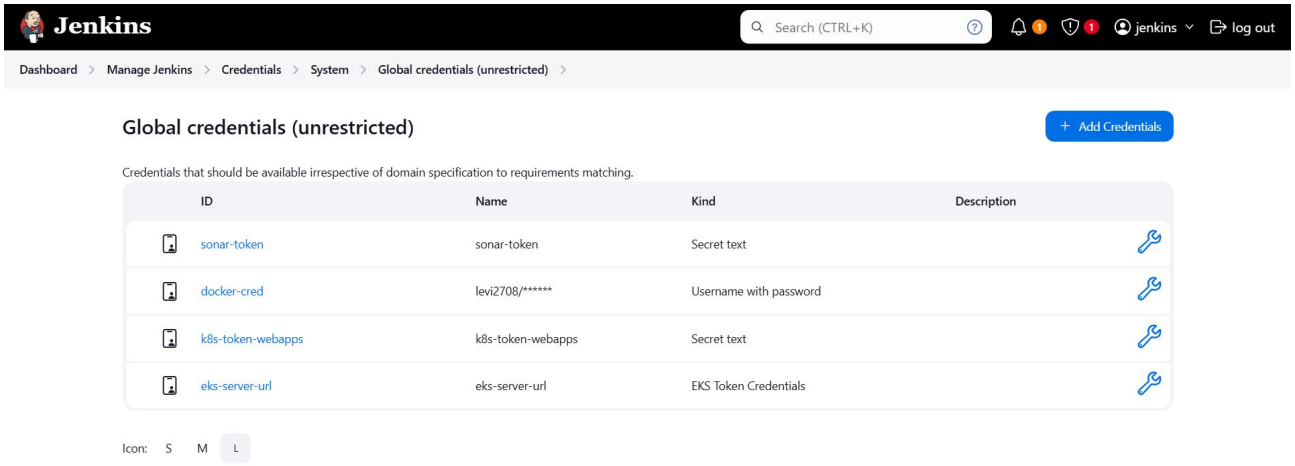


Copy đoạn URL của EKS server endpoint rồi lưu vào credentials của Jenkins.

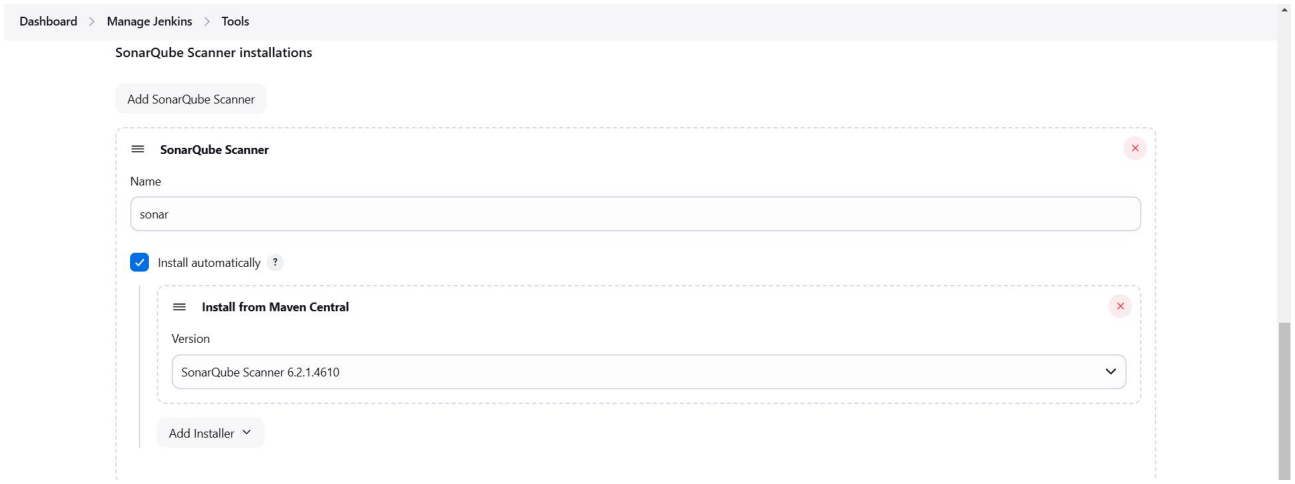




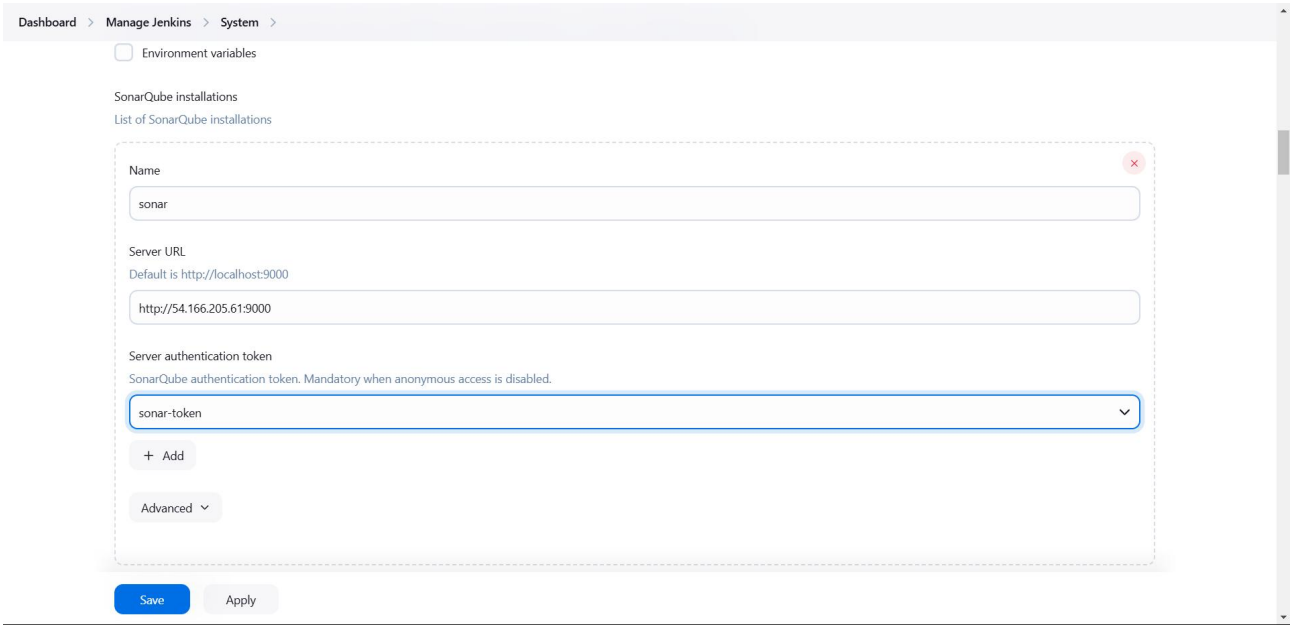
Tất cả các Credentials đã tạo



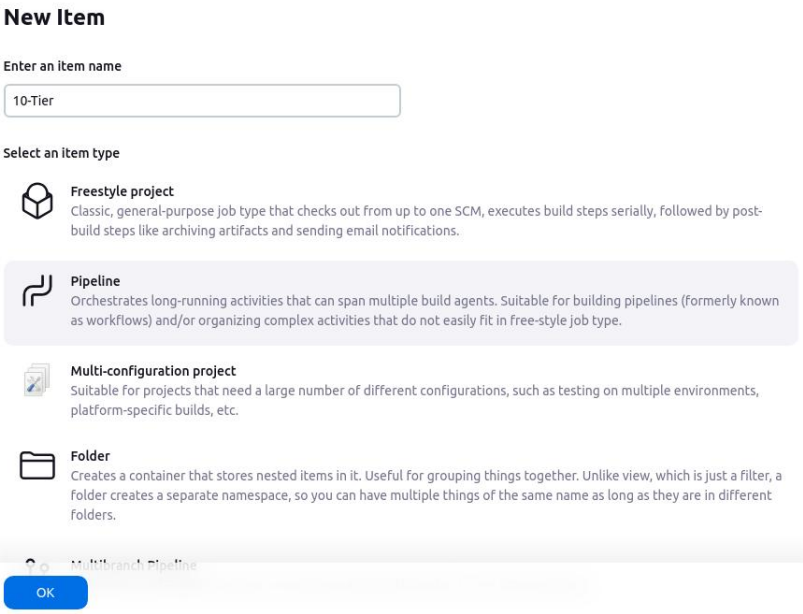
Tạo SonarQube Scanner bằng cách vào Manage Jenkins -> tools.



Vào Manage Jenkins -> System, sau đó tạo một Jenkins server với IP là <PublicIP>:9000, với token là ‘sonar-token’ đã tạo trước đó.



Tạo Pipeline tên là ’10-Tier’, các cấu hình đều để mặc định, sau đó paste Jenkinsfile vào phần Script.



Đoạn script sẽ checkout nhánh ‘latest’ từ Github repo đã cho. Sử dụng SonarQube để scan và trả về kết quả ở SonarQube server.

1st 2024-2025 > DevOps > Project > Multi-Tier-MicroService-Appliction-Deployment > Jenkinsfile2

```
pipeline {
    agent any
    environment{
        SCANNER_HOME= tool 'sonar-scanner'
    }
    stages {
        stage('git checkout') {
            steps {
                git branch: 'latest', url: 'https://github.com/just-vile/Multi-Tier-MicroService-Appliction-Deployment.git'
            }
        }
        stage('SonarQube') {
            steps {
                withSonarQubeEnv('sonar') {
                    sh '''
                    $SCANNER_HOME/bin/sonar-scanner -Dsonar.projectKey=10-Tier -Dsonar.ProjectName=10-Tier -Dsonar.java.binaries=.
                    '''
                }

            }

        }
```

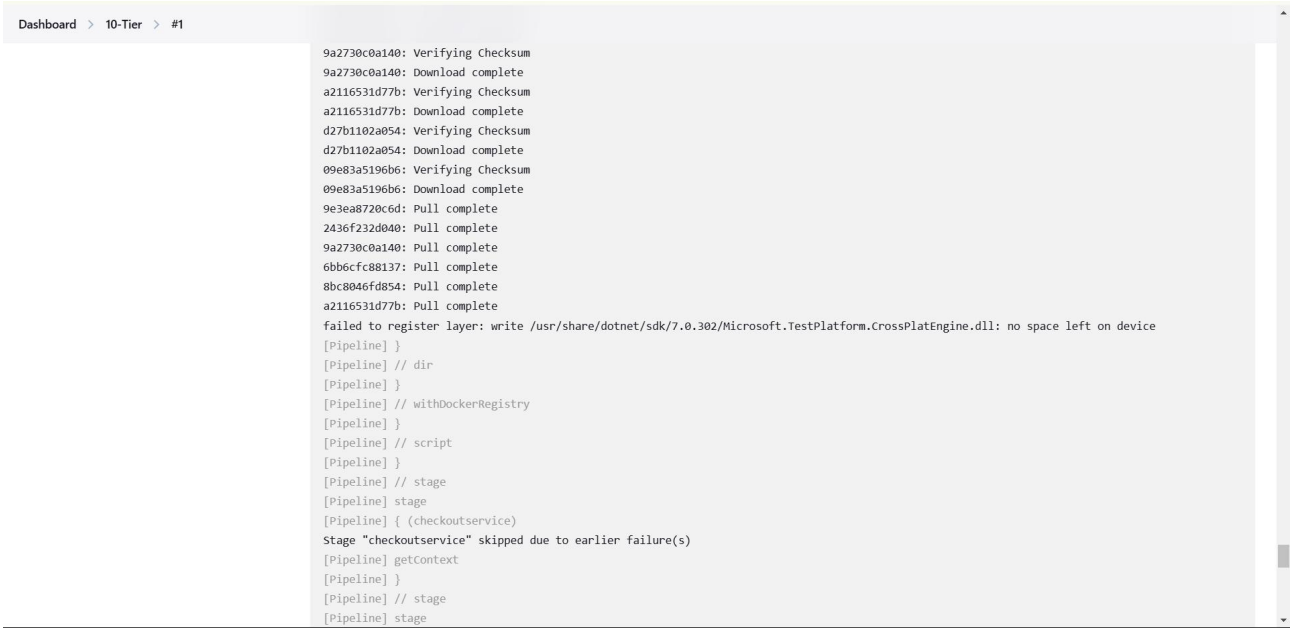
Kế tiếp sẽ sử dụng Docker để Containerize các service, sử dụng tài khoản đã lưu trước đó ở credentials của Jenkins.

```
        stage('adservice'){
            steps{
             script{
              withDockerRegistry(credentialsId: 'docker-cred', toolName: 'docker') {
                        dir('/var/lib/jenkins/workspace/10-Tier/src/adservice') {
                            sh 'docker build -t levi2708/adservice:latest .'
                            sh "docker push levi2708/adservice:latest"
                            sh "docker rmi levi2708/adservice:latest"
                        }
                    }
            }
            }
        }

        stage('cartservice'){
            steps{
             script{
              withDockerRegistry(credentialsId: 'docker-cred', toolName: 'docker') {
                        dir('/var/lib/jenkins/workspace/10-Tier/src/cartservice/src/') {
                            sh 'docker build -t levi2708/cartservice:latest .'
                            sh "docker push levi2708/cartservice:latest"
                            sh "docker rmi levi2708/cartservice:latest"
                        }
                    }
            }
            }
        }

        stage('checkoutservice'){
            steps{
             script{
              withDockerRegistry(credentialsId: 'docker-cred', toolName: 'docker') {
                        dir('/var/lib/jenkins/workspace/10-Tier/src/checkoutservice/') {
                            sh 'docker build -t levi2708/checkoutservice:latest .'
                            sh "docker push levi2708/checkoutservice:latest"
                            sh "docker rmi levi2708/checkoutservice:latest"
                        }
```

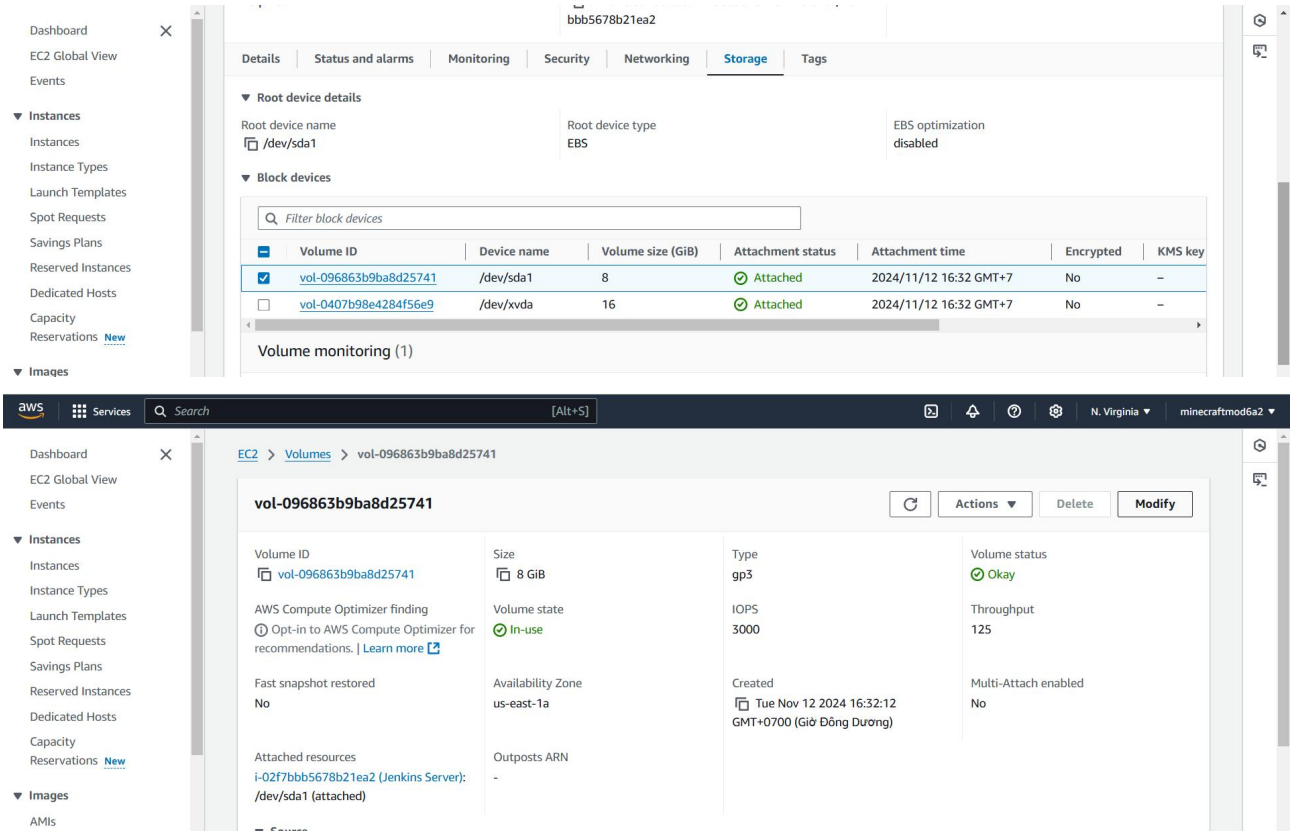
Ở stage cuối cùng, nó sẽ deploy tài nguyên lên cụm EKS.

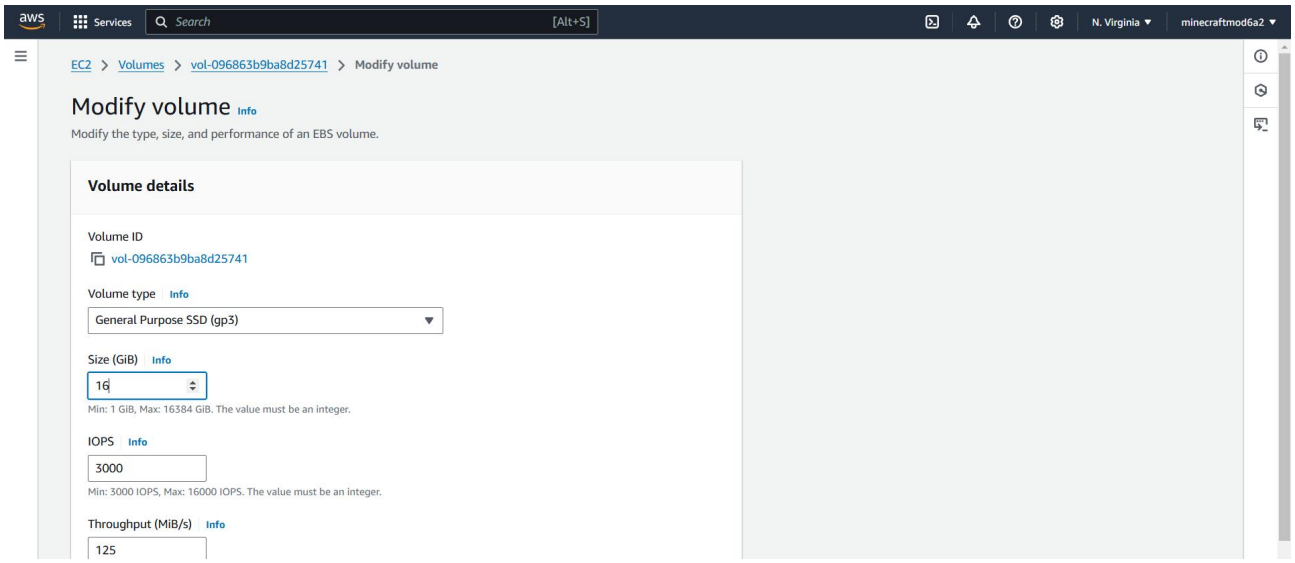
```
        stage('K8-Deploy') {
            steps {
                script {
                    withKubeConfig(
                        caCertificate: '',
                        clusterName: 'my-eks2',
                        contextName: '',
                        credentialsId: 'k8s-token-webapps',
                        namespace: 'webapps',
                        restrictKubeConfigAccess: false,
                        serverUrl: 'eks-server-url'
                    ) {
                        sh 'kubectl apply -f deployment-service.yml'
                        sh 'kubectl get pods'
                        sh 'kubectl get svc'
                    }
                }
            }
        }
}
}
```

Có vẻ như EC2 instance 'Jenkins Server' không đủ dung lương để Docker có thể Containerize các Services của ứng dụng. Ta cần phải nâng dung lượng của nó.



Vào Volume của Jenkins Server và nâng lên 16GB.

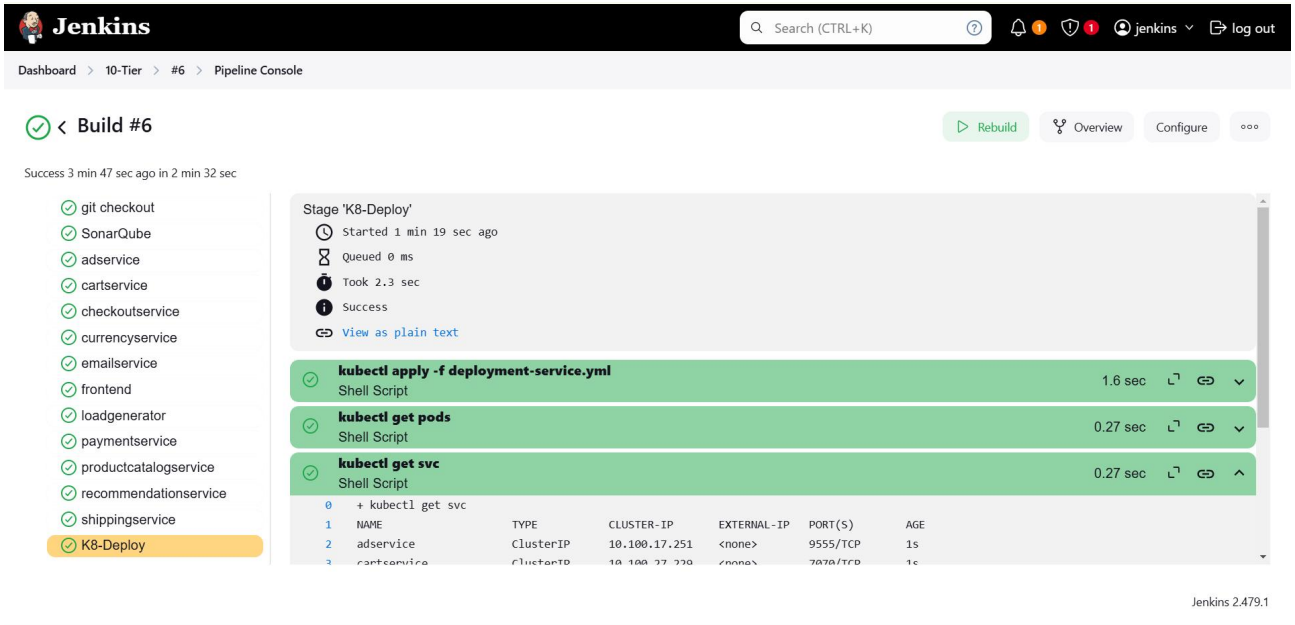




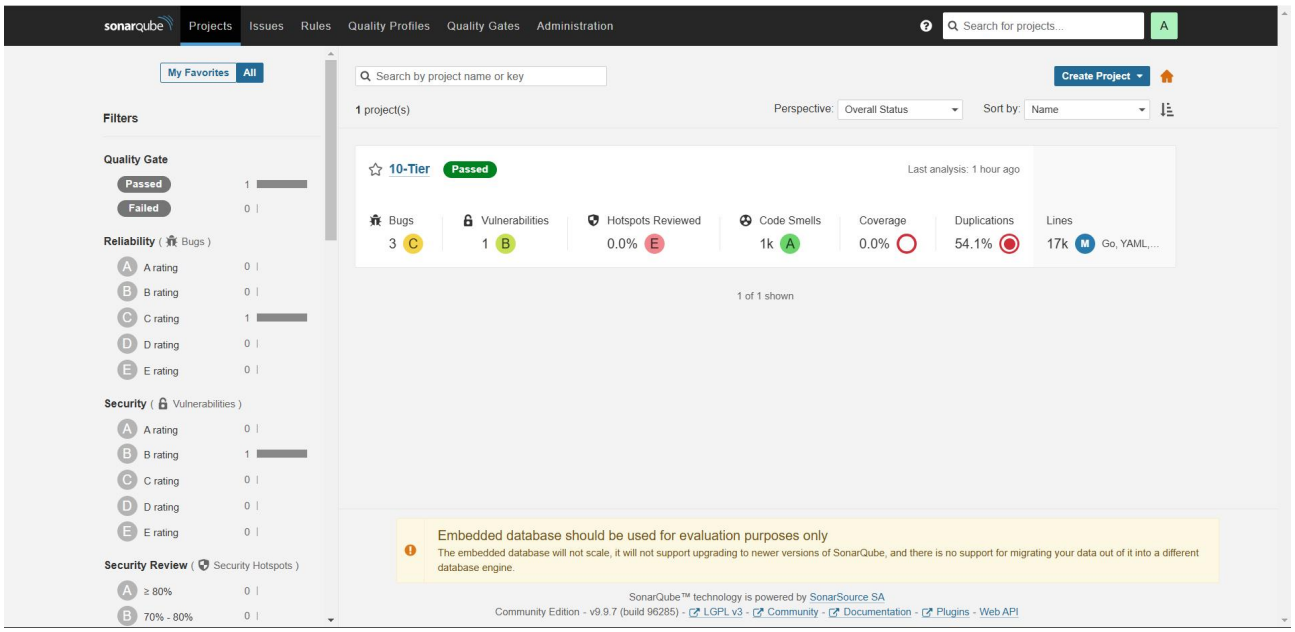
Sau đó gõ 2 câu lệnh này vào để mở rộng dung lượng.

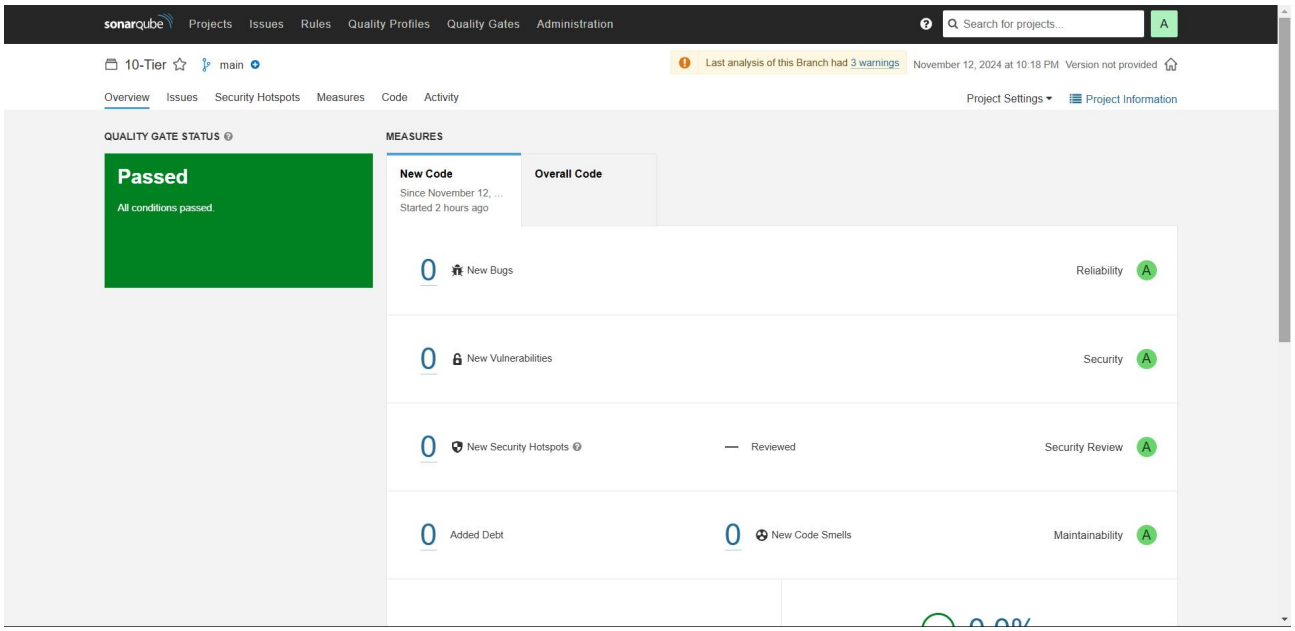
```
ubuntu@ip-172-31-39-121:~$ sudo growpart /dev/nvme0n1 1
CHANGED: partition=1 start=2099200 old: size=14677983 end=16777182 new: size=31455199 end=33554398
ubuntu@ip-172-31-39-121:~$ sudo resize2fs /dev/nvme0n1p1
resize2fs 1.47.0 (5-Feb-2023)
Filesystem at /dev/nvme0n1p1 is mounted on /; on-line resizing required
old_desc_blocks = 1, new_desc_blocks = 2
The filesystem on /dev/nvme0n1p1 is now 3931899 (4k) blocks long.

ubuntu@ip-172-31-39-121:~$
```

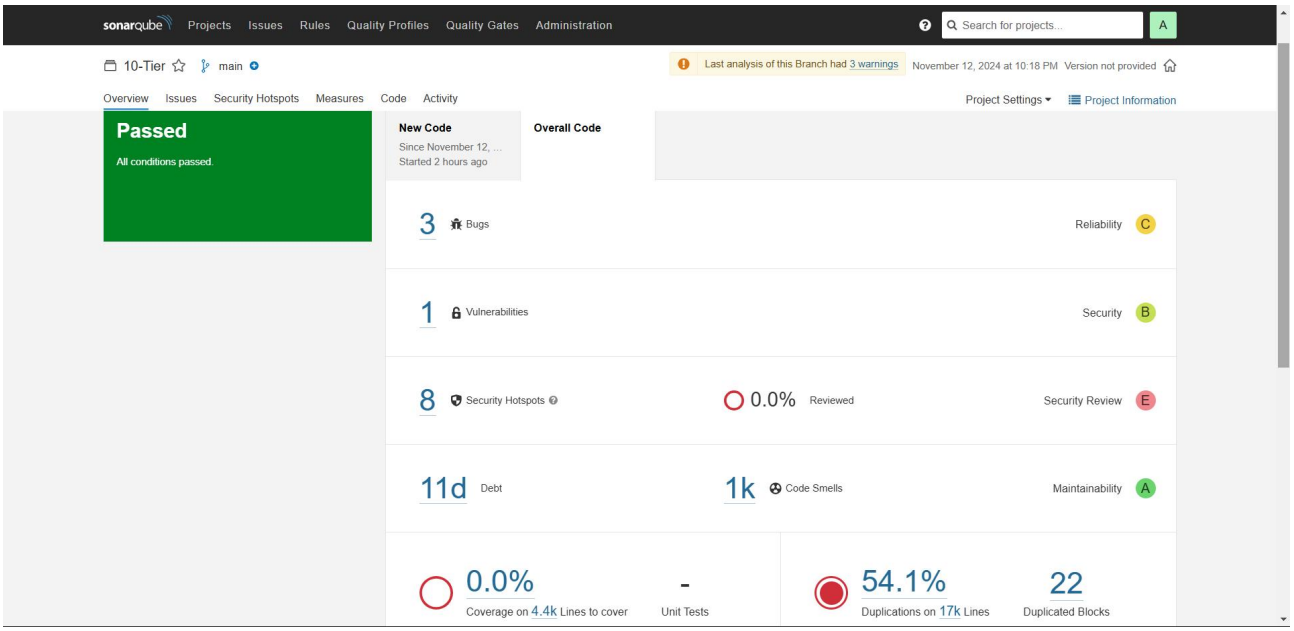
Sau rất nhiều lần fail, Pipeline đã chạy thành công!

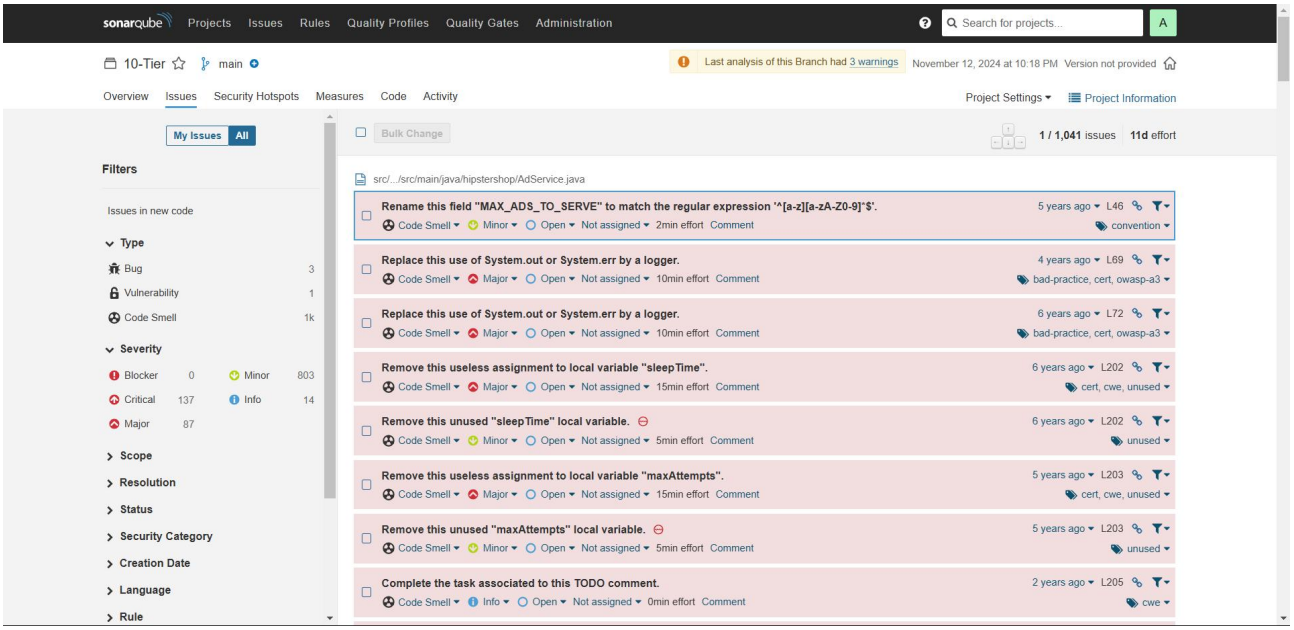


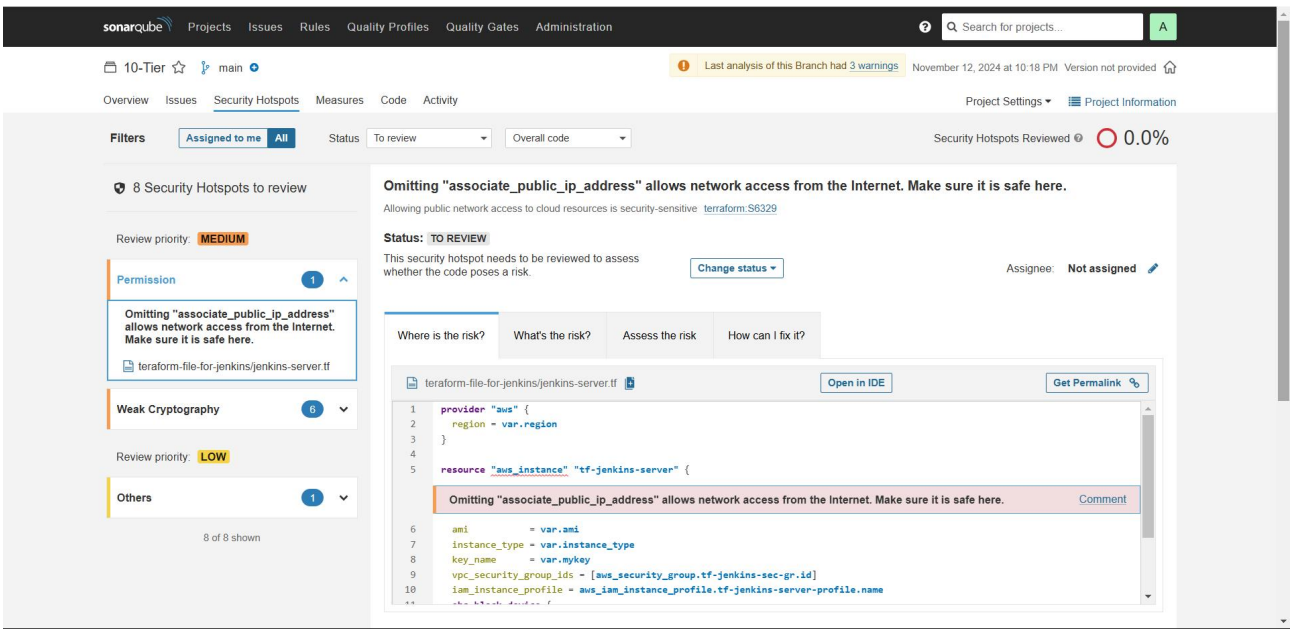
Kết quả của SonarQube scan.

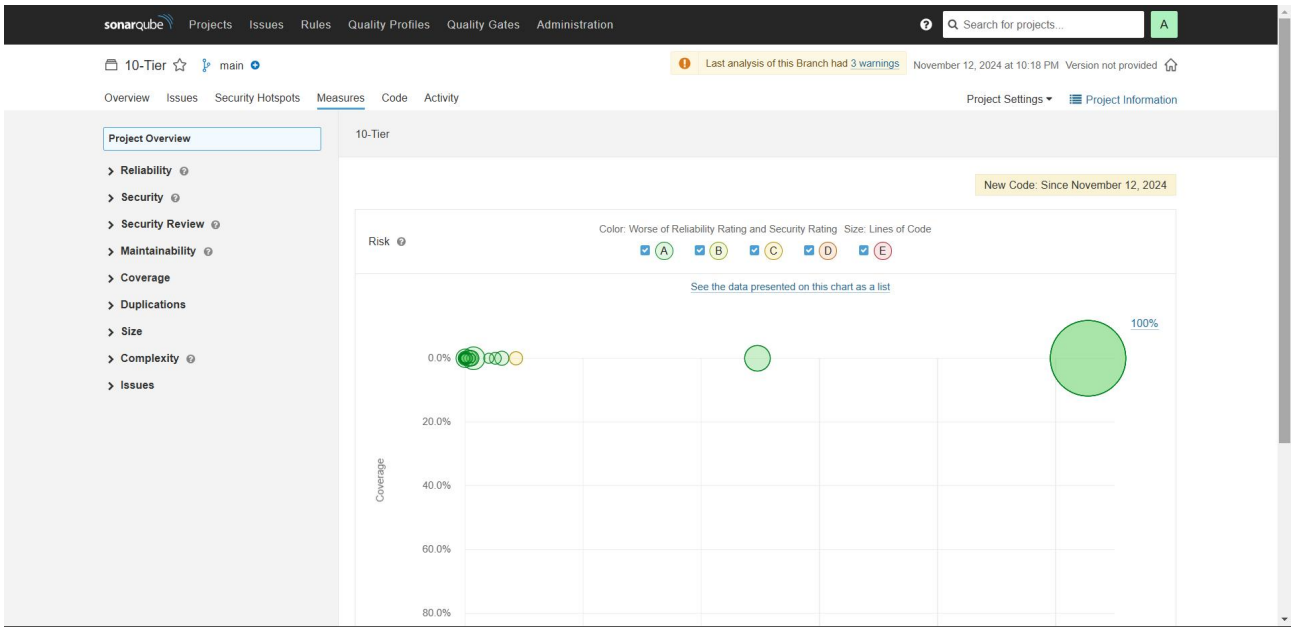


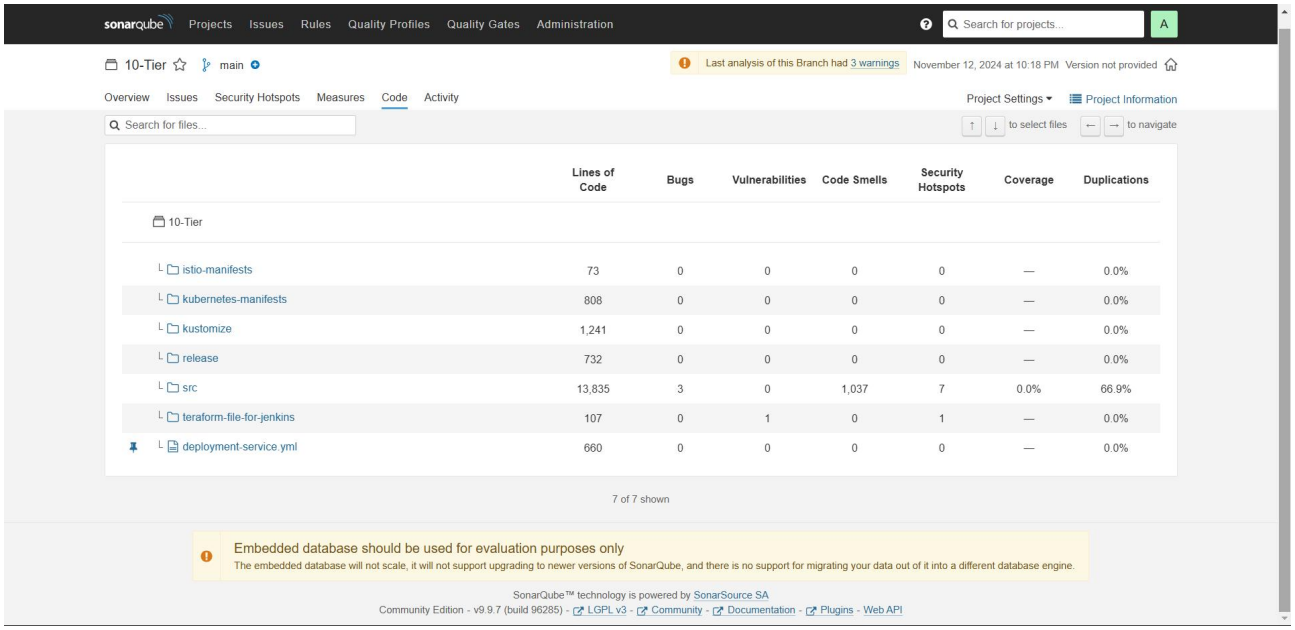



sonarqube
Projects
Issues
Rules
Quality Profiles
Quality Gates
Administration
Search for projects...
10-Tier
main
Last analysis of this Branch had 3 warnings
November 12, 2024 at 10:18 PM Version not provided
Overview
Issues
Security Hotspots
Measures
Code
Activity
Project Settings
Project Information
Passed
All conditions passed.
New Code
Since November 12, ...
Started 2 hours ago
Overall Code
3 Bugs
Reliability C
1 Vulnerabilities
Security B
8 Security Hotspots
0.0% Reviewed
Security Review E
11d Debt
1k Code Smells
Maintainability A
0.0%
Coverage on 4.4k Lines to cover
-
Unit Tests
54.1%
Duplications on 17k Lines
22
Duplicated Blocks

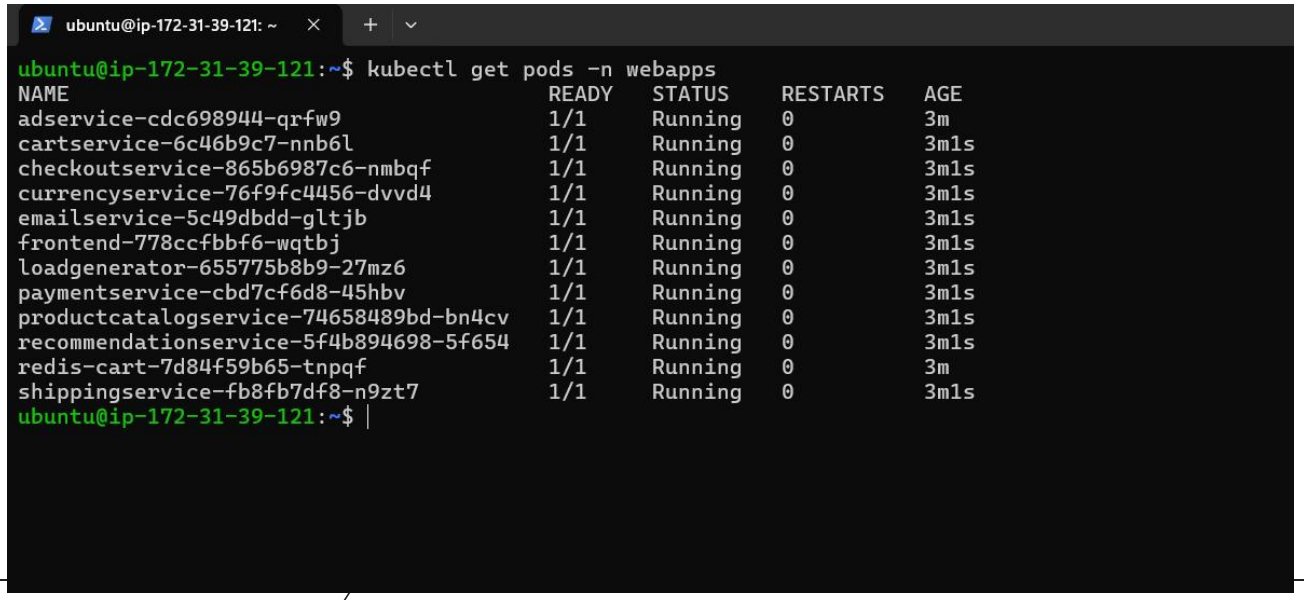

sonarqube
Projects
Issues
Rules
Quality Profiles
Quality Gates
Administration
Search for projects...
10-Tier
main
Last analysis of this Branch had 3 warnings
November 12, 2024 at 10:18 PM Version not provided
Overview
Issues
Security Hotspots
Measures
Code
Activity
Project Settings
Project Information
My Issues
All
Bulk Change
1 / 1,041 issues
11d effort
Filters
Issues in new code
Type
Bug 3
Vulnerability 1
Code Smell 1k
Severity
Blocker 0
Minor 803
Critical 137
Info 14
Major 87
Scope
Resolution
Status
Security Category
Creation Date
Language
Rule
src/.../src/main/java/hipstershop/AdService.java
Rename this field "MAX_ADS_TO_SERVE" to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
5 years ago L46
Code Smell Minor Open Not assigned 2min effort Comment
convention
Replace this use of System.out or System.err by a logger.
4 years ago L69
Code Smell Major Open Not assigned 10min effort Comment
bad-practice, cert, owasp-a3
Replace this use of System.out or System.err by a logger.
6 years ago L72
Code Smell Major Open Not assigned 10min effort Comment
bad-practice, cert, owasp-a3
Remove this useless assignment to local variable "sleepTime".
6 years ago L202
Code Smell Major Open Not assigned 15min effort Comment
cert, cwe, unused
Remove this unused "sleepTime" local variable.
6 years ago L202
Code Smell Minor Open Not assigned 5min effort Comment
unused
Remove this useless assignment to local variable "maxAttempts".
5 years ago L203
Code Smell Major Open Not assigned 15min effort Comment
cert, cwe, unused
Remove this unused "maxAttempts" local variable.
5 years ago L203
Code Smell Minor Open Not assigned 5min effort Comment
unused
Complete the task associated to this TODO comment.
2 years ago L205
Code Smell Info Open Not assigned 0min effort Comment
cwe


sonarqube
Projects
Issues
Rules
Quality Profiles
Quality Gates
Administration
Search for projects...
10-Tier
main
Last analysis of this Branch had 3 warnings
November 12, 2024 at 10:18 PM Version not provided
Overview
Issues
Security Hotspots
Measures
Code
Activity
Project Settings
Project Information
Filters
Assigned to me
All
Status
To review
Overall code
Security Hotspots Reviewed 0.0%
8 Security Hotspots to review
Review priority: MEDIUM
Permission 1
Omitting "associate_public_ip_address" allows network access from the Internet. Make sure it is safe here.
teraform-file-for-jenkins/jenkins-server.tf
Weak Cryptography 6
Review priority: LOW
Others 1
8 of 8 shown
Omitting "associate_public_ip_address" allows network access from the Internet. Make sure it is safe here.
Allowing public network access to cloud resources is security-sensitive terraform:S6329
Status: TO REVIEW
This security hotspot needs to be reviewed to assess whether the code poses a risk.
Change status
Assignee: Not assigned
Where is the risk?
What's the risk?
Assess the risk
How can I fix it?
teraform-file-for-jenkins/jenkins-server.tf
Open in IDE
Get Permalink
provider "aws" {
region = var.region
}
resource "aws_instance" "tf-jenkins-server" {
Omitting "associate_public_ip_address" allows network access from the Internet. Make sure it is safe here.
Comment
ami = var.ami
instance_type = var.instance_type
key_name = var.mykey
vpc_security_group_ids = [aws_security_group.tf-jenkins-sec-gr.id]
iam_instance_profile = aws_iam_instance_profile.tf-jenkins-server-profile.name



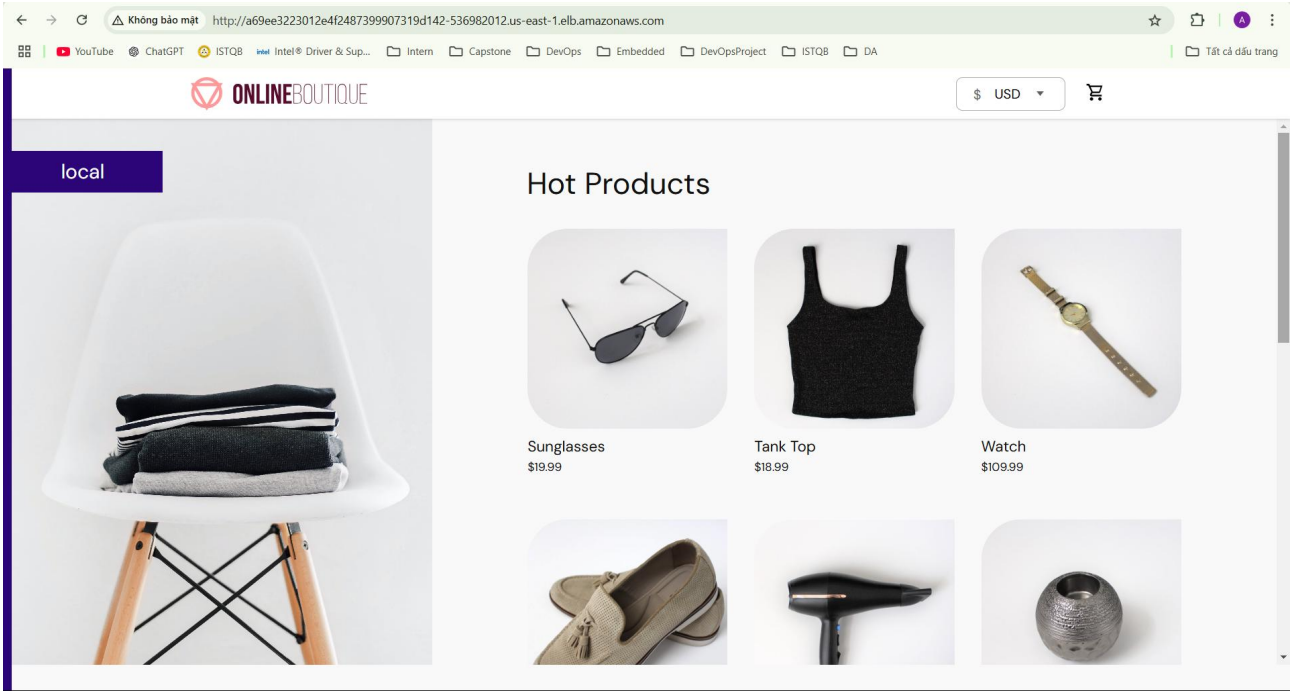


Các pod của Kubernetes đã khởi tạo thành công.



```
ubuntu@ip-172-31-39-121:~$ kubectl get svc -n webapps
NAME                 TYPE           CLUSTER-IP       EXTERNAL-IP                                                                PORT(S)        AGE
adservice            ClusterIP      10.100.17.251    <none>                                                                     9555/TCP       3m29s
cartservice          ClusterIP      10.100.27.229    <none>                                                                     7070/TCP       3m29s
checkoutservice      ClusterIP      10.100.104.214   <none>                                                                     5050/TCP       3m29s
currencyservice      ClusterIP      10.100.130.101   <none>                                                                     7000/TCP       3m29s
emailservice         ClusterIP      10.100.149.93    <none>                                                                     5000/TCP       3m29s
frontend             NodePort       10.100.31.65     <none>                                                                     80:30783/TCP   3m29s
frontend-external    LoadBalancer   10.100.222.210   a69ee3223012e4f2487399907319d142-536982012.us-east-1.elb.amazonaws.com   80:30284/TCP   3m29s
paymentservice       ClusterIP      10.100.187.130   <none>                                                                     50051/TCP      3m29s
productcatalogservice ClusterIP     10.100.133.249   <none>                                                                     3550/TCP       3m29s
recommendationservice ClusterIP     10.100.58.138    <none>                                                                     8080/TCP       3m29s
redis-cart           ClusterIP      10.100.245.175   <none>                                                                     6379/TCP       3m29s
shippingservice      ClusterIP      10.100.25.103    <none>                                                                     50051/TCP      3m29s
ubuntu@ip-172-31-39-121:~$
```

Ta có thể truy cập thông qua LoadBalancer IP.



Để tích hợp Trivy scan vào Pipeline, ta thêm các câu lệnh dưới đây vào từng services, nó sẽ được lưu lại ở Jenkins Server.

Đối với adservice.

```
        stage('adservice'){
            steps{
             script{
              withDockerRegistry(credentialsId: 'docker-cred', toolName: 'docker') {
                        dir('/var/lib/jenkins/workspace/10-Tier/src/adservice') {
                            sh 'docker build -t levi2708/adservice:latest .'

                            sh '''
                            trivy image levi2708/adservice:latest > trivy_adservice_report.txt || true
                            cat trivy_adservice_report.txt
                            '''

                            sh "docker push levi2708/adservice:latest"
                            sh "docker rmi levi2708/adservice:latest"
                        }
                    }
            }
            }
        }
```

Đối với cartservice và tương tự các service khác.

```
43          stage('cartservice'){
44              steps{
45               script{
46                withDockerRegistry(credentialsId: 'docker-cred', toolName: 'docker') {
47                          dir('/var/lib/jenkins/workspace/10-Tier/src/cartservice/src/') {
48                              sh 'docker build -t levi2708/cartservice:latest .'
49
50                              sh '''
51                              trivy image levi2708/carservice:latest > trivy_carservice_report.txt || true
52                              cat trivy_carservice_report.txt
53                              '''
54
55                              sh "docker push levi2708/cartservice:latest"
56                              sh "docker rmi levi2708/cartservice:latest"
57                          }
58                      }
59              }
60              }
61          }
```

Output của Trivy scan.

```
ubuntu@ip-172-31-39-121:/var/lib/jenkins/workspace/10-Tier/src$ cd currencyservice
ubuntu@ip-172-31-39-121:/var/lib/jenkins/workspace/10-Tier/src/currencyservice$ ls
Dockerfile  client.js  data  genproto.sh  package-lock.json  package.json  proto  server.js  trivy_currencyservice_report.txt
ubuntu@ip-172-31-39-121:/var/lib/jenkins/workspace/10-Tier/src/currencyservice$ cat trivy_currencyservice_report.txt

For OSS Maintainers: VEX Notice
--------------------------------
If you're an OSS maintainer and Trivy has detected vulnerabilities in your project that you believe are not actually exploitable, consider issuing a VEX (Vu
lnerability Exploitability eXchange) statement.
VEX allows you to communicate the actual status of vulnerabilities in your project, improving security transparency and reducing false positives for your us
ers.
Learn more and start using VEX: https://aquasecurity.github.io/trivy/v0.57/docs/supply-chain/vex/repo#publishing-vex-documents

To disable this notice, set the TRIVY_DISABLE_VEX_NOTICE environment variable.


levi2708/currencyservice:latest (alpine 3.17.3)
================================================
Total: 44 (UNKNOWN: 0, LOW: 4, MEDIUM: 38, HIGH: 2, CRITICAL: 0)
```

Library	Vulnerability	Severity	Status	Installed Version	Fixed Version	Title
busybox	CVE-2023-42363	MEDIUM	fixed	1.35.0-r29	1.35.0-r31	busybox: use-after-free in awk https://avd.aquasec.com/nvd/cve-2023-42363
	CVE-2023-42364					busybox: use-after-free https://avd.aquasec.com/nvd/cve-2023-42364
	CVE-2023-42365					busybox: use-after-free https://avd.aquasec.com/nvd/cve-2023-42365
	CVE-2023-42366				1.35.0-r30	busybox: A heap-buffer-overflow https://avd.aquasec.com/nvd/cve-2023-42366
busybox-binsh	CVE-2023-42363				1.35.0-r31	busybox: use-after-free in awk https://avd.aquasec.com/nvd/cve-2023-42363
	CVE-2023-42364					busybox: use-after-free

```
ubuntu@ip-172-31-39-121:/var/lib/jenkins/workspace/10-Tier/src$ cd frontend
ubuntu@ip-172-31-39-121:/var/lib/jenkins/workspace/10-Tier/src/frontend$ ls
Dockerfile  deployment_details.go  genproto.sh  go.sum       main.go       money   static     trivy_frontend_report.txt
README.md   genproto               go.mod       handlers.go  middleware.go  rpc.go  templates
ubuntu@ip-172-31-39-121:/var/lib/jenkins/workspace/10-Tier/src/frontend$ cat trivy_frontend_report.txt

For OSS Maintainers: VEX Notice
--------------------------------
If you're an OSS maintainer and Trivy has detected vulnerabilities in your project that you believe are not actually exploitable, consider issuing a VEX (Vu
lnerability Exploitability eXchange) statement.
VEX allows you to communicate the actual status of vulnerabilities in your project, improving security transparency and reducing false positives for your us
ers.
Learn more and start using VEX: https://aquasecurity.github.io/trivy/v0.57/docs/supply-chain/vex/repo#publishing-vex-documents

To disable this notice, set the TRIVY_DISABLE_VEX_NOTICE environment variable.


levi2708/frontend:latest (alpine 3.18.0)
========================================
Total: 45 (UNKNOWN: 0, LOW: 4, MEDIUM: 36, HIGH: 2, CRITICAL: 3)
```

Library	Vulnerability	Severity	Status	Installed Version	Fixed Version	Title
busybox	CVE-2022-48174	CRITICAL	fixed	1.36.0-r9	1.36.1-r1	busybox: stack overflow vulnerability in ash.c leads to arbitrary code execution https://avd.aquasec.com/nvd/cve-2022-48174
	CVE-2023-42363	MEDIUM			1.36.1-r7	busybox: use-after-free in awk https://avd.aquasec.com/nvd/cve-2023-42363
	CVE-2023-42364					busybox: use-after-free