# BUILD INFRASTRUCTURE USING TERRAFORM:

Download and install AWS CLI

Configure AWS from Command line with secret key and access keys

Command:

```
aws –version
```

```
aws configure
```

Navigate to the AWS_Build_Infrastructure folder and use below commands

Command:

- terraform init
- terraform plan
- terraform apply

# Get Login

```
aws ecr get-login-password --region us-west-2 | docker login --username AWS --password-stdin
220XXXXXXX90.dkr.ecr.us-west-2.amazonaws.com/rockstar
```

# Create ECR Repository:

```
aws ecr create-repository --repository-name rockstar --region us-west-2
```

# Build the project,

```
npm run build
```

# Build docker images using URI of the ECR,

```
docker build -f rockstar/Dockerfile -t 220XXXXXXX90.dkr.ecr.us-west-2.amazonaws.com/rockstar:latest
./rockstar
```

# Run Docker Image locally & Test

```
docker run --name rockstar -p 80:80 --rm -d
```

```
220XXXXXXX90.dkr.ecr.us-west-2.amazonaws.com/rockstar:latest
```

# Push the Docker Image

```
docker push 220XXXXXXX90.dkr.ecr.us-west-2.amazonaws.com/rockstar:latest
```

# Run Docker Image locally & Test

```
docker run --name rockstar -p 80:80 --rm -d 220XXXXXXX90.dkr.ecr.us-west-
2.amazonaws.com/rockstar:latest
```

```
# Export your Account ID

export ACCOUNT_ID=220XXXXXXX90


# Create Asssume Role & Set Trust Policy

TRUST="{ \"Version\": \"2012-10-17\", \"Statement\": [ { \"Effect\": \"Allow\", \"Principal\": { \"AWS\": \"arn:aws:iam::${ACCOUNT_ID}:root\" }, \"Action\": \"sts:AssumeRole\" } ] }"
```

Note: If not code build fails and throws an error GetTokenAuthorization

```
# Create IAM Role for CodeBuild to Interact EKS

aws iam create-role --role-name EksCodeBuildKubectlRole --assume-role-policy-document "$TRUST" --output text --query 'Role.Arn'


# Define Inline Policy with eks Describe permission in a file iam-eks-describe-policy

echo '{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": "eks:Describe*", "Resource": "*" } ] }' > /tmp/iam-eks-describe-policy

# Associate Inline Policy to our newly created IAM Role

aws iam put-role-policy --role-name EksCodeBuildKubectlRole --policy-name eks-describe --policy-document file://C:/Users/keert/AppData/Local/Temp/iam-eks-describe-policy


# Set ROLE value

ROLE="    - rolearn: arn:aws:iam::$ACCOUNT_ID:role/EksCodeBuildKubectlRole\n      username: build\n      groups:\n        - system:masters"


# Get current aws-auth configMap data and attach new role info to it

kubectl get -n kube-system configmap/aws-auth -o yaml | awk "/mapRoles: \|/{print;print \"$ROLE\";next}1" > /tmp/aws-auth-patch.yml


# Patch the aws-auth configmap with new role

kubectl patch configmap/aws-auth -n kube-system --patch "$(cat /tmp/aws-auth-patch.yml)"
```

# Verify what is updated in aws-auth configmap after change

kubectl get configmap aws-auth -o yaml -n kube-system

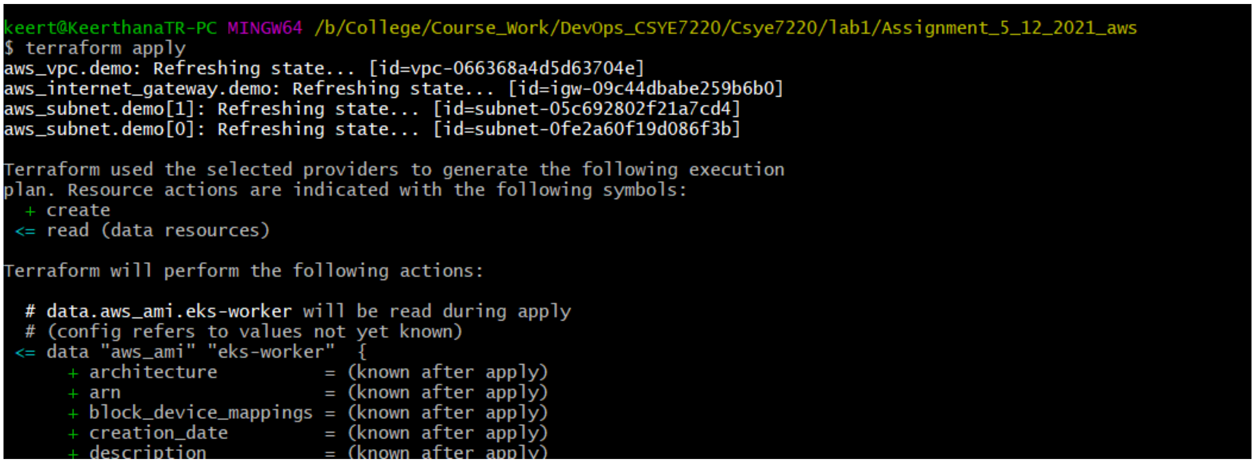
# Environment Variables for CODEBUILD:

REPOSITORY_URI = 220XXXXXXX90.dkr.ecr.us-west-2.amazonaws.com/rockstar

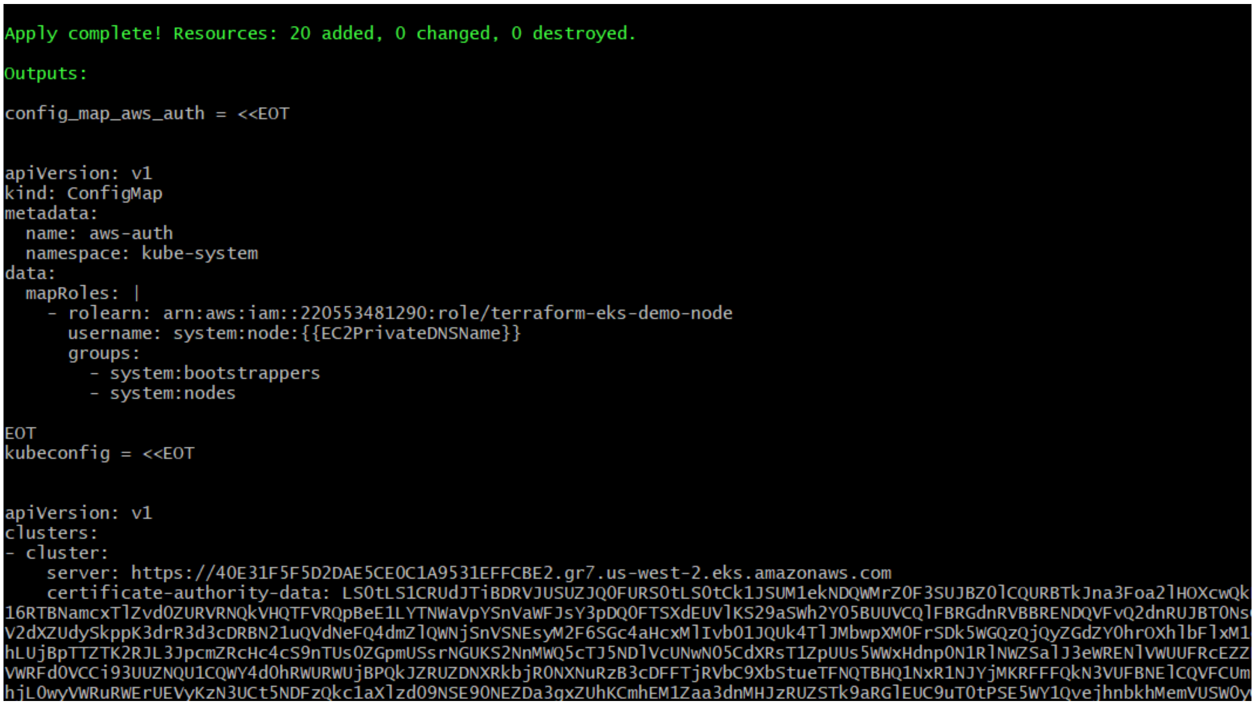
EKS_KUBECTL_ROLE_ARN = arn:aws:iam::220XXXXXXX90:role/EksCodeBuildKubectlRole

EKS_CLUSTER_NAME = terraform-eks-demo

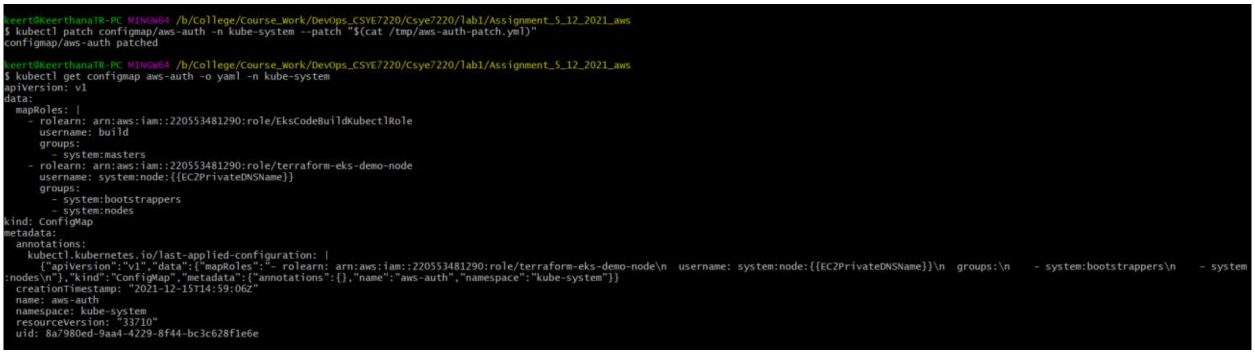
# Below are the screenshots of the above commands,



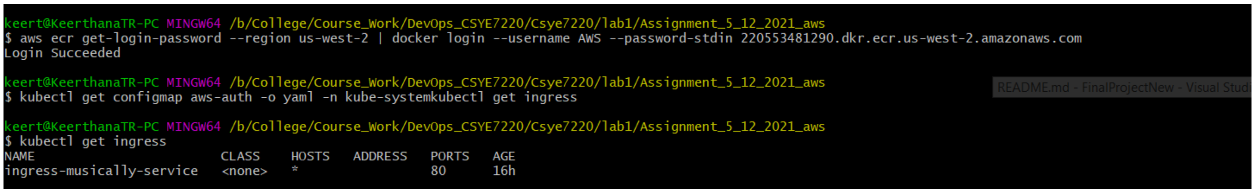
```
keert@KeerthanaTR-PC MINGW64 /b/College/Course_Work/DevOps_CSYE7220/Csye7220/lab1/Assignment_5_12_2021_aws
$ terraform apply
aws_vpc.demo: Refreshing state... [id=vpc-066368a4d5d63704e]
aws_internet_gateway.demo: Refreshing state... [id=igw-09c44dbabe259b6b0]
aws_subnet.demo[1]: Refreshing state... [id=subnet-05c692802f21a7cd4]
aws_subnet.demo[0]: Refreshing state... [id=subnet-0fe2a60f19d086f3b]

Terraform used the selected providers to generate the following execution
plan. Resource actions are indicated with the following symbols:
  + create
 <= read (data resources)

Terraform will perform the following actions:

  # data.aws_ami.eks-worker will be read during apply
  # (config refers to values not yet known)
 <= data "aws_ami" "eks-worker"  {
      + architecture          = (known after apply)
      + arn                   = (known after apply)
      + block_device_mappings = (known after apply)
      + creation_date         = (known after apply)
      + description           = (known after apply)
```



```
Apply complete! Resources: 20 added, 0 changed, 0 destroyed.

Outputs:

config_map_aws_auth = <<EOT


apiVersion: v1
kind: ConfigMap
metadata:
  name: aws-auth
  namespace: kube-system
data:
  mapRoles: |
    - rolearn: arn:aws:iam::220553481290:role/terraform-eks-demo-node
      username: system:node:{{EC2PrivateDNSName}}
      groups:
        - system:bootstrappers
        - system:nodes

EOT
kubeconfig = <<EOT


apiVersion: v1
clusters:
- cluster:
    server: https://40E31F5F5D2DAE5CE0C1A9531EFFCBE2.gr7.us-west-2.eks.amazonaws.com
    certificate-authority-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUM1ekNDQWMrZ0F3SUJBZ0lCQURBTkJna3Foa2lHOXcwQk
16RTBNamcxTlZvd0ZURVRNQkVHQTFVRQpBeE1LYTNWaVpYSnVaWFJsY3pDQ0FTSXdEUVlKS29aSWh2Y05BUUVCQlFBRGdnRVBBRENDQVFvQ2dnRUJBT0Ns
V2dXZUdySkppK3drR3d3cDRBN21uQVdNeFQ4dmZlQWNjSnVSNEsyM2F6SGc4aHcxMlIvb01JQUk4TlJMbwpXM0FrSDk5WGQzQjQyZGdZY0hrOXhlbFlxM1
hLUjBpTTZTK2RJL3JpcmZRcHc4cS9nTUs0ZGpmUSsrNGUKS2NnMWQ5cTJ5NDlVcUNwN05CdXRsT1ZpUUs5WWxHdnp0N1RlNWZSalJ3eWRENlVWUUFRcEZZ
VWRFd0VCCi93UUZNQU1CQWY4d0hRWURWUjBPQkJZRUZDNXRkbjR0NXNuRzB3cDFFTjRVbC9XbStueTFNQTBHQ1NxR1NJYjMKRFFFQkN3VUFBNElCQVFCUm
hjL0wyVWRuRWErUEVyKzN3UCt5NDFzQkc1aXlzd09NSE90NEZDa3gxZUhKCmhEM1Zaa3dnMHJzRUZSTk9aRGlEUC9uT0tPSE5WY1QvejhnbkhMemVUSW0y
```

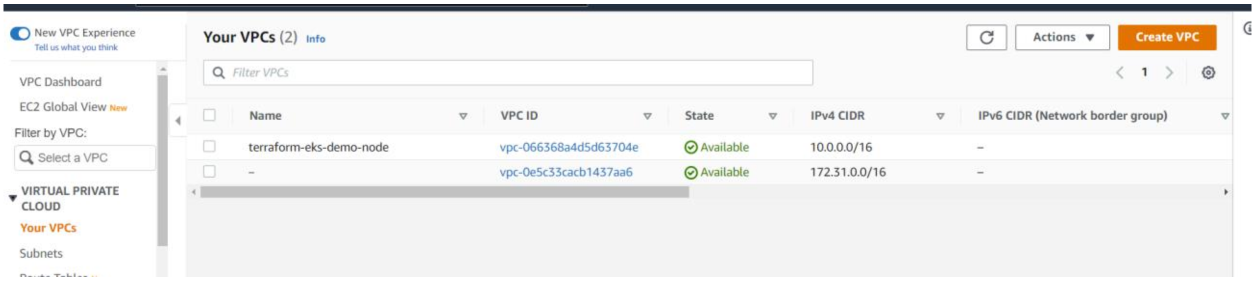


```
keert@KeerthanaTR-PC MINGW64 /b/College/Course_Work/DevOps_CSYE7220/Csye7220/lab1/Assignment_5_12_2021_aws
$ kubectl patch configmap/aws-auth -n kube-system --patch "$(cat /tmp/aws-auth-patch.yml)"
configmap/aws-auth patched

keert@KeerthanaTR-PC MINGW64 /b/College/Course_Work/DevOps_CSYE7220/Csye7220/lab1/Assignment_5_12_2021_aws
$ kubectl get configmap aws-auth -o yaml -n kube-system
apiVersion: v1
data:
  mapRoles: |
    - rolearn: arn:aws:iam::220553481290:role/EksCodeBuildKubectlRole
      username: build
      groups:
        - system:masters
    - rolearn: arn:aws:iam::220553481290:role/terraform-eks-demo-node
      username: system:node:{{EC2PrivateDNSName}}
      groups:
        - system:bootstrappers
        - system:nodes
kind: ConfigMap
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |
      {"apiVersion":"v1","data":{"mapRoles":"- rolearn: arn:aws:iam::220553481290:role/terraform-eks-demo-node\n  username: system:node:{{EC2PrivateDNSName}}\n  groups:\n    - system:bootstrappers\n    - system
:nodes\n"},"kind":"ConfigMap","metadata":{"annotations":{},"name":"aws-auth","namespace":"kube-system"}}
  creationTimestamp: "2021-12-15T14:59:06Z"
  name: aws-auth
  namespace: kube-system
  resourceVersion: "33710"
  uid: 8a7980ed-9aa4-4229-8f44-bc3c628f1e6e
```



```
keert@KeerthanaTR-PC MINGW64 /b/College/Course_Work/DevOps_CSYE7220/Csye7220/lab1/Assignment_5_12_2021_aws
$ aws ecr get-login-password --region us-west-2 | docker login --username AWS --password-stdin 220553481290.dkr.ecr.us-west-2.amazonaws.com
Login Succeeded

keert@KeerthanaTR-PC MINGW64 /b/College/Course_Work/DevOps_CSYE7220/Csye7220/lab1/Assignment_5_12_2021_aws
$ kubectl get configmap aws-auth -o yaml -n kube-systemkubectl get ingress

keert@KeerthanaTR-PC MINGW64 /b/College/Course_Work/DevOps_CSYE7220/Csye7220/lab1/Assignment_5_12_2021_aws
$ kubectl get ingress
NAME                         CLASS    HOSTS   ADDRESS   PORTS   AGE
ingress-musically-service    <none>   *                 80      16h
```

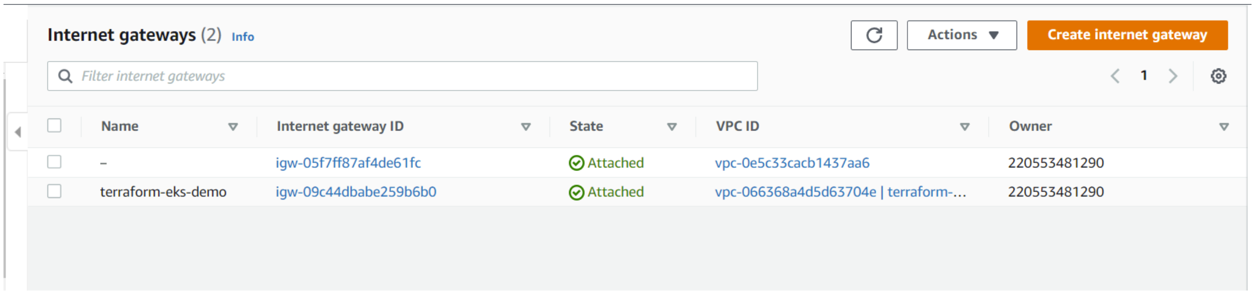
# AWS VPC



Your VPCs (2) Info

□	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR (Network border group)
□	terraform-eks-demo-node	vpc-066368a4d5d63704e	✓ Available	10.0.0.0/16	–
□	–	vpc-0e5c33cacb1437aa6	✓ Available	172.31.0.0/16	–

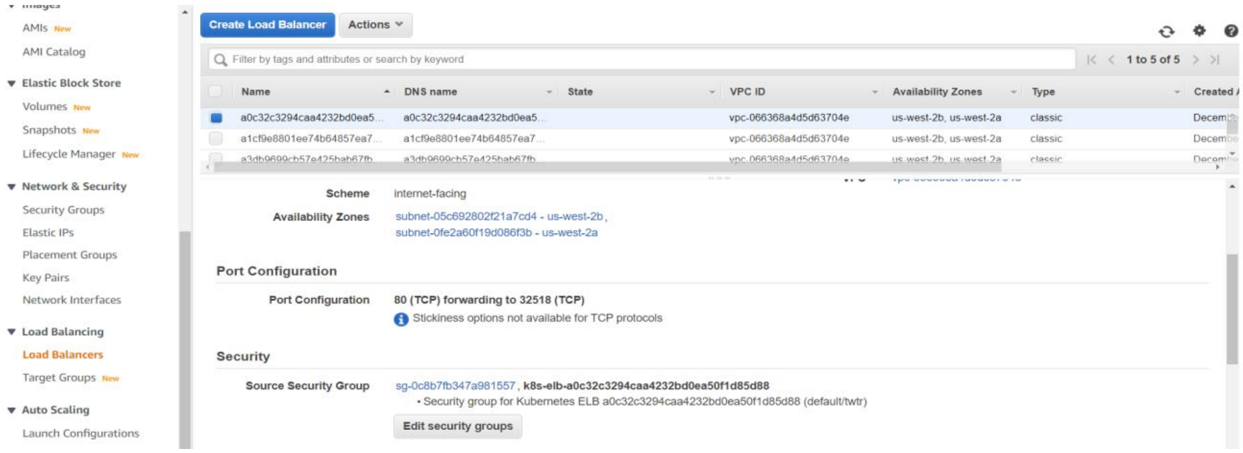
# AWS INTERNET GATEWAY



Internet gateways (2) Info

□	Name	Internet gateway ID	State	VPC ID	Owner
□	–	igw-05f7ff87af4de61fc	✓ Attached	vpc-0e5c33cacb1437aa6	220553481290
□	terraform-eks-demo	igw-09c44dbabe259b6b0	✓ Attached	vpc-066368a4d5d63704e | terraform-...	220553481290

# AWS LOAD BALANCER



	Name	DNS name	State	VPC ID	Availability Zones	Type	Created
■	a0c32c3294caa4232bd0ea5...	a0c32c3294caa4232bd0ea5...		vpc-066368a4d5d63704e	us-west-2b, us-west-2a	classic	Decem
□	a1cf9e8801ee74b64857ea7...	a1cf9e8801ee74b64857ea7...		vpc-066368a4d5d63704e	us-west-2b, us-west-2a	classic	Decembe
□	a3db9699cb57e425bab67fb...	a3db9699cb57e425bab67fb...		vpc-066368a4d5d63704e	us-west-2b, us-west-2a	classic	Decem

Scheme internet-facing
Availability Zones subnet-05c692802f21a7cd4 - us-west-2b,
subnet-0fe2a60f19d086f3b - us-west-2a

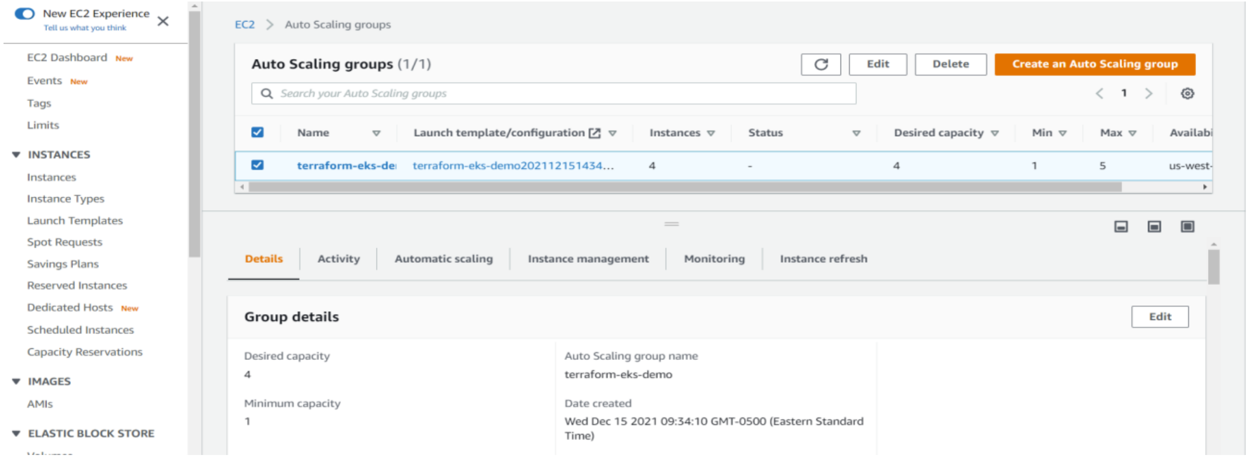
Port Configuration

Port Configuration 80 (TCP) forwarding to 32518 (TCP)
Stickiness options not available for TCP protocols

Security

Source Security Group sg-0c8b7fb347a981557, k8s-elb-a0c32c3294caa4232bd0ea50f1d85d88
• Security group for Kubernetes ELB a0c32c3294caa4232bd0ea50f1d85d88 (default/twtr)

# AWS AUTOSCALING GROUP



Auto Scaling groups (1/1)

☑	Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availabi
☑	terraform-eks-de	terraform-eks-demo202112151434...	4	-	4	1	5	us-west-

Group details

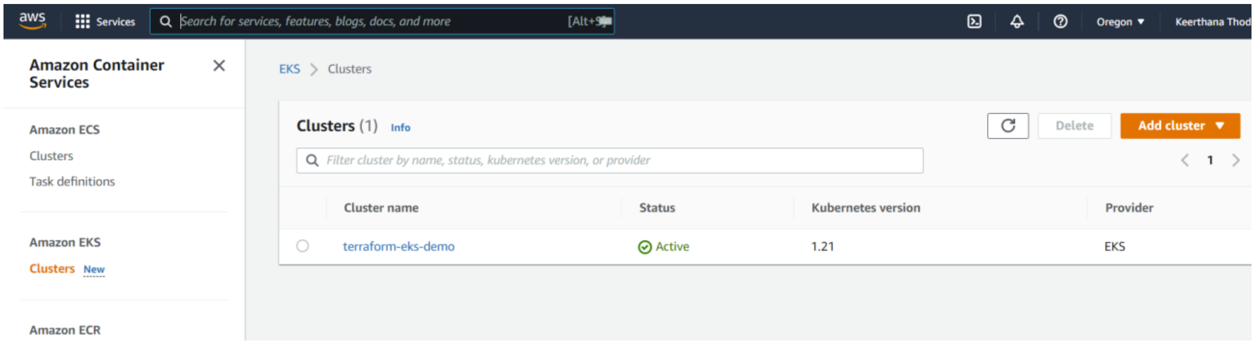
Desired capacity
4

Minimum capacity
1

Auto Scaling group name
terraform-eks-demo

Date created
Wed Dec 15 2021 09:34:10 GMT-0500 (Eastern Standard Time)

# EKS Cluster:

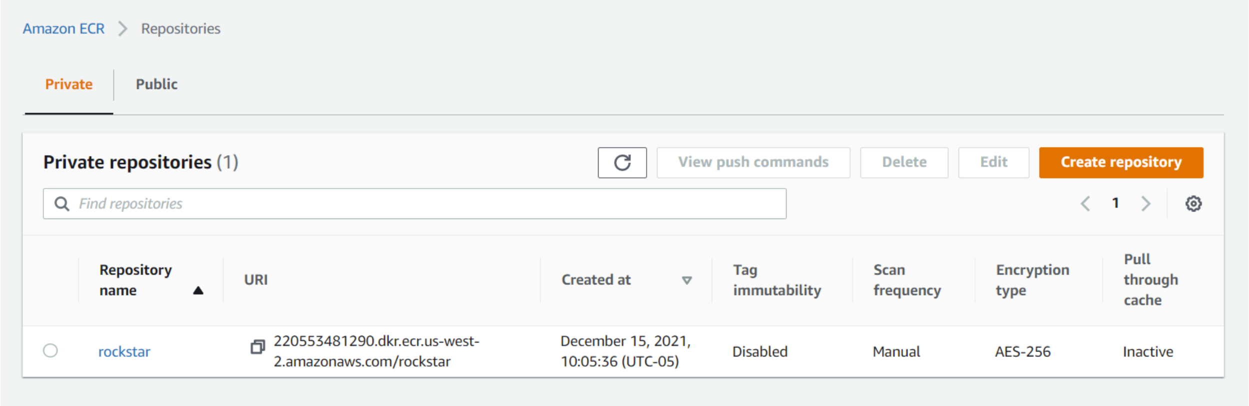

aws
Services
Search for services, features, blogs, docs, and more
[Alt+S]
Oregon
Keerthana Thod
Amazon Container Services
Amazon ECS
Clusters
Task definitions
Amazon EKS
Clusters New
Amazon ECR
EKS > Clusters
Info
Delete
Add cluster
Filter cluster by name, status, kubernetes version, or provider
1

Clusters (1)

	Cluster name	Status	Kubernetes version	Provider
○	terraform-eks-demo	✓ Active	1.21	EKS

# AWS ECR REPOSITORY

- Create a repository
- Tag Immutability: Disable
- Scan on Push – Enable
- Create and make a note of the repository name

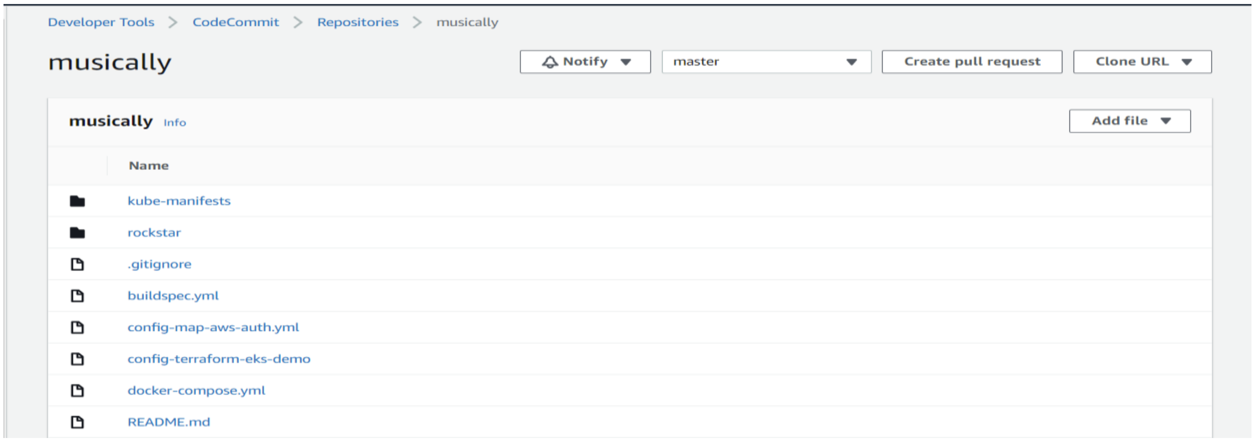


Private repositories (1)

	Repository name ▲	URI	Created at ▽	Tag immutability	Scan frequency	Encryption type	Pull through cache
○	rockstar	220553481290.dkr.ecr.us-west-2.amazonaws.com/rockstar	December 15, 2021, 10:05:36 (UTC-05)	Disabled	Manual	AES-256	Inactive

# AWS CODE COMMIT

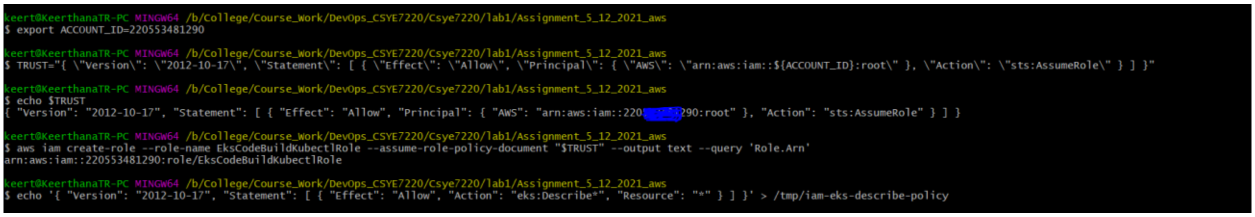
- Create a repository in code commit named rockstar
- Clone and load the project
- Create git credentials from IAM Service and make a note of those credentials.
- Clone the git repository from Code Commit to local repository, provide git credentials generated to login to git repo
- Push the code and Kubernetes manifests into the repository



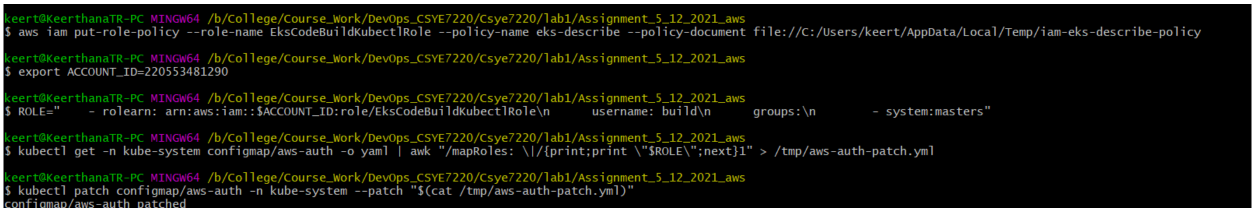
Name
kube-manifests
rockstar
.gitignore
buildspec.yml
config-map-aws-auth.yml
config-terraform-eks-demo
docker-compose.yml
README.md

# IAM ROLES

- Create EksCodeBuildKubectl Role for CodeBuild to interact with AWS EKS
- Set Trust Policy
- Define inline policy and associate with the role
- Update EKS cluster with the new role
- Patch the aws-auth config-map with new role

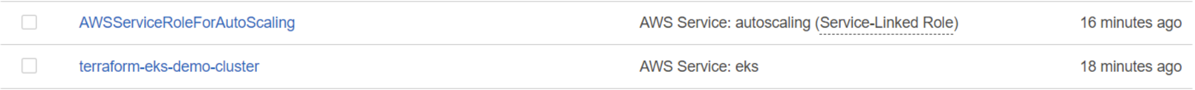


```
keert@KeerthanaTR-PC MINGW64 /b/College/Course_Work/DevOps_CSYE7220/Csye7220/lab1/Assignment_5_12_2021_aws
$ export ACCOUNT_ID=220553481290

keert@KeerthanaTR-PC MINGW64 /b/College/Course_Work/DevOps_CSYE7220/Csye7220/lab1/Assignment_5_12_2021_aws
$ TRUST="{ \"Version\": \"2012-10-17\", \"Statement\": [ { \"Effect\": \"Allow\", \"Principal\": { \"AWS\": \"arn:aws:iam::${ACCOUNT_ID}:root\" }, \"Action\": \"sts:AssumeRole\" } ] }"

keert@KeerthanaTR-PC MINGW64 /b/College/Course_Work/DevOps_CSYE7220/Csye7220/lab1/Assignment_5_12_2021_aws
$ echo $TRUST
{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Principal": { "AWS": "arn:aws:iam::220[redacted]290:root" }, "Action": "sts:AssumeRole" } ] }

keert@KeerthanaTR-PC MINGW64 /b/College/Course_Work/DevOps_CSYE7220/Csye7220/lab1/Assignment_5_12_2021_aws
$ aws iam create-role --role-name EksCodeBuildKubectlRole --assume-role-policy-document "$TRUST" --output text --query 'Role.Arn'
arn:aws:iam::220553481290:role/EksCodeBuildKubectlRole

keert@KeerthanaTR-PC MINGW64 /b/College/Course_Work/DevOps_CSYE7220/Csye7220/lab1/Assignment_5_12_2021_aws
$ echo '{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": "eks:Describe*", "Resource": "*" } ] }' > /tmp/iam-eks-describe-policy
```

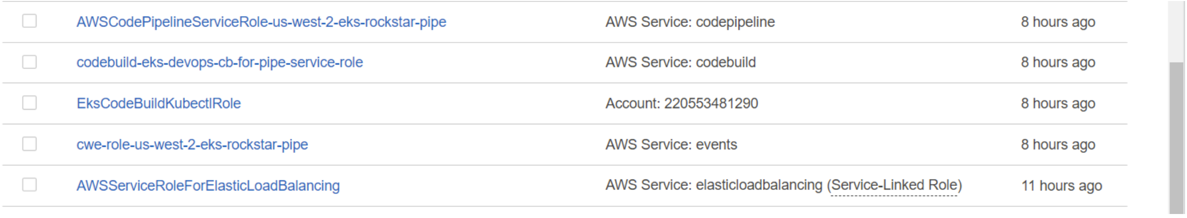


```
keert@KeerthanaTR-PC MINGW64 /b/College/Course_Work/DevOps_CSYE7220/Csye7220/lab1/Assignment_5_12_2021_aws
$ aws iam put-role-policy --role-name EksCodeBuildKubectlRole --policy-name eks-describe --policy-document file://C:/Users/keert/AppData/Local/Temp/iam-eks-describe-policy

keert@KeerthanaTR-PC MINGW64 /b/College/Course_Work/DevOps_CSYE7220/Csye7220/lab1/Assignment_5_12_2021_aws
$ export ACCOUNT_ID=220553481290

keert@KeerthanaTR-PC MINGW64 /b/College/Course_Work/DevOps_CSYE7220/Csye7220/lab1/Assignment_5_12_2021_aws
$ ROLE="    - rolearn: arn:aws:iam::$ACCOUNT_ID:role/EksCodeBuildKubectlRole\n      username: build\n      groups:\n        - system:masters"

keert@KeerthanaTR-PC MINGW64 /b/College/Course_Work/DevOps_CSYE7220/Csye7220/lab1/Assignment_5_12_2021_aws
$ kubectl get -n kube-system configmap/aws-auth -o yaml | awk "/mapRoles: \|/{print;print \"$ROLE\";next}1" > /tmp/aws-auth-patch.yml

keert@KeerthanaTR-PC MINGW64 /b/College/Course_Work/DevOps_CSYE7220/Csye7220/lab1/Assignment_5_12_2021_aws
$ kubectl patch configmap/aws-auth -n kube-system --patch "$(cat /tmp/aws-auth-patch.yml)"
configmap/aws-auth patched
```

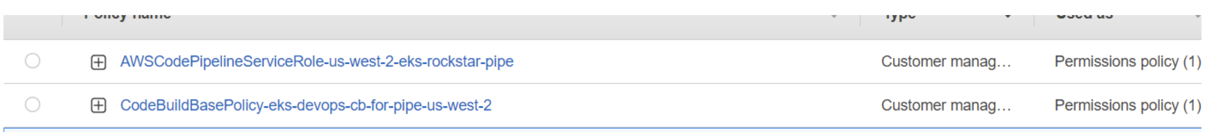
# Confirm the following roles are created:



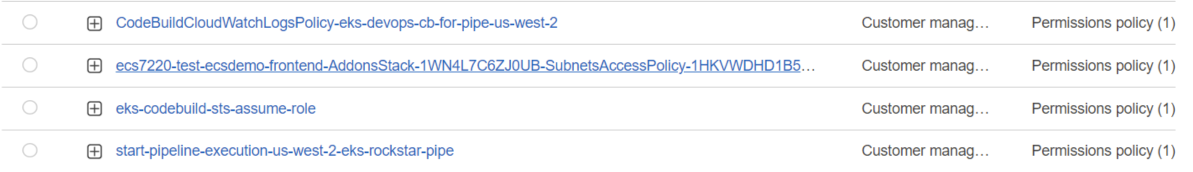
□	AWSServiceRoleForAutoScaling	AWS Service: autoscaling (Service-Linked Role)	16 minutes ago
□	terraform-eks-demo-cluster	AWS Service: eks	18 minutes ago



□	AWSCodePipelineServiceRole-us-west-2-eks-rockstar-pipe	AWS Service: codepipeline	8 hours ago
□	codebuild-eks-devops-cb-for-pipe-service-role	AWS Service: codebuild	8 hours ago
□	EksCodeBuildKubectlRole	Account: 220553481290	8 hours ago
□	cwe-role-us-west-2-eks-rockstar-pipe	AWS Service: events	8 hours ago
□	AWSServiceRoleForElasticLoadBalancing	AWS Service: elasticloadbalancing (Service-Linked Role)	11 hours ago



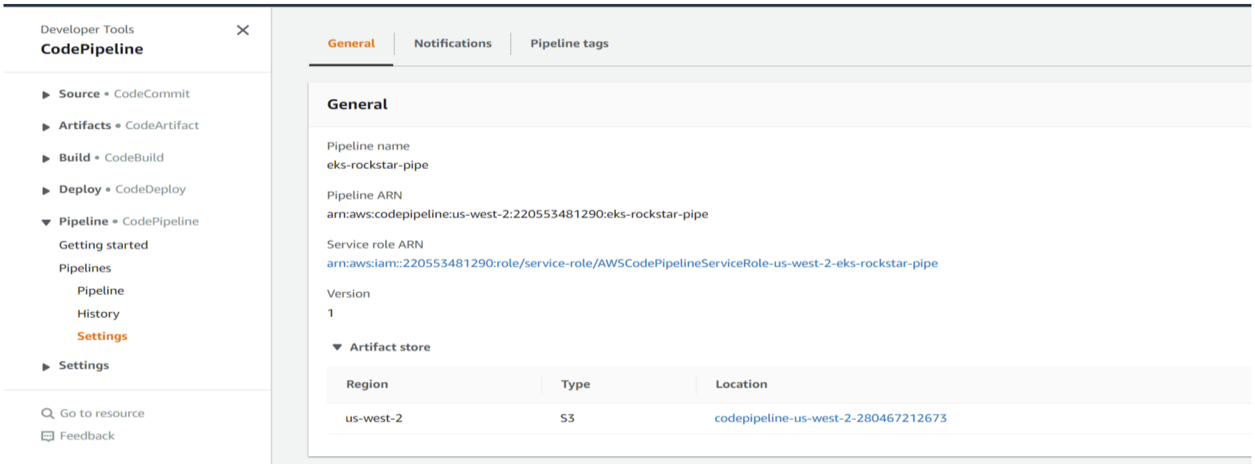
		Policy name	Type	Used as
○	⊞	AWSCodePipelineServiceRole-us-west-2-eks-rockstar-pipe	Customer manag…	Permissions policy (1)
○	⊞	CodeBuildBasePolicy-eks-devops-cb-for-pipe-us-west-2	Customer manag…	Permissions policy (1)



○	⊞	CodeBuildCloudWatchLogsPolicy-eks-devops-cb-for-pipe-us-west-2	Customer manag…	Permissions policy (1)
○	⊞	ecs7220-test-ecsdemo-frontend-AddonsStack-1WN4L7C6ZJ0UB-SubnetsAccessPolicy-1HKVWDHD1B5…	Customer manag…	Permissions policy (1)
○	⊞	eks-codebuild-sts-assume-role	Customer manag…	Permissions policy (1)
○	⊞	start-pipeline-execution-us-west-2-eks-rockstar-pipe	Customer manag…	Permissions policy (1)

AWS CODE PIPELINE

- Create a pipeline and leave to default role
- Source Provider: AWS CodeCommit
- Change Detection Options: CloudWatch Events
- Environment Image: Managed Image
- Operating System: Amazon Linux 2
- Environment Variables: REPOSITORY_URI, EKS_KUBECTL_ROLE_ARN, EKS_CLUSTER_NAME



General

Pipeline name
eks-rockstar-pipe

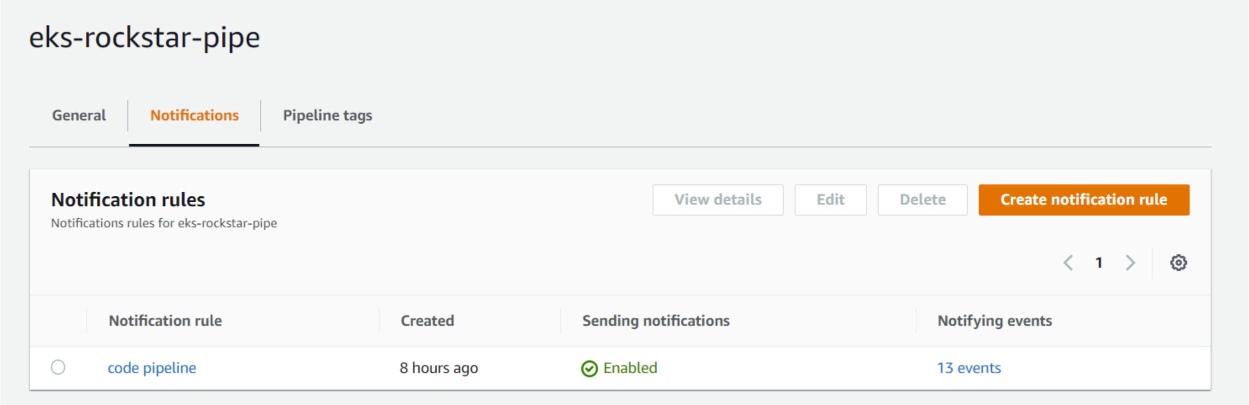
Pipeline ARN
arn:aws:codepipeline:us-west-2:220553481290:eks-rockstar-pipe

Service role ARN
arn:aws:iam::220553481290:role/service-role/AWSCodePipelineServiceRole-us-west-2-eks-rockstar-pipe

Version
1

Artifact store

Region	Type	Location
us-west-2	S3	codepipeline-us-west-2-280467212673

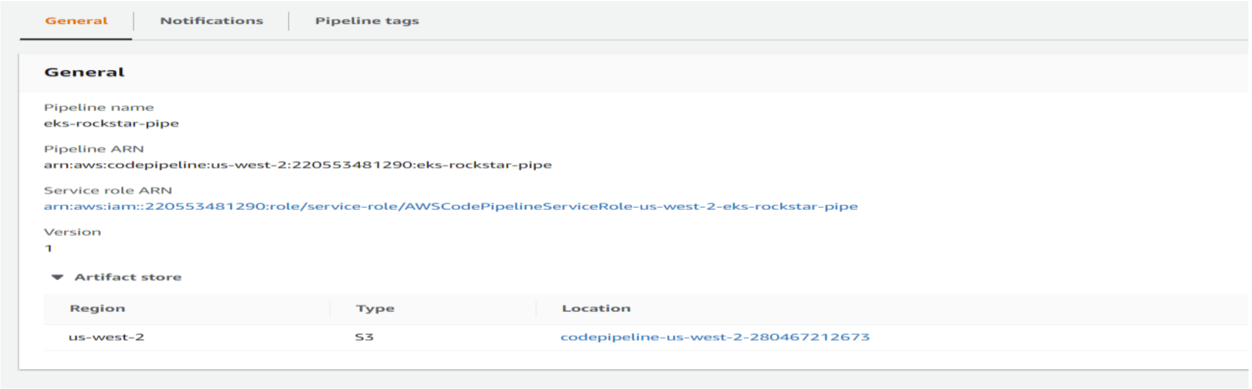


eks-rockstar-pipe

Notification rules

Notifications rules for eks-rockstar-pipe

	Notification rule	Created	Sending notifications	Notifying events
○	code pipeline	8 hours ago	✓ Enabled	13 events



General

Pipeline name
eks-rockstar-pipe

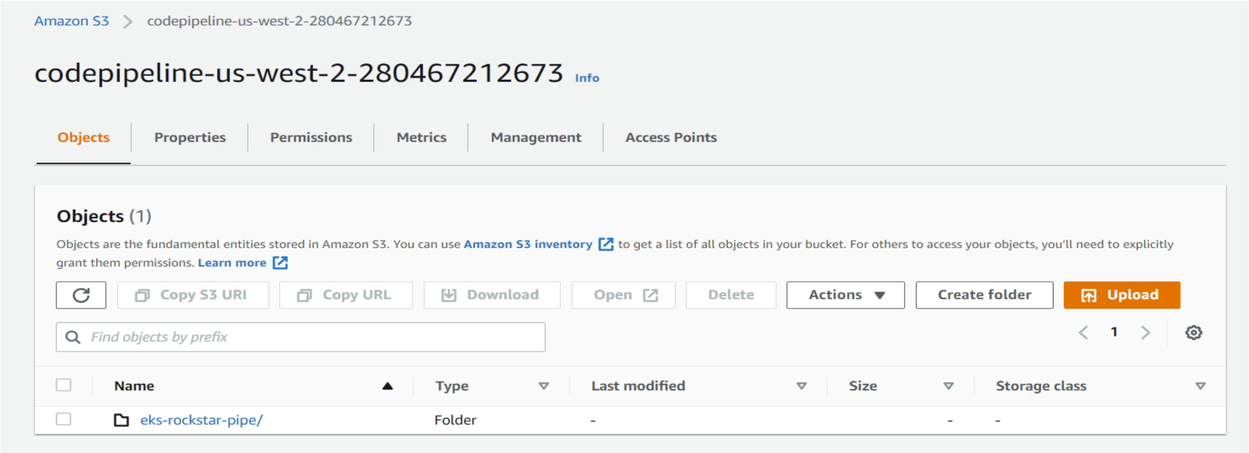
Pipeline ARN
arn:aws:codepipeline:us-west-2:220553481290:eks-rockstar-pipe

Service role ARN
arn:aws:iam::220553481290:role/service-role/AWSCodePipelineServiceRole-us-west-2-eks-rockstar-pipe

Version
1

Artifact store

Region	Type	Location
us-west-2	S3	codepipeline-us-west-2-280467212673

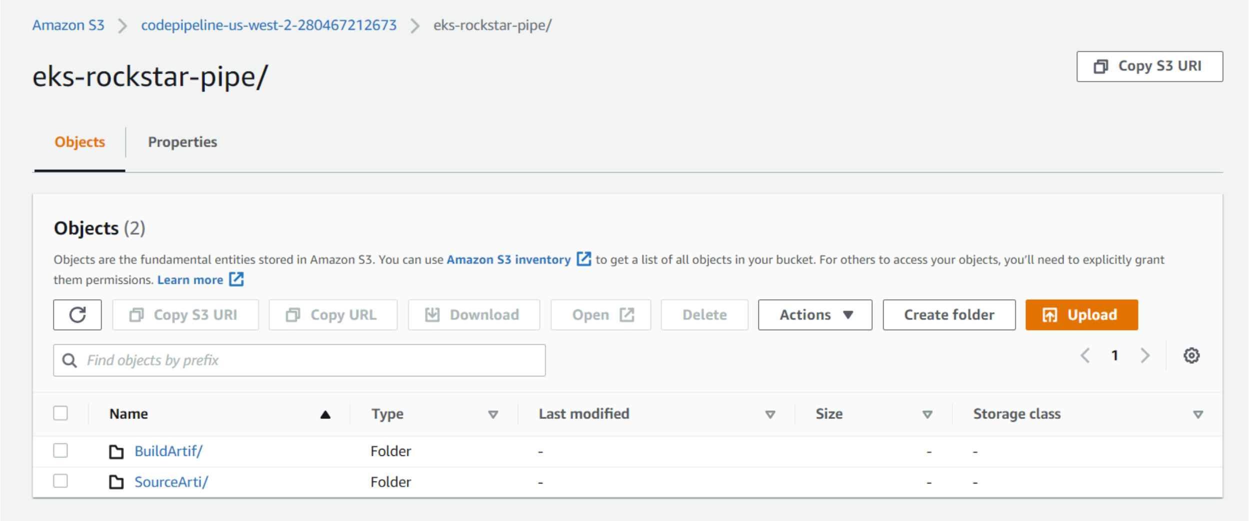


Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class
eks-rockstar-pipe/	Folder	-	-	-

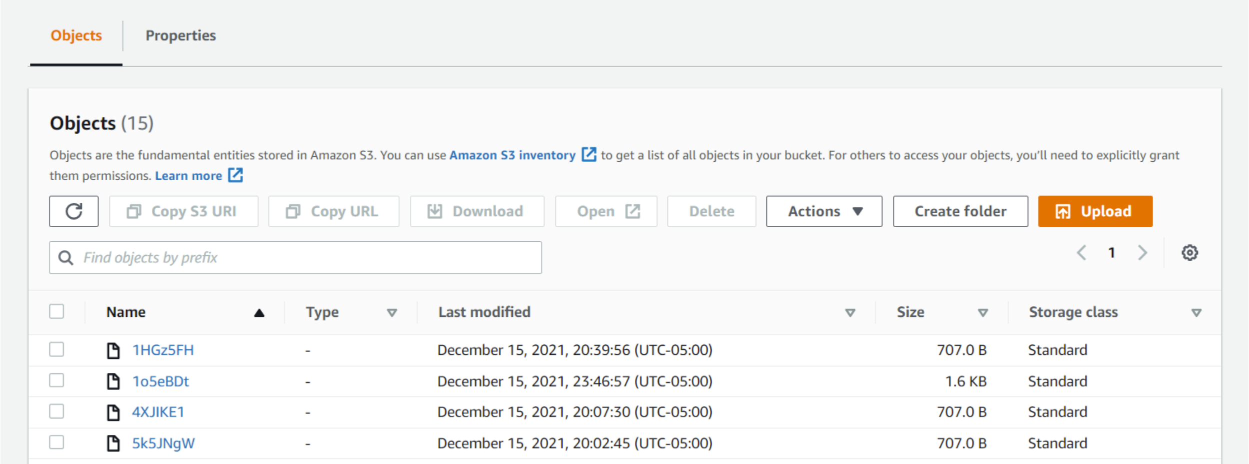
# Aritifacts Stored on S3:



Objects (2)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class
BuildArtif/	Folder	-	-	-
SourceArti/	Folder	-	-	-



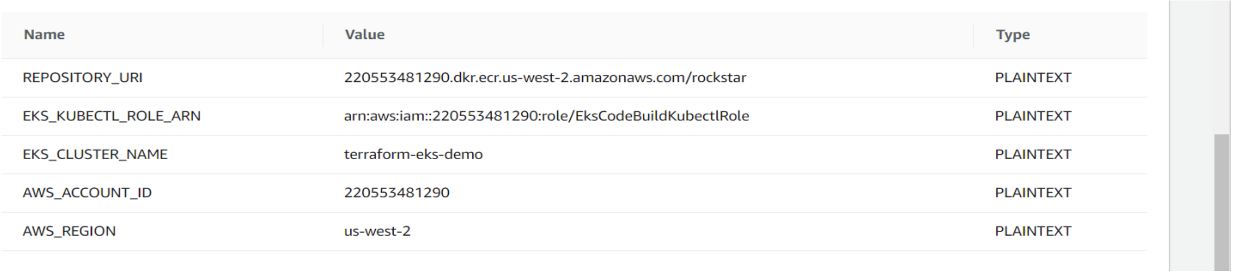
Objects (15)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class
1HGz5FH	-	December 15, 2021, 20:39:56 (UTC-05:00)	707.0 B	Standard
1o5eBDt	-	December 15, 2021, 23:46:57 (UTC-05:00)	1.6 KB	Standard
4XJIKE1	-	December 15, 2021, 20:07:30 (UTC-05:00)	707.0 B	Standard
5k5JNgW	-	December 15, 2021, 20:02:45 (UTC-05:00)	707.0 B	Standard

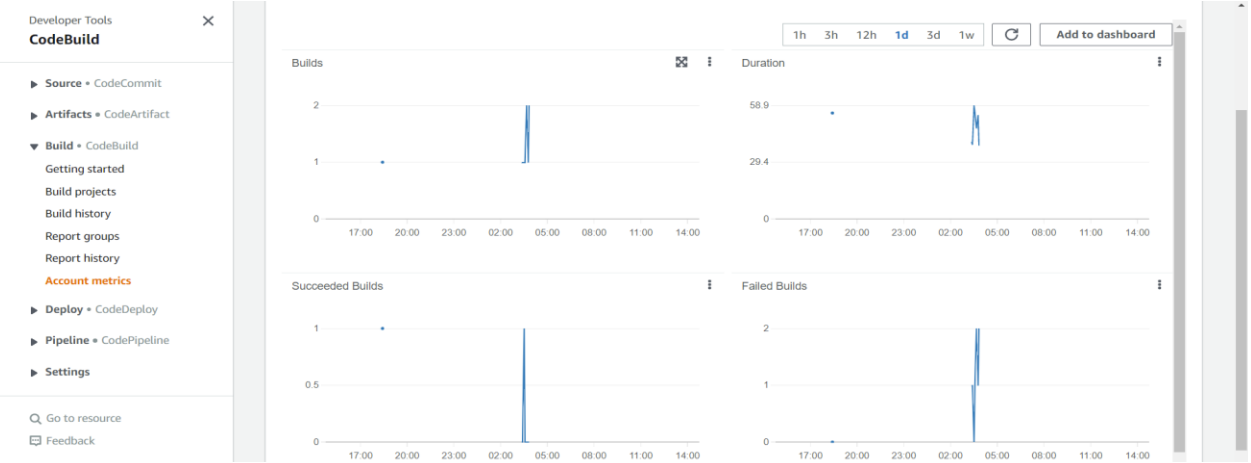
# AWS CODEBUILD

- Following Environment variables should be passed to the code build,

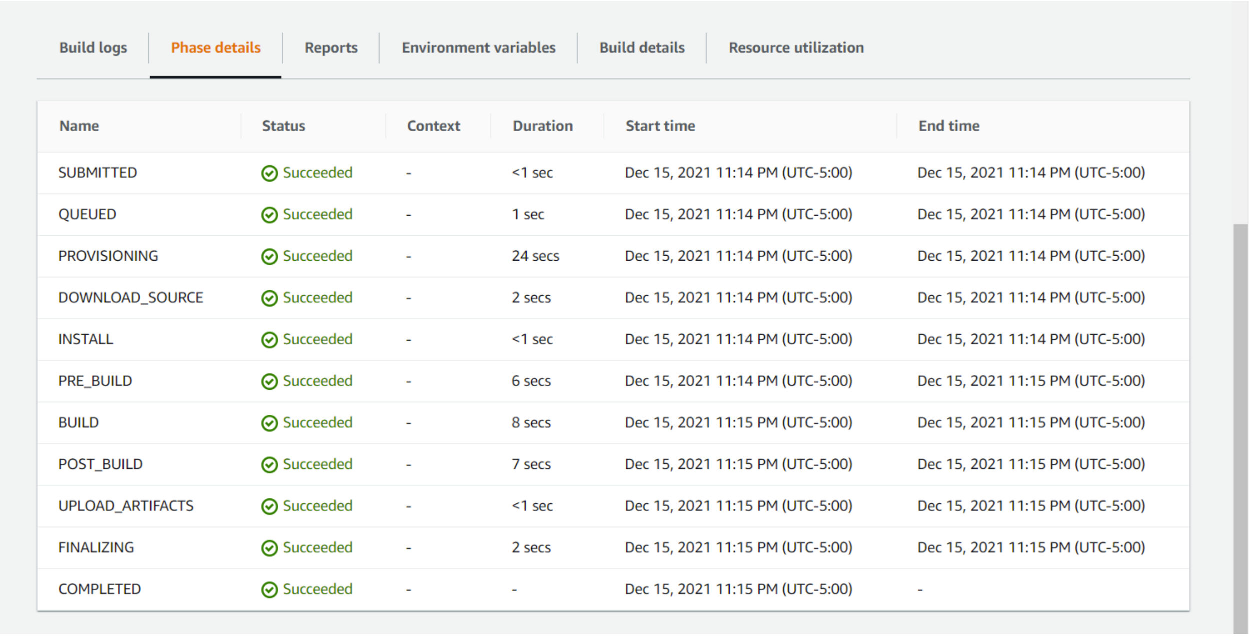


Name	Value	Type
REPOSITORY_URI	220553481290.dkr.ecr.us-west-2.amazonaws.com/rockstar	PLAINTEXT
EKS_KUBECTL_ROLE_ARN	arn:aws:iam::220553481290:role/EksCodeBuildKubectlRole	PLAINTEXT
EKS_CLUSTER_NAME	terraform-eks-demo	PLAINTEXT
AWS_ACCOUNT_ID	220553481290	PLAINTEXT
AWS_REGION	us-west-2	PLAINTEXT

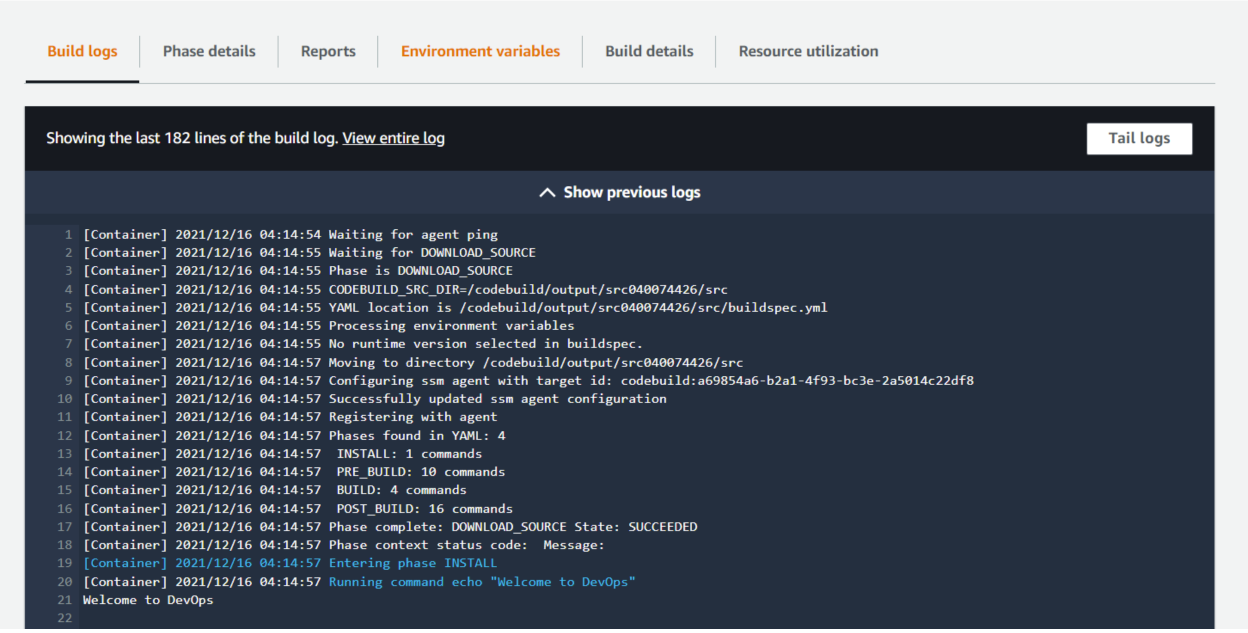
- CodeBuild Metrics



- Build Phase Details

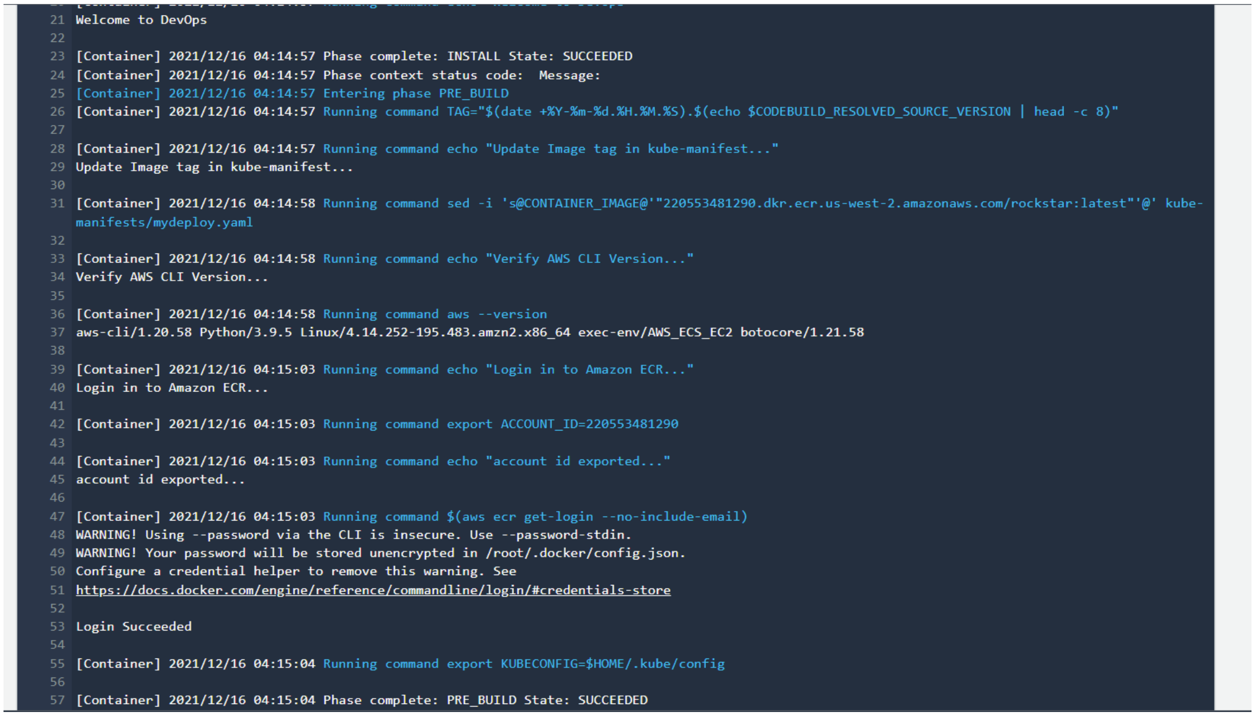


Name	Status	Context	Duration	Start time	End time
SUBMITTED	✓ Succeeded	-	<1 sec	Dec 15, 2021 11:14 PM (UTC-5:00)	Dec 15, 2021 11:14 PM (UTC-5:00)
QUEUED	✓ Succeeded	-	1 sec	Dec 15, 2021 11:14 PM (UTC-5:00)	Dec 15, 2021 11:14 PM (UTC-5:00)
PROVISIONING	✓ Succeeded	-	24 secs	Dec 15, 2021 11:14 PM (UTC-5:00)	Dec 15, 2021 11:14 PM (UTC-5:00)
DOWNLOAD_SOURCE	✓ Succeeded	-	2 secs	Dec 15, 2021 11:14 PM (UTC-5:00)	Dec 15, 2021 11:14 PM (UTC-5:00)
INSTALL	✓ Succeeded	-	<1 sec	Dec 15, 2021 11:14 PM (UTC-5:00)	Dec 15, 2021 11:14 PM (UTC-5:00)
PRE_BUILD	✓ Succeeded	-	6 secs	Dec 15, 2021 11:14 PM (UTC-5:00)	Dec 15, 2021 11:15 PM (UTC-5:00)
BUILD	✓ Succeeded	-	8 secs	Dec 15, 2021 11:15 PM (UTC-5:00)	Dec 15, 2021 11:15 PM (UTC-5:00)
POST_BUILD	✓ Succeeded	-	7 secs	Dec 15, 2021 11:15 PM (UTC-5:00)	Dec 15, 2021 11:15 PM (UTC-5:00)
UPLOAD_ARTIFACTS	✓ Succeeded	-	<1 sec	Dec 15, 2021 11:15 PM (UTC-5:00)	Dec 15, 2021 11:15 PM (UTC-5:00)
FINALIZING	✓ Succeeded	-	2 secs	Dec 15, 2021 11:15 PM (UTC-5:00)	Dec 15, 2021 11:15 PM (UTC-5:00)
COMPLETED	✓ Succeeded	-	-	Dec 15, 2021 11:15 PM (UTC-5:00)	-

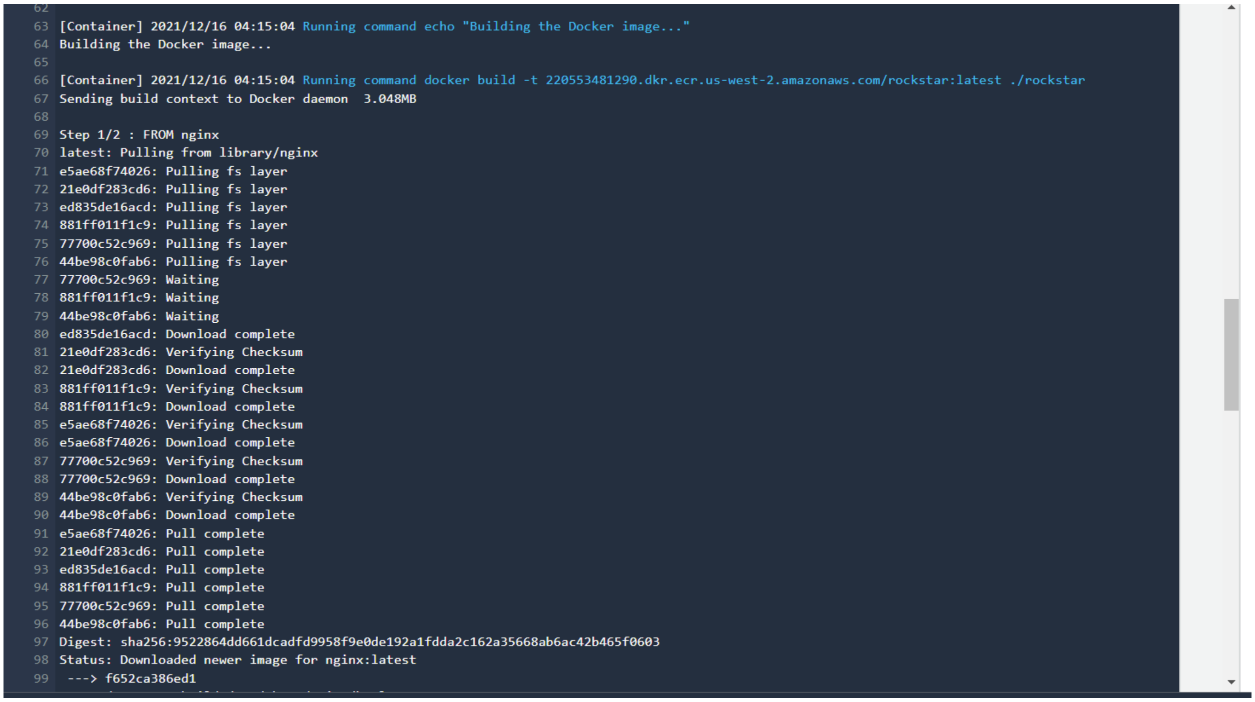


Showing the last 182 lines of the build log. View entire log

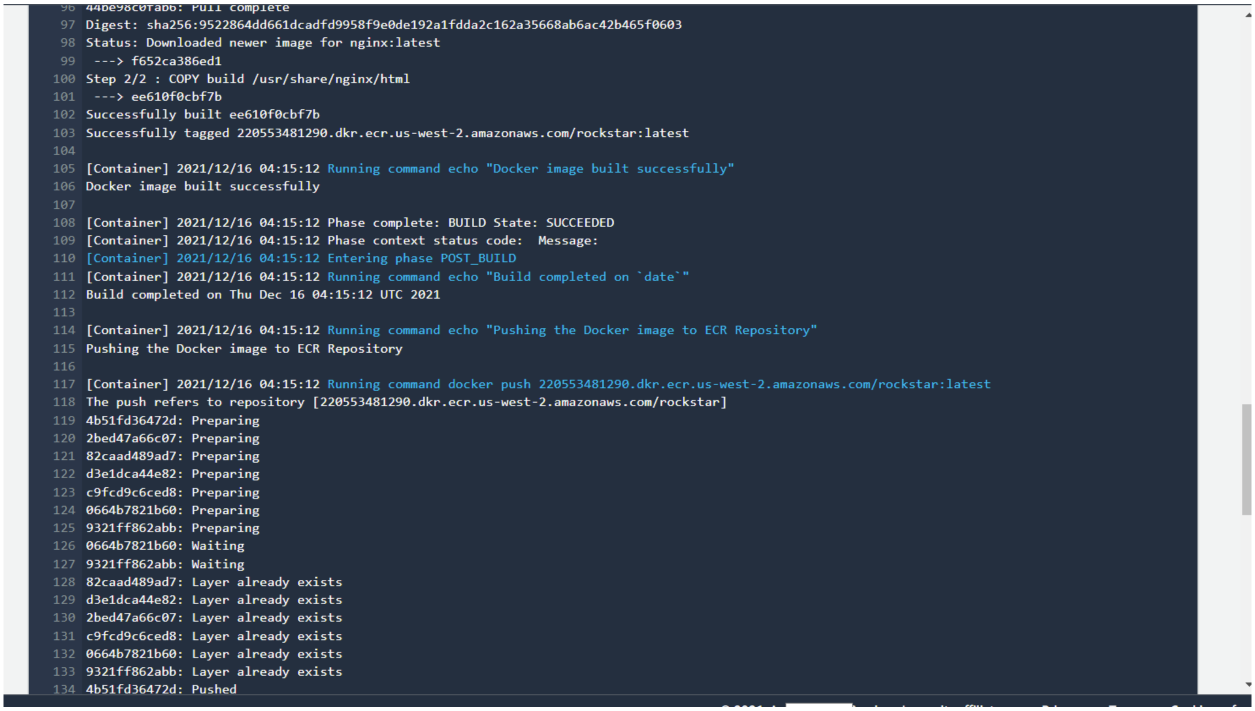
```
[Container] 2021/12/16 04:14:54 Waiting for agent ping
[Container] 2021/12/16 04:14:55 Waiting for DOWNLOAD_SOURCE
[Container] 2021/12/16 04:14:55 Phase is DOWNLOAD_SOURCE
[Container] 2021/12/16 04:14:55 CODEBUILD_SRC_DIR=/codebuild/output/src040074426/src
[Container] 2021/12/16 04:14:55 YAML location is /codebuild/output/src040074426/src/buildspec.yml
[Container] 2021/12/16 04:14:55 Processing environment variables
[Container] 2021/12/16 04:14:55 No runtime version selected in buildspec.
[Container] 2021/12/16 04:14:57 Moving to directory /codebuild/output/src040074426/src
[Container] 2021/12/16 04:14:57 Configuring ssm agent with target id: codebuild:a69854a6-b2a1-4f93-bc3e-2a5014c22df8
[Container] 2021/12/16 04:14:57 Successfully updated ssm agent configuration
[Container] 2021/12/16 04:14:57 Registering with agent
[Container] 2021/12/16 04:14:57 Phases found in YAML: 4
[Container] 2021/12/16 04:14:57  INSTALL: 1 commands
[Container] 2021/12/16 04:14:57  PRE_BUILD: 10 commands
[Container] 2021/12/16 04:14:57  BUILD: 4 commands
[Container] 2021/12/16 04:14:57  POST_BUILD: 16 commands
[Container] 2021/12/16 04:14:57 Phase complete: DOWNLOAD_SOURCE State: SUCCEEDED
[Container] 2021/12/16 04:14:57 Phase context status code:  Message:
[Container] 2021/12/16 04:14:57 Entering phase INSTALL
[Container] 2021/12/16 04:14:57 Running command echo "Welcome to DevOps"
Welcome to DevOps

```



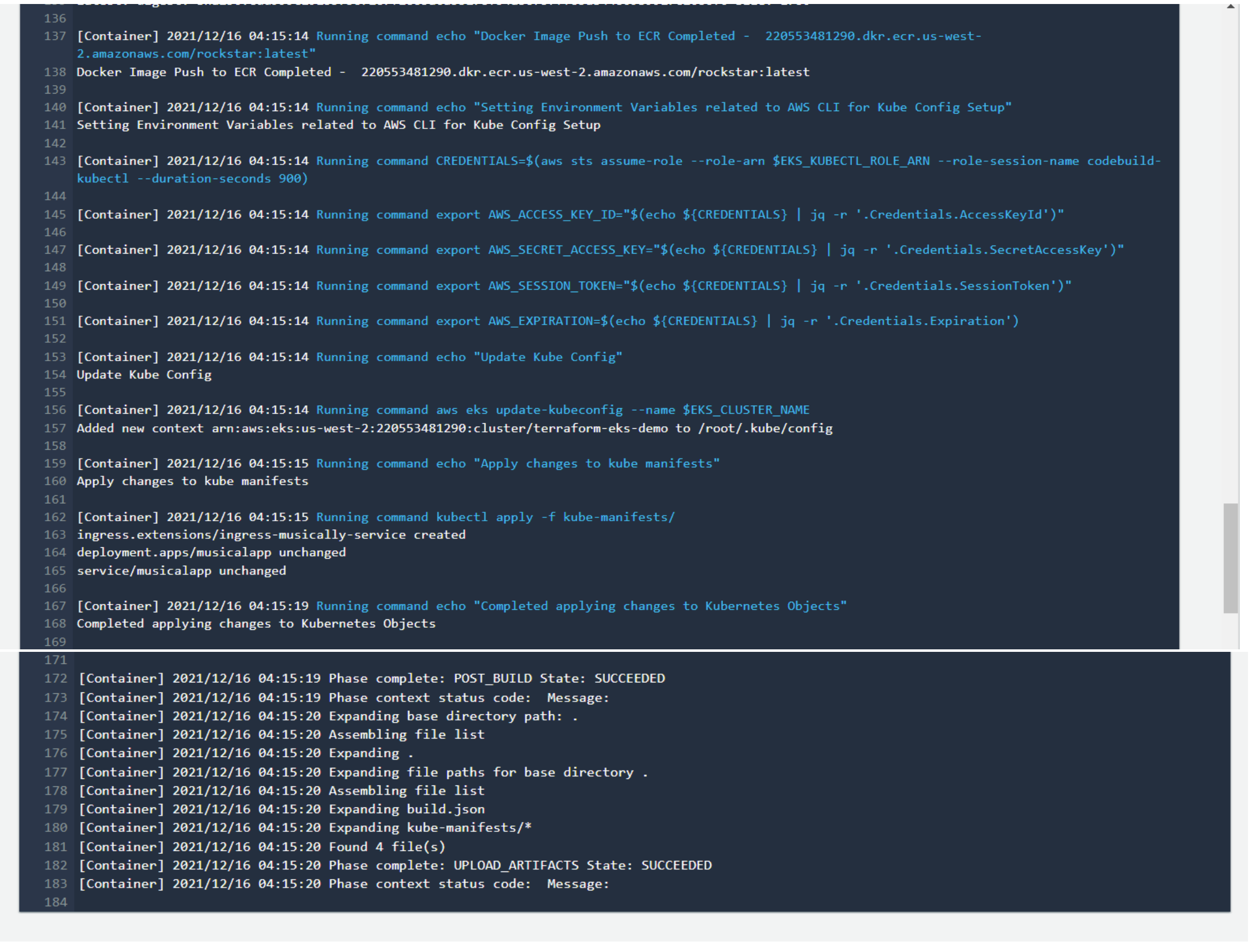
```
Welcome to DevOps

[Container] 2021/12/16 04:14:57 Phase complete: INSTALL State: SUCCEEDED
[Container] 2021/12/16 04:14:57 Phase context status code:  Message:
[Container] 2021/12/16 04:14:57 Entering phase PRE_BUILD
[Container] 2021/12/16 04:14:57 Running command TAG="$(date +%Y-%m-%d.%H.%M.%S).$(echo $CODEBUILD_RESOLVED_SOURCE_VERSION | head -c 8)"

[Container] 2021/12/16 04:14:57 Running command echo "Update Image tag in kube-manifest..."
Update Image tag in kube-manifest...

[Container] 2021/12/16 04:14:58 Running command sed -i 's@CONTAINER_IMAGE@'"220553481290.dkr.ecr.us-west-2.amazonaws.com/rockstar:latest"'@' kube-
manifests/mydeploy.yaml

[Container] 2021/12/16 04:14:58 Running command echo "Verify AWS CLI Version..."
Verify AWS CLI Version...

[Container] 2021/12/16 04:14:58 Running command aws --version
aws-cli/1.20.58 Python/3.9.5 Linux/4.14.252-195.483.amzn2.x86_64 exec-env/AWS_ECS_EC2 botocore/1.21.58

[Container] 2021/12/16 04:15:03 Running command echo "Login in to Amazon ECR..."
Login in to Amazon ECR...

[Container] 2021/12/16 04:15:03 Running command export ACCOUNT_ID=220553481290

[Container] 2021/12/16 04:15:03 Running command echo "account id exported..."
account id exported...

[Container] 2021/12/16 04:15:03 Running command $(aws ecr get-login --no-include-email)
WARNING! Using --password via the CLI is insecure. Use --password-stdin.
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded

[Container] 2021/12/16 04:15:04 Running command export KUBECONFIG=$HOME/.kube/config

[Container] 2021/12/16 04:15:04 Phase complete: PRE_BUILD State: SUCCEEDED
```



```
[Container] 2021/12/16 04:15:04 Running command echo "Building the Docker image..."
Building the Docker image...

[Container] 2021/12/16 04:15:04 Running command docker build -t 220553481290.dkr.ecr.us-west-2.amazonaws.com/rockstar:latest ./rockstar
Sending build context to Docker daemon  3.048MB

Step 1/2 : FROM nginx
latest: Pulling from library/nginx
e5ae68f74026: Pulling fs layer
21e0df283cd6: Pulling fs layer
ed835de16acd: Pulling fs layer
881ff011f1c9: Pulling fs layer
77700c52c969: Pulling fs layer
44be98c0fab6: Pulling fs layer
77700c52c969: Waiting
881ff011f1c9: Waiting
44be98c0fab6: Waiting
ed835de16acd: Download complete
21e0df283cd6: Verifying Checksum
21e0df283cd6: Download complete
881ff011f1c9: Verifying Checksum
881ff011f1c9: Download complete
e5ae68f74026: Verifying Checksum
e5ae68f74026: Download complete
77700c52c969: Verifying Checksum
77700c52c969: Download complete
44be98c0fab6: Verifying Checksum
44be98c0fab6: Download complete
e5ae68f74026: Pull complete
21e0df283cd6: Pull complete
ed835de16acd: Pull complete
881ff011f1c9: Pull complete
77700c52c969: Pull complete
44be98c0fab6: Pull complete
Digest: sha256:9522864dd661dcadfd9958f9e0de192a1fdda2c162a35668ab6ac42b465f0603
Status: Downloaded newer image for nginx:latest
 ---> f652ca386ed1
```

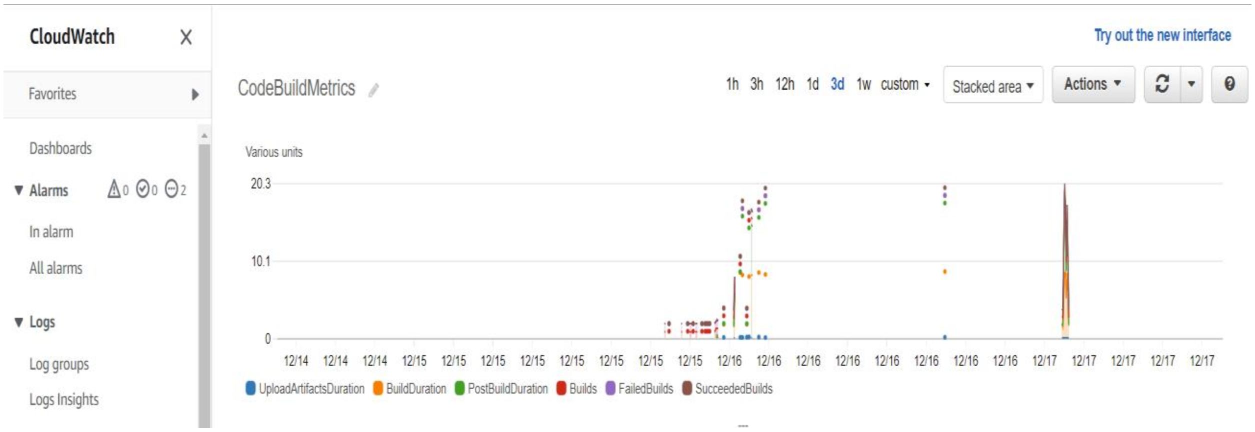


```
Digest: sha256:9522864dd661dcadfd9958f9e0de192a1fdda2c162a35668ab6ac42b465f0603
Status: Downloaded newer image for nginx:latest
 ---> f652ca386ed1
Step 2/2 : COPY build /usr/share/nginx/html
 ---> ee610f0cbf7b
Successfully built ee610f0cbf7b
Successfully tagged 220553481290.dkr.ecr.us-west-2.amazonaws.com/rockstar:latest

[Container] 2021/12/16 04:15:12 Running command echo "Docker image built successfully"
Docker image built successfully

[Container] 2021/12/16 04:15:12 Phase complete: BUILD State: SUCCEEDED
[Container] 2021/12/16 04:15:12 Phase context status code:  Message:
[Container] 2021/12/16 04:15:12 Entering phase POST_BUILD
[Container] 2021/12/16 04:15:12 Running command echo "Build completed on `date`"
Build completed on Thu Dec 16 04:15:12 UTC 2021

[Container] 2021/12/16 04:15:12 Running command echo "Pushing the Docker image to ECR Repository"
Pushing the Docker image to ECR Repository

[Container] 2021/12/16 04:15:12 Running command docker push 220553481290.dkr.ecr.us-west-2.amazonaws.com/rockstar:latest
The push refers to repository [220553481290.dkr.ecr.us-west-2.amazonaws.com/rockstar]
4b51fd36472d: Preparing
2bed47a66c07: Preparing
82caad489ad7: Preparing
d3e1dca44e82: Preparing
c9fcd9c6ced8: Preparing
0664b7821b60: Preparing
9321ff862abb: Preparing
0664b7821b60: Waiting
9321ff862abb: Waiting
82caad489ad7: Layer already exists
d3e1dca44e82: Layer already exists
2bed47a66c07: Layer already exists
c9fcd9c6ced8: Layer already exists
0664b7821b60: Layer already exists
9321ff862abb: Layer already exists
4b51fd36472d: Pushed
```



```

[Container] 2021/12/16 04:15:14 Running command echo "Docker Image Push to ECR Completed -  220553481290.dkr.ecr.us-west-
2.amazonaws.com/rockstar:latest"
Docker Image Push to ECR Completed -  220553481290.dkr.ecr.us-west-2.amazonaws.com/rockstar:latest

[Container] 2021/12/16 04:15:14 Running command echo "Setting Environment Variables related to AWS CLI for Kube Config Setup"
Setting Environment Variables related to AWS CLI for Kube Config Setup

[Container] 2021/12/16 04:15:14 Running command CREDENTIALS=$(aws sts assume-role --role-arn $EKS_KUBECTL_ROLE_ARN --role-session-name codebuild-
kubectl --duration-seconds 900)

[Container] 2021/12/16 04:15:14 Running command export AWS_ACCESS_KEY_ID="$(echo ${CREDENTIALS} | jq -r '.Credentials.AccessKeyId')"

[Container] 2021/12/16 04:15:14 Running command export AWS_SECRET_ACCESS_KEY="$(echo ${CREDENTIALS} | jq -r '.Credentials.SecretAccessKey')"

[Container] 2021/12/16 04:15:14 Running command export AWS_SESSION_TOKEN="$(echo ${CREDENTIALS} | jq -r '.Credentials.SessionToken')"

[Container] 2021/12/16 04:15:14 Running command export AWS_EXPIRATION=$(echo ${CREDENTIALS} | jq -r '.Credentials.Expiration')

[Container] 2021/12/16 04:15:14 Running command echo "Update Kube Config"
Update Kube Config

[Container] 2021/12/16 04:15:14 Running command aws eks update-kubeconfig --name $EKS_CLUSTER_NAME
Added new context arn:aws:eks:us-west-2:220553481290:cluster/terraform-eks-demo to /root/.kube/config

[Container] 2021/12/16 04:15:15 Running command echo "Apply changes to kube manifests"
Apply changes to kube manifests

[Container] 2021/12/16 04:15:15 Running command kubectl apply -f kube-manifests/
ingress.extensions/ingress-musically-service created
deployment.apps/musicalapp unchanged
service/musicalapp unchanged

[Container] 2021/12/16 04:15:19 Running command echo "Completed applying changes to Kubernetes Objects"
Completed applying changes to Kubernetes Objects


[Container] 2021/12/16 04:15:19 Phase complete: POST_BUILD State: SUCCEEDED
[Container] 2021/12/16 04:15:19 Phase context status code:  Message:
[Container] 2021/12/16 04:15:20 Expanding base directory path: .
[Container] 2021/12/16 04:15:20 Assembling file list
[Container] 2021/12/16 04:15:20 Expanding .
[Container] 2021/12/16 04:15:20 Expanding file paths for base directory .
[Container] 2021/12/16 04:15:20 Assembling file list
[Container] 2021/12/16 04:15:20 Expanding build.json
[Container] 2021/12/16 04:15:20 Expanding kube-manifests/*
[Container] 2021/12/16 04:15:20 Found 4 file(s)
[Container] 2021/12/16 04:15:20 Phase complete: UPLOAD_ARTIFACTS State: SUCCEEDED
[Container] 2021/12/16 04:15:20 Phase context status code:  Message:

```

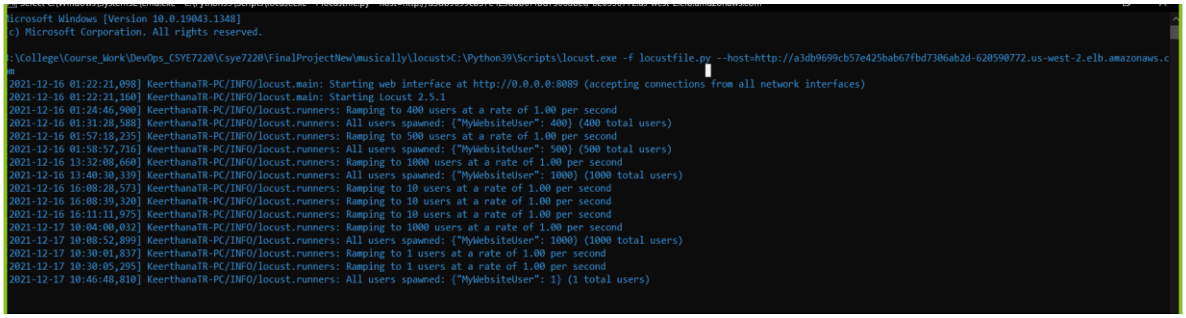
# Cloud Watch Metrics:



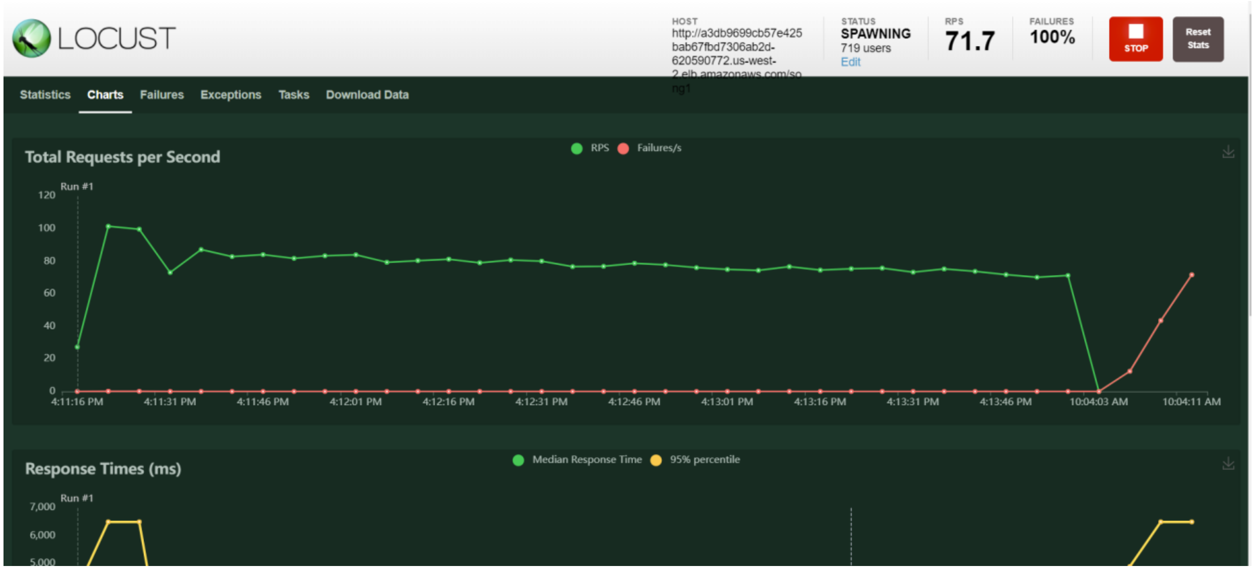
# Locust – Load Testing

LOCUST INSTALLATION STEPS:

- pip install -U setuptools
- pip install –U locust
- pip uninstall locust
- C:\Python39\Scripts\locust.exe -f locustfile.py --host=http://a3db9699cb57e425bab67fbd7306ab2d-620590772.us-west-2.elb.amazonaws.com
- Number of Users – 1000, hatch rate – 1ps



```
icrosoft Windows [Version 10.0.19043.1348]
c) Microsoft Corporation. All rights reserved.

:\College\Course_Work\DevOps_CSYE7220\Csye7220\FinalProjectNew\musically\locust>C:\Python39\Scripts\locust.exe -f locustfile.py --host=http://a3db9699cb57e425bab67fbd7306ab2d-620590772.us-west-2.elb.amazonaws.c
m
2021-12-16 01:22:21,098] KeerthanaTR-PC/INFO/locust.main: Starting web interface at http://0.0.0.0:8089 (accepting connections from all network interfaces)
2021-12-16 01:22:21,160] KeerthanaTR-PC/INFO/locust.main: Starting Locust 2.5.1
2021-12-16 01:24:46,900] KeerthanaTR-PC/INFO/locust.runners: Ramping to 400 users at a rate of 1.00 per second
2021-12-16 01:31:28,588] KeerthanaTR-PC/INFO/locust.runners: All users spawned: {"MyWebsiteUser": 400} (400 total users)
2021-12-16 01:57:18,235] KeerthanaTR-PC/INFO/locust.runners: Ramping to 500 users at a rate of 1.00 per second
2021-12-16 01:58:57,716] KeerthanaTR-PC/INFO/locust.runners: All users spawned: {"MyWebsiteUser": 500} (500 total users)
2021-12-16 13:32:08,660] KeerthanaTR-PC/INFO/locust.runners: Ramping to 1000 users at a rate of 1.00 per second
2021-12-16 13:40:30,339] KeerthanaTR-PC/INFO/locust.runners: All users spawned: {"MyWebsiteUser": 1000} (1000 total users)
2021-12-16 16:08:28,573] KeerthanaTR-PC/INFO/locust.runners: Ramping to 10 users at a rate of 1.00 per second
2021-12-16 16:08:39,320] KeerthanaTR-PC/INFO/locust.runners: Ramping to 10 users at a rate of 1.00 per second
2021-12-16 16:11:11,975] KeerthanaTR-PC/INFO/locust.runners: Ramping to 10 users at a rate of 1.00 per second
2021-12-17 10:04:00,032] KeerthanaTR-PC/INFO/locust.runners: Ramping to 1000 users at a rate of 1.00 per second
2021-12-17 10:08:52,899] KeerthanaTR-PC/INFO/locust.runners: All users spawned: {"MyWebsiteUser": 1000} (1000 total users)
2021-12-17 10:30:01,837] KeerthanaTR-PC/INFO/locust.runners: Ramping to 1 users at a rate of 1.00 per second
2021-12-17 10:30:05,295] KeerthanaTR-PC/INFO/locust.runners: Ramping to 1 users at a rate of 1.00 per second
2021-12-17 10:46:48,810] KeerthanaTR-PC/INFO/locust.runners: All users spawned: {"MyWebsiteUser": 1} (1 total users)
```



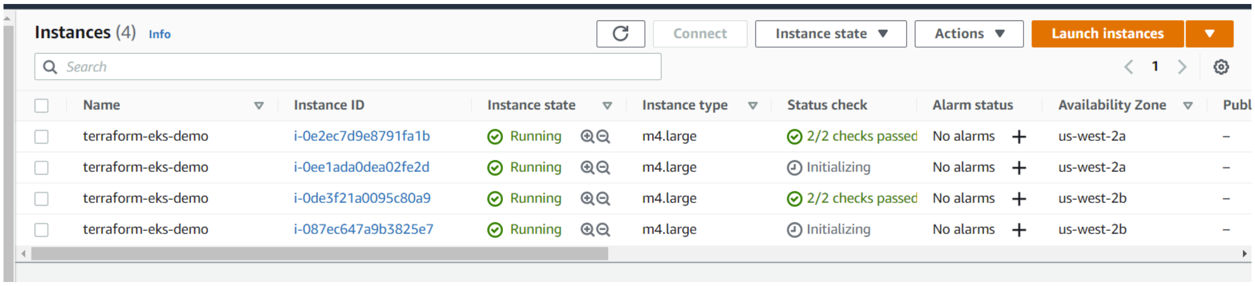
HOST http://a3db9699cb57e425bab67fbd7306ab2d-620590772.us-west-2.elb.amazonaws.com/song1

STATUS SPAWNING 719 users

RPS 71.7

FAILURES 100%

# Auto Scale of nodes:



Instances (4) Info

□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Publ
□	terraform-eks-demo	i-0e2ec7d9e8791fa1b	✓ Running	m4.large	✓ 2/2 checks passed	No alarms +	us-west-2a	–
□	terraform-eks-demo	i-0ee1ada0dea02fe2d	✓ Running	m4.large	Initializing	No alarms +	us-west-2a	–
□	terraform-eks-demo	i-0de3f21a0095c80a9	✓ Running	m4.large	✓ 2/2 checks passed	No alarms +	us-west-2b	–
□	terraform-eks-demo	i-087ec647a9b3825e7	✓ Running	m4.large	Initializing	No alarms +	us-west-2b	–

# GRAFANA - Prometheus

# Installation Steps:

- kubectl get pods --all-namespaces
- helm repo add prometheus-community https://prometheus-community.github.io/helm-charts
- helm repo add grafana https://grafana.github.io/helm-charts
- kubectl create namespace prometheus
- kubectl get namespaces

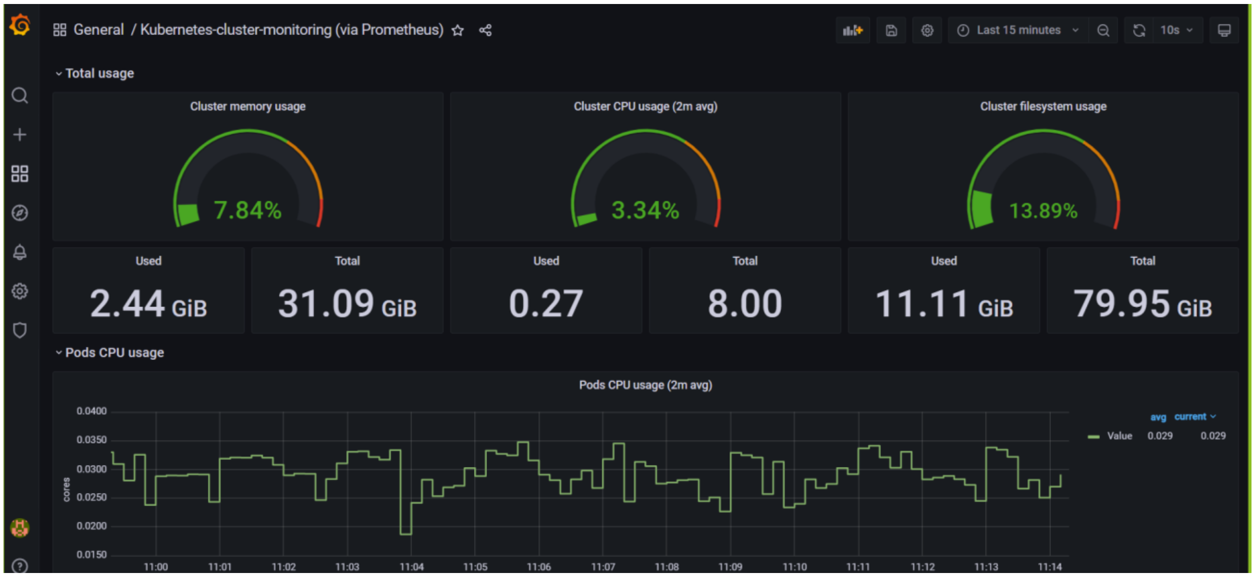
- helm install prometheus prometheus-community/prometheus --namespace prometheus --set alertmanager.persistentVolume.storageClass="gp2" --set server.persistentVolume.storageClass="gp2"

- kubectl get all -n prometheus
- kubectl port-forward -n prometheus deploy/prometheus-server 8080:9091

- helm install grafana grafana/grafana --namespace grafana --set persistence.storageClassName="gp2" --set persistence.enabled=true --set adminPassword='grafana!' --values ./grafana.yaml --set service.type=LoadBalancer

- kubectl get all -n Grafana
- kubectl get svc -n grafana

- c:\users\keert\appdata\roaming\python\python39\scripts\locust.exe -f locustfile.py --host=http://a3db9699cb57e425bab67fbd7306ab2d-620590772.us-west-2.elb.amazonaws.com



General / Kubernetes-cluster-monitoring (via Prometheus)

Last 15 minutes · 10s

Total usage

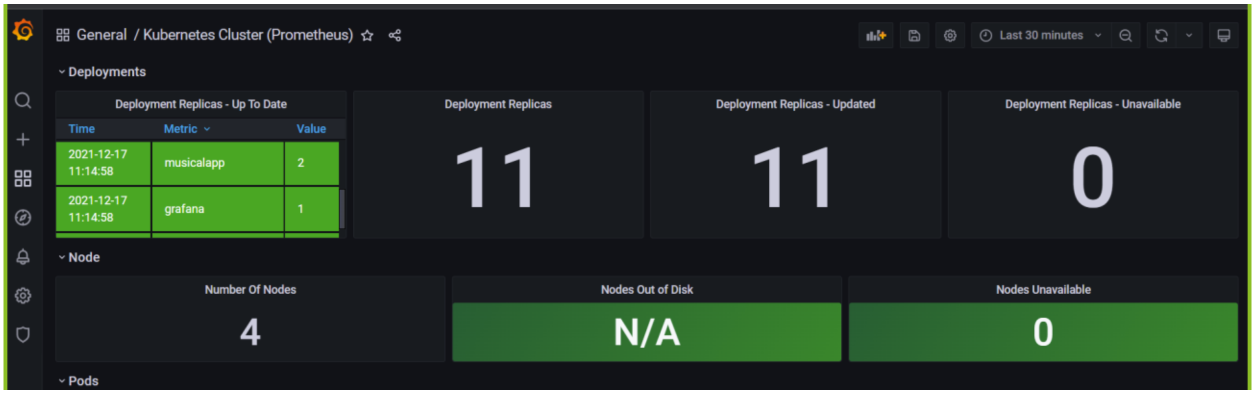
Cluster memory usage: 7.84% — Used 2.44 GiB, Total 31.09 GiB

Cluster CPU usage (2m avg): 3.34% — Used 0.27, Total 8.00

Cluster filesystem usage: 13.89% — Used 11.11 GiB, Total 79.95 GiB

Pods CPU usage

Pods CPU usage (2m avg)



Deployments

Deployment Replicas - Up To Date

Time	Metric	Value
2021-12-17 11:14:58	musicalapp	2
2021-12-17 11:14:58	grafana	1

Deployment Replicas: 11

Deployment Replicas - Updated: 11

Deployment Replicas - Unavailable: 0

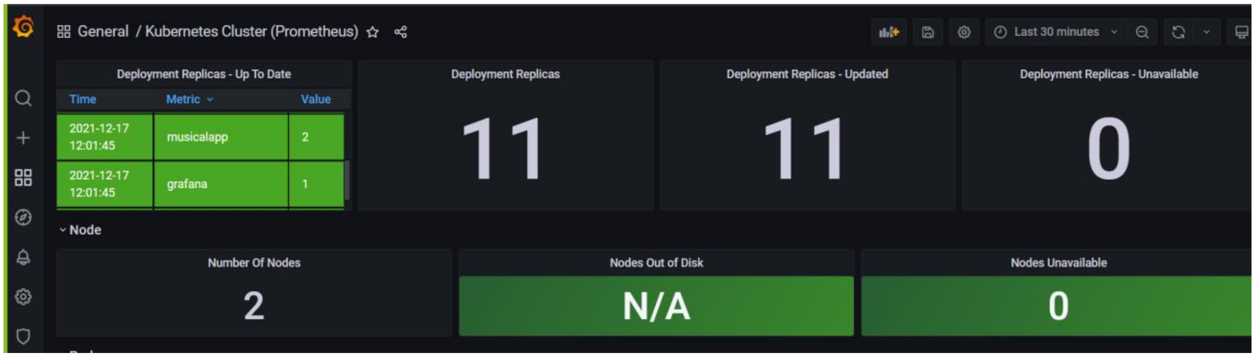
Node

Number Of Nodes: 4

Nodes Out of Disk: N/A

Nodes Unavailable: 0

Pods



Deployment Replicas - Up To Date

Time	Metric	Value
2021-12-17 12:01:45	musicalapp	2
2021-12-17 12:01:45	grafana	1

Deployment Replicas: 11

Deployment Replicas - Updated: 11

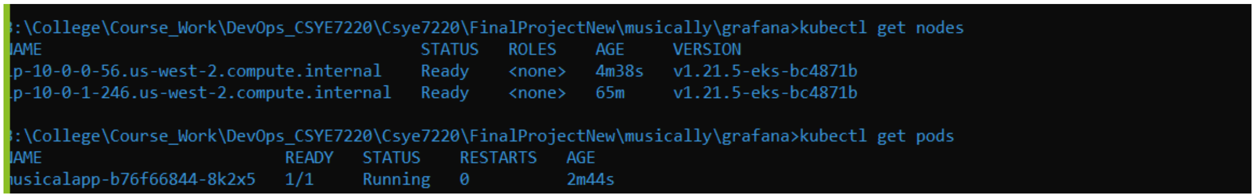
Deployment Replicas - Unavailable: 0

Node

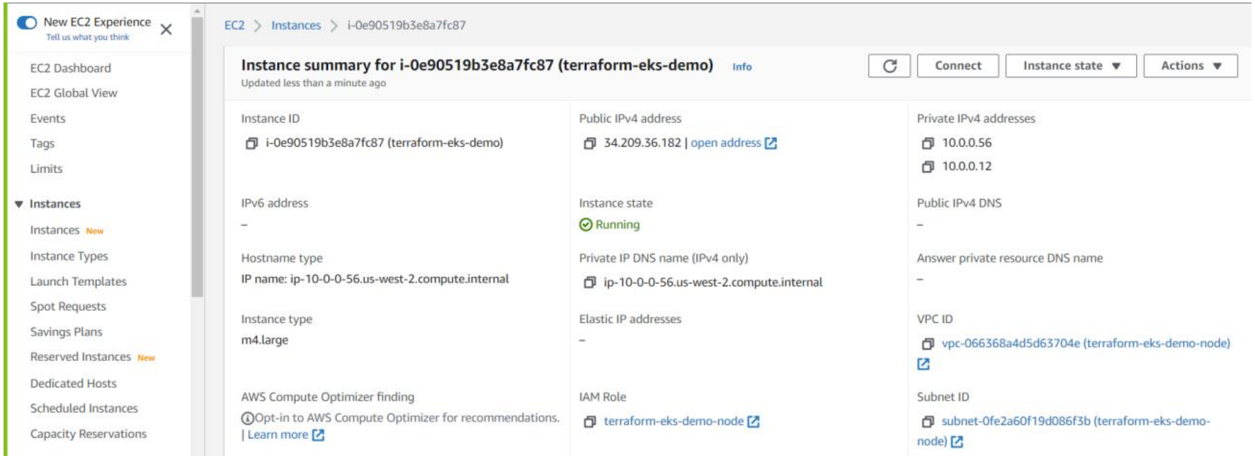
Number Of Nodes: 2

Nodes Out of Disk: N/A

Nodes Unavailable: 0



```
:\College\Course_Work\DevOps_CSYE7220\Csye7220\FinalProjectNew\musically\grafana>kubectl get nodes
IAME                                        STATUS   ROLES    AGE     VERSION
ip-10-0-0-56.us-west-2.compute.internal     Ready    <none>   4m38s   v1.21.5-eks-bc4871b
ip-10-0-1-246.us-west-2.compute.internal    Ready    <none>   65m     v1.21.5-eks-bc4871b

:\College\Course_Work\DevOps_CSYE7220\Csye7220\FinalProjectNew\musically\grafana>kubectl get pods
IAME                           READY   STATUS    RESTARTS   AGE
usicalapp-b76f66844-8k2x5      1/1     Running   0          2m44s
```



Instance summary for i-0e90519b3e8a7fc87 (terraform-eks-demo)

Updated less than a minute ago

Instance ID: i-0e90519b3e8a7fc87 (terraform-eks-demo)

Public IPv4 address: 34.209.36.182 | open address

Private IPv4 addresses: 10.0.0.56, 10.0.0.12

IPv6 address: –

Instance state: Running

Public IPv4 DNS: –

Hostname type: IP name: ip-10-0-0-56.us-west-2.compute.internal

Private IP DNS name (IPv4 only): ip-10-0-0-56.us-west-2.compute.internal

Answer private resource DNS name: –

Instance type: m4.large

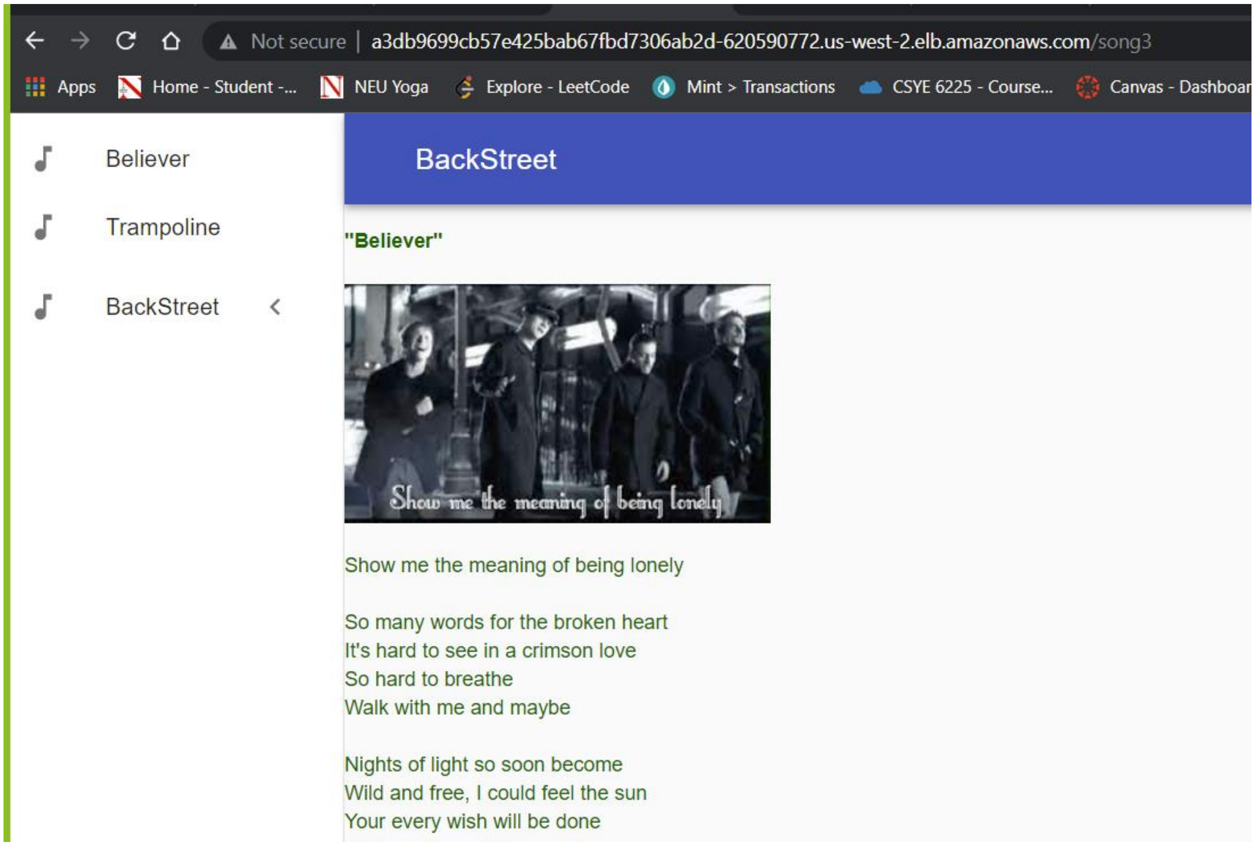
Elastic IP addresses: –

VPC ID: vpc-066368a4d5d63704e (terraform-eks-demo-node)

AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations. | Learn more

IAM Role: terraform-eks-demo-node

Subnet ID: subnet-0fe2a60f19d086f3b (terraform-eks-demo-node)



BackStreet

"Believer"

Show me the meaning of being lonely

So many words for the broken heart
It's hard to see in a crimson love
So hard to breathe
Walk with me and maybe

Nights of light so soon become
Wild and free, I could feel the sun
Your every wish will be done

References:

- https://docs.aws.amazon.com/codebuild/latest/userguide/build-env-ref-available.html
- https://github.com/aws/aws-codebuild-docker-images/blob/master/al2/x86_64/standard/3.0/Dockerfile
- **STS Assume Role:** https://docs.aws.amazon.com/cli/latest/reference/sts/assume-role.html
- https://docs.aws.amazon.com/IAM/latest/UserGuide/troubleshoot_roles.html