AWS_EKS_CI_CD_gitactions

Create EKS-cluster via Terraform

```
prakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/terraform-eks-greens$ aws configure --profile greens
AWS Access Key ID [****************HJNA]: AKIASPNC43WWWI4RK45G
AWS Secret Access Key [****************GTg4]: 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
Default region name [ap-south-1]:
Default output format [None]:
```

Go to terraform-eks-greens demo folder then enter
Terraform init
Terraform validate
Terraform plan
Terraform apply before that we need kubectl,aws-cli-v2,terraform latest version,github accounts that all

```
prakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/terraform-eks-greens$ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v4.31.0...
- Installed hashicorp/aws v4.31.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

```
prakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/terraform-eks-greens$ terraform validate
Success! The configuration is valid.
```

```
prakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/terraform-eks-greens$ terraform plan

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

Terraform will perform the following actions:

  # aws_cloudwatch_log_group.greens-cluster will be created
  + resource "aws_cloudwatch_log_group" "greens-cluster" {
      + arn               = (known after apply)
      + id                = (known after apply)
      + name              = "/aws/eks/greens-cluster/cluster"
      + retention_in_days = 7
      + tags_all          = (known after apply)
    }

  # aws_eks_cluster.greens-cluster-eks will be created
  + resource "aws_eks_cluster" "greens-cluster-eks" {
      + arn                   = (known after apply)
      + certificate_authority = (known after apply)
      + created_at            = (known after apply)
      + endpoint              = (known after apply)
      + id                    = (known after apply)
      + identity              = (known after apply)
      + name                  = "greens-cluster-cluster"
      + platform_version      = (known after apply)
      + role_arn              = (known after apply)
      + status                = (known after apply)
```

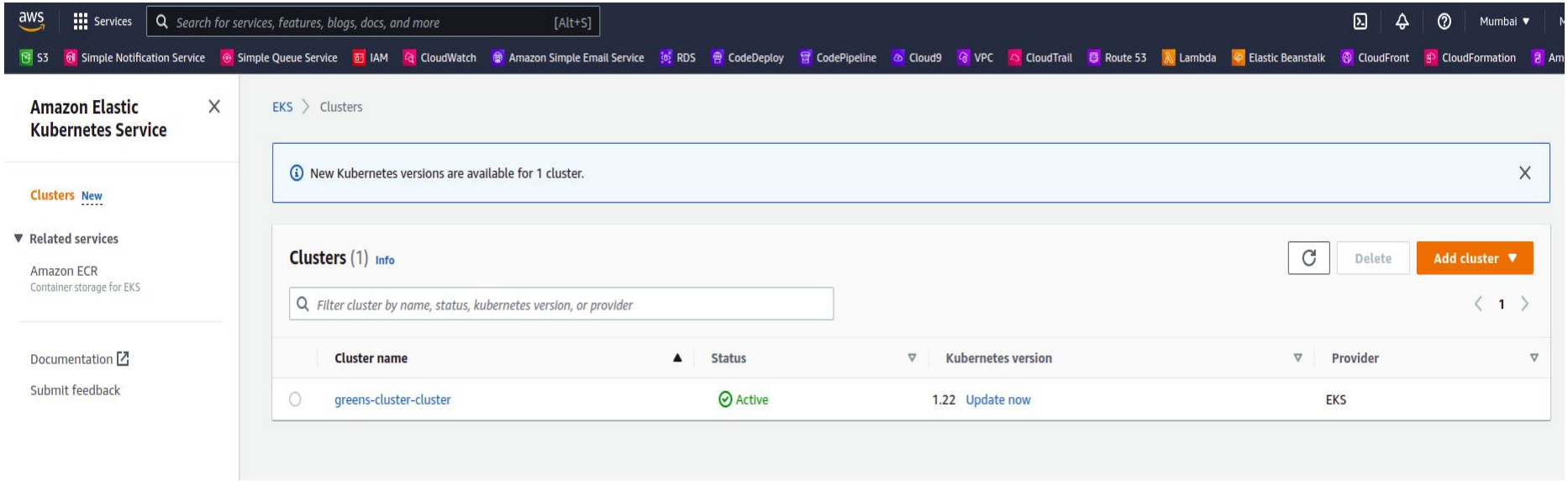
```
prakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/terraform-eks-greens$ terraform apply
aws_cloudwatch_log_group.greens-cluster: Refreshing state... [id=/aws/eks/greens-cluster/cluster]
aws_iam_role.workernodes: Refreshing state... [id=eks-node-group-example]
aws_iam_role.eks-iam-role: Refreshing state... [id=greens-cluster-eks-iam-role]
aws_iam_role_policy_attachment.AmazonEKS_CNI_Policy: Refreshing state... [id=eks-node-group-example-20220919052231577900000005]
aws_iam_role_policy_attachment.AmazonEC2ContainerRegistryReadOnly: Refreshing state... [id=eks-node-group-example-20220919052231460600000004]
aws_iam_role_policy_attachment.AmazonEKSWorkerNodePolicy: Refreshing state... [id=eks-node-group-example-20220919052230570100000001]
aws_iam_role_policy_attachment.EC2InstanceProfileForImageBuilderECRContainerBuilds: Refreshing state... [id=eks-node-group-example-20220919052230570100000002]
aws_iam_role_policy_attachment.AmazonEKSClusterPolicy: Refreshing state... [id=greens-cluster-eks-iam-role-20220919052231582600000006]
aws_iam_role_policy_attachment.AmazonEC2ContainerRegistryReadOnly-EKS: Refreshing state... [id=greens-cluster-eks-iam-role-20220919052231451900000003]

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

Terraform will perform the following actions:

  # aws_eks_cluster.greens-cluster-eks will be created
  + resource "aws_eks_cluster" "greens-cluster-eks" {
      + arn                   = (known after apply)
      + certificate_authority = (known after apply)
      + created_at            = (known after apply)
      + endpoint              = (known after apply)
      + id                    = (known after apply)
      + identity              = (known after apply)
      + name                  = "greens-cluster-cluster"
      + platform_version      = (known after apply)
      + role_arn              = "arn:aws:iam::170529709485:role/greens-cluster-eks-iam-role"
      + status                = (known after apply)
      + tags_all              = (known after apply)
      + version               = (known after apply)
```

After that terraform apply can you cluster in aws console



After that we need create ECR repo click create repository and put private

aws | Services | Search for services, features, blogs, docs, and more [Alt+S] | Mumbai | Martin

S3 | Simple Notification Service | Simple Queue Service | IAM | CloudWatch | Amazon Simple Email Service | RDS | CodeDeploy | CodePipeline | Cloud9 | VPC | CloudTra

Amazon Elastic Container Registry

- Private registry
- Public registry
- Repositories

- Getting started
- Documentation
- Public gallery

Amazon ECR > Repositories

Private | Public

Private repositories (1)

View push commands | Delete | Actions | Create repository

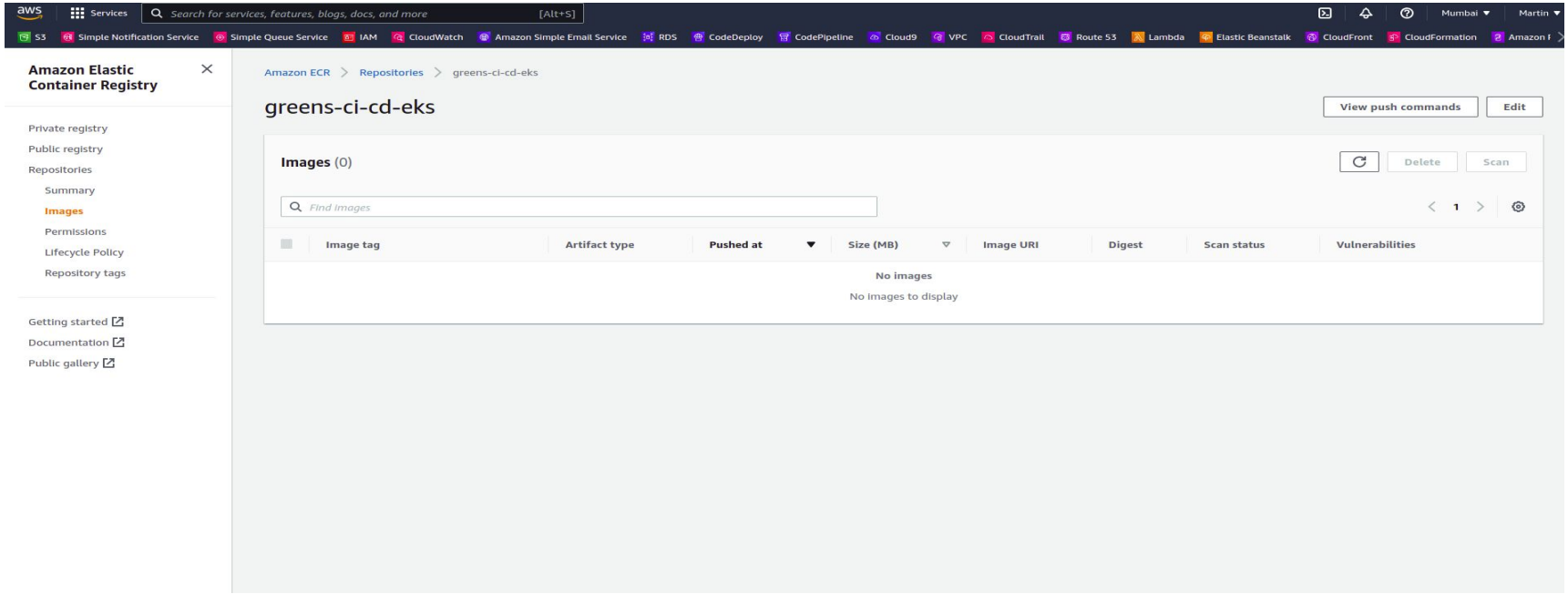
Find repositories

1

	Repository name	URI	Created at	Tag immutability	Scan frequency	Encryption type	Pull-thro cach
○	greens-test	170529709485.dkr.ecr.ap-south-1.amazonaws.com/greens-test	04 September 2022, 13:50:21 (UTC+05.5)	Disabled	Manual	AES-256	Inact

Feedback | Looking for language selection? Find it in the new Unified Settings | © 2022, Amazon Internet Services Private Ltd. or its affiliates. | Privacy | Terms | Cookie preferences

After creation repo u can see top right view push command login ecr and push images



Push commands for greens-ci-cd-eks

macOS / Linux | Windows

Make sure that you have the latest version of the AWS CLI and Docker installed. For more information, see Getting started with Amazon ECR.

Use the following steps to authenticate and push an image to your repository. For additional registry authentication methods, including the Amazon ECR credential helper, see Registry authentication.

1. Retrieve an authentication token and authenticate your Docker client to your registry.
   Use the AWS CLI:

```
aws ecr get-login-password --region ap-south-1 | docker login --username AWS --password-stdin
170529709485.dkr.ecr.ap-south-1.amazonaws.com
```

   Note: if you receive an error using the AWS CLI, make sure that you have the latest version of the AWS CLI and Docker installed.

2. Build your Docker image using the following command. For information on building a Docker file from scratch, see the instructions here. You can skip this step if your image has already been built:

```
docker build -t greens-ci-cd-eks .
```

3. After the build is completed, tag your image so you can push the image to this repository:

```
docker tag greens-ci-cd-eks:latest 170529709485.dkr.ecr.ap-south-1.amazonaws.com/greens-ci-cd-eks:latest
```

4. Run the following command to push this image to your newly created AWS repository:

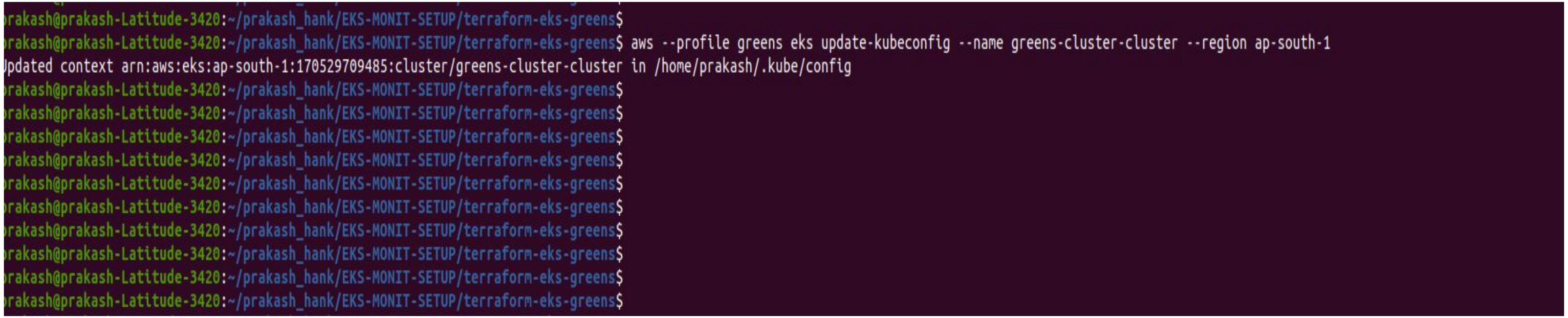
```
docker push 170529709485.dkr.ecr.ap-south-1.amazonaws.com/greens-ci-cd-eks:latest
```

Before ecr get login u need add profile name like aws –profile greens ecr get-login **********

Close

Connect eks cluster via kubeconfig update

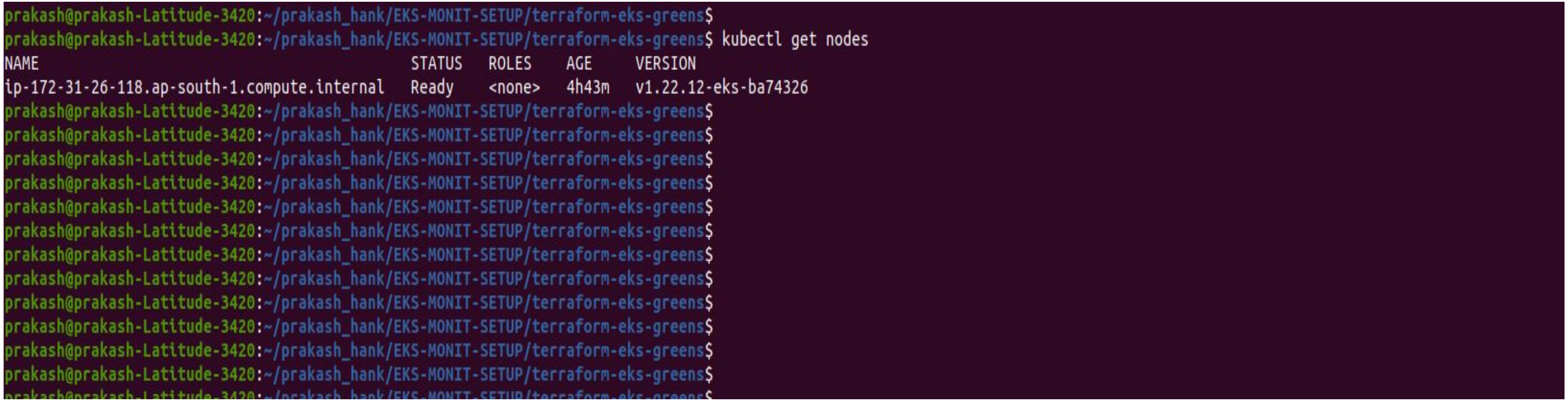
aws –profile greens eks update-kubeconfig –name clustername –region rigion_name



```
rakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/terraform-eks-greens$
rakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/terraform-eks-greens$ aws --profile greens eks update-kubeconfig --name greens-cluster-cluster --region ap-south-1
pdated context arn:aws:eks:ap-south-1:170529709485:cluster/greens-cluster-cluster in /home/prakash/.kube/config
rakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/terraform-eks-greens$
rakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/terraform-eks-greens$
rakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/terraform-eks-greens$
rakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/terraform-eks-greens$
rakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/terraform-eks-greens$
rakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/terraform-eks-greens$
rakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/terraform-eks-greens$
rakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/terraform-eks-greens$
rakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/terraform-eks-greens$
rakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/terraform-eks-greens$
```

Test if your cluster connected or not some basic commands

Kubectl get nodes
Kubectl get pods -n kube-system

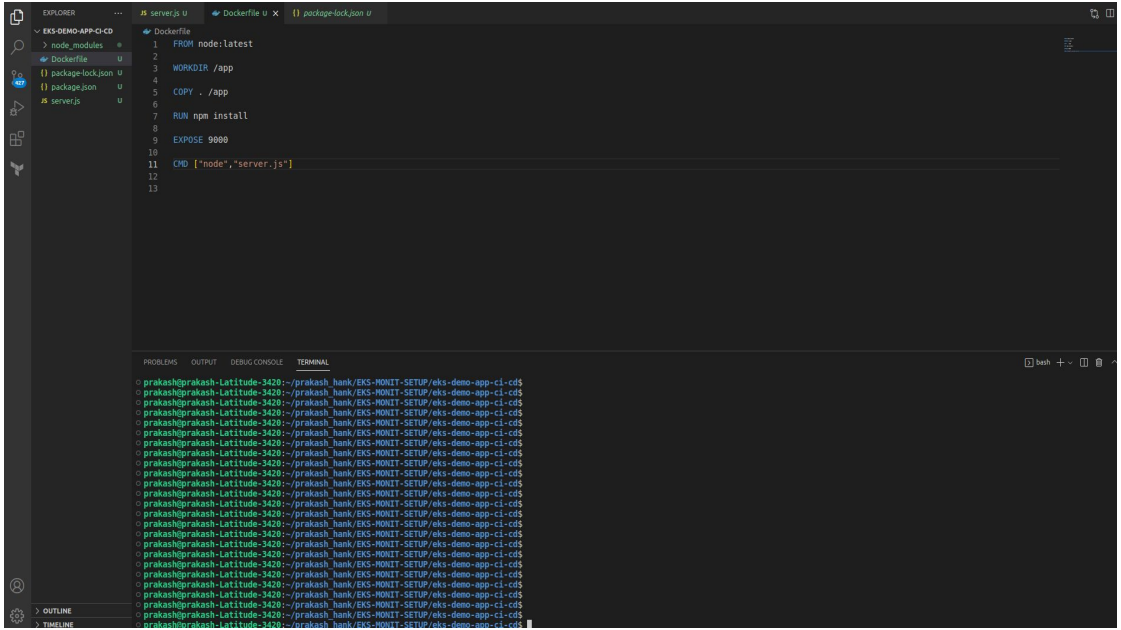


```
prakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/terraform-eks-greens$
prakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/terraform-eks-greens$ kubectl get nodes
NAME                                         STATUS   ROLES    AGE     VERSION
ip-172-31-26-118.ap-south-1.compute.internal   Ready    <none>   4h43m   v1.22.12-eks-ba74326
prakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/terraform-eks-greens$
prakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/terraform-eks-greens$
prakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/terraform-eks-greens$
prakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/terraform-eks-greens$
prakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/terraform-eks-greens$
prakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/terraform-eks-greens$
prakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/terraform-eks-greens$
prakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/terraform-eks-greens$
prakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/terraform-eks-greens$
prakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/terraform-eks-greens$
prakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/terraform-eks-greens$
prakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/terraform-eks-greens$
```

Lets we move to deployment side

Create Dockerfile
and server.js

All file are inside the github
Please refer



Login AWS ECR account

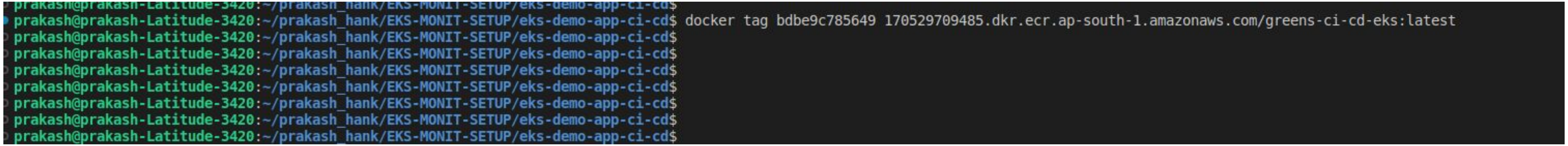
```
prakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/eks-demo-app-ci-cd$ aws --profile greens ecr get-login-password --region ap-south-1 | docker login --username AWS --password-stdin 170529709485.d
r.ecr.ap-south-1.amazonaws.com
WARNING! Your password will be stored unencrypted in /home/prakash/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
prakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/eks-demo-app-ci-cd$
```

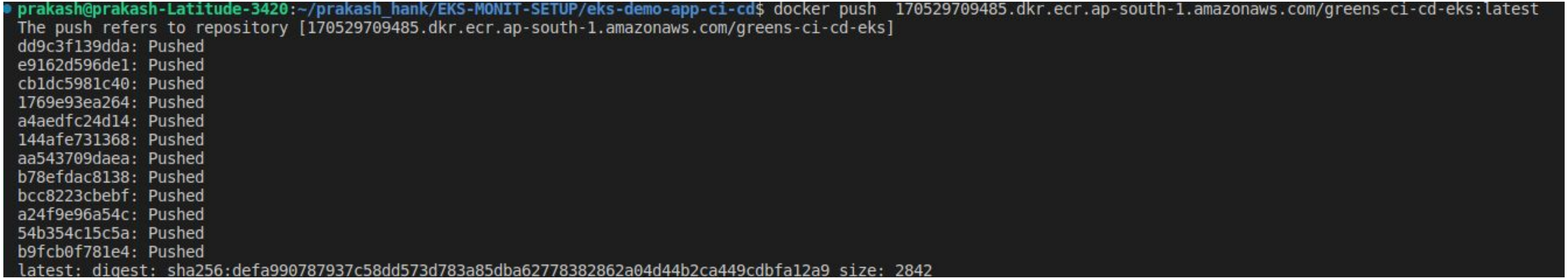
Docker build

```
prakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/eks-demo-app-ci-cd$ docker build .
Sending build context to Docker daemon  2.454MB
Step 1/6 : FROM node:latest
 ---> 2577ab2cda97
Step 2/6 : WORKDIR /app
 ---> Using cache
 ---> 48aba5907250
Step 3/6 : COPY . /app
 ---> Using cache
 ---> 7e5c126f6d3a
Step 4/6 : RUN npm install
 ---> Using cache
 ---> ac5568a96e7c
Step 5/6 : EXPOSE 9000
 ---> Using cache
 ---> 674dec84b860
Step 6/6 : CMD ["node","server.js"]
 ---> Using cache
 ---> bdbe9c785649
Successfully built bdbe9c785649
```

Docker tag to ecr repo and push please refer push commands in ECR right top



```
prakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/eks-demo-app-ci-cd$
prakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/eks-demo-app-ci-cd$ docker tag bdbe9c785649 170529709485.dkr.ecr.ap-south-1.amazonaws.com/greens-ci-cd-eks:latest
prakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/eks-demo-app-ci-cd$
prakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/eks-demo-app-ci-cd$
prakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/eks-demo-app-ci-cd$
prakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/eks-demo-app-ci-cd$
prakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/eks-demo-app-ci-cd$
prakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/eks-demo-app-ci-cd$
prakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/eks-demo-app-ci-cd$
```



```
prakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/eks-demo-app-ci-cd$ docker push  170529709485.dkr.ecr.ap-south-1.amazonaws.com/greens-ci-cd-eks:latest
The push refers to repository [170529709485.dkr.ecr.ap-south-1.amazonaws.com/greens-ci-cd-eks]
dd9c3f139dda: Pushed
e9162d596de1: Pushed
cb1dc5981c40: Pushed
1769e93ea264: Pushed
a4aedfc24d14: Pushed
144afe731368: Pushed
aa543709daea: Pushed
b78efdac8138: Pushed
bcc8223cbebf: Pushed
a24f9e96a54c: Pushed
54b354c15c5a: Pushed
b9fcb0f781e4: Pushed
latest: digest: sha256:defa990787937c58dd573d783a85dba62778382862a04d44b2ca449cdbfa12a9 size: 2842
```

After image push we need create deployment with k8s.yaml

```
prakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/eks-demo-app-ci-cd$ kubectl apply -f k8s.yaml
deployment.apps/green-eks created
```

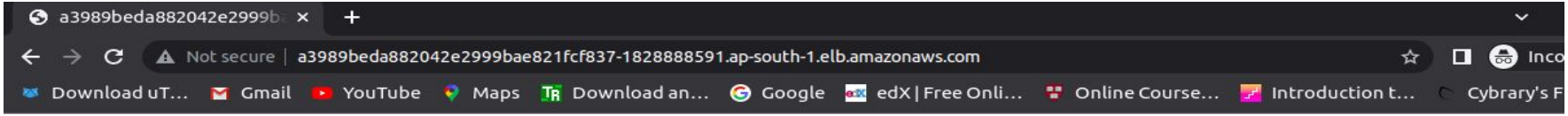
After deployment please verify deployment running or not

```
prakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/eks-demo-app-ci-cd$ kubectl get pods
NAME                          READY   STATUS    RESTARTS   AGE
green-eks-758bd6b6c7-2957g    1/1     Running   0          7m33s
prakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/eks-demo-app-ci-cd$
```

After deployment we need create service

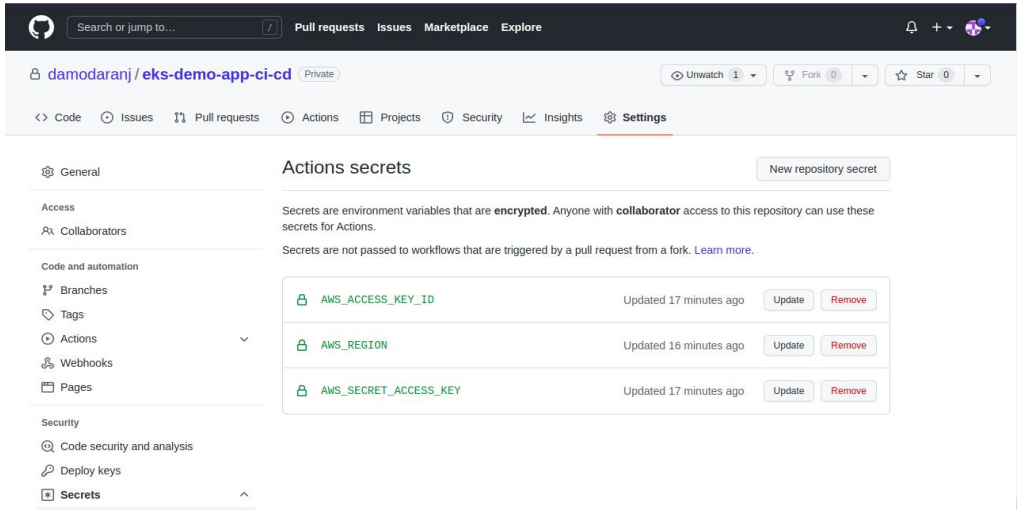
```
prakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/eks-demo-app-ci-cd$ kubectl apply -f k8s-service.yaml
service/green-eks unchanged
prakash@prakash-Latitude-3420:~/prakash_hank/EKS-MONIT-SETUP/eks-demo-app-ci-cd$ kubectl get svc
NAME         TYPE           CLUSTER-IP     EXTERNAL-IP                                                              PORT(S)                   AGE
green-eks    LoadBalancer   10.100.147.64  a3989beda882042e2999bae821fcf837-1828888591.ap-south-1.elb.amazonaws.com  80:31331/TCP,443:30366/TCP  8m12s
kubernetes   ClusterIP      10.100.0.1     <none>                                                                   443/TCP                   5h31m
```

After service u get load balancer external-ip


a3989beda882042e2999b
Not secure
a3989beda882042e2999bae821fcf837-1828888591.ap-south-1.elb.amazonaws.com
Inco
Download uT...
Gmail
YouTube
Maps
Download an...
Google
edX | Free Onli...
Online Course...
Introduction t...
Cybrary's F

Hello World!

Deployment done lets we move to CI and CD with git action
Already below i attached source code and terraform code
Inside the source u can see gitaction workflow yaml file
After we need configure repository secret in git actions



Search or jump to... / Pull requests Issues Marketplace Explore

damodaranj / eks-demo-app-ci-cd Private

Unwatch 1 Fork 0 Star 0

Code Issues Pull requests Actions Projects Security Insights Settings

master 1 branch 0 tags Go to file Add file Code

damodaranj server.js-changed c22bad5 1 hour ago 7 commits

.github/workflows	final-changes	1 hour ago
Dockerfile	initial-Rollout	5 hours ago
k8s-service.yaml	initial-Rollout	5 hours ago
k8s.yaml	Update k8s.yaml	1 hour ago
package-lock.json	initial-Rollout	5 hours ago
package.json	initial-Rollout	5 hours ago
server.js	server.js-changed	1 hour ago

Add a README with an overview of your project. Add a README

About

No description, website, or topics provided.

0 stars
1 watching
0 forks

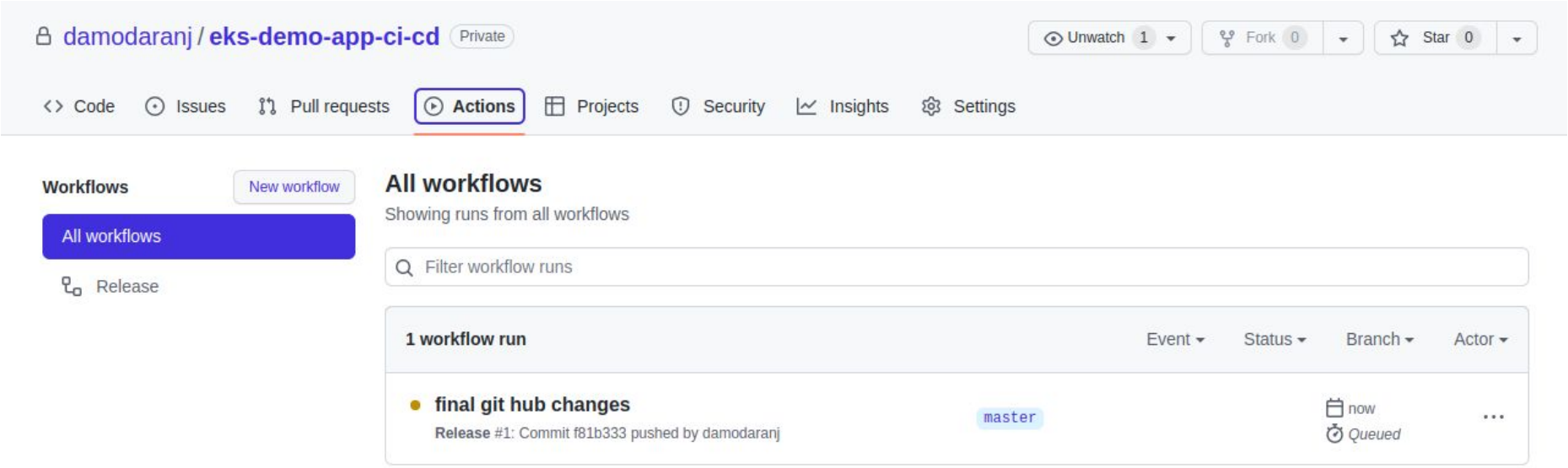
Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Click action create new workflow copy my green-ci-cd.yaml and paste and commit it will be start automatically



damodaranj / **eks-demo-app-ci-cd** Private

Unwatch 1 | Fork 0 | Star 0

Code | Issues | Pull requests | **Actions** | Projects | Security | Insights | Settings

final-changes Release #2

Cancel workflow

Summary

Jobs

Release

Release

Started 24s ago

Search logs

Set up job	1s
Cancel Previous Runs	0s
Checkout	1s
Configure AWS credentials	1s
Login to Amazon ECR	3s
Set up Docker Buildx	3s
Build & Push Image	12s

```
1  ▶ Run docker buildx create --use
32
32 silly_moser
33 #1 [internal] booting buildkit
34 #1 pulling image moby/buildkit:buildx-stable-1 0.1s done
```

final-changes Release #2

Re-run all jobs

Summary

Jobs

Release

Release

succeeded 14 seconds ago in 44s

Search logs

Set up job	1s
Cancel Previous Runs	0s
Checkout	1s
Configure AWS credentials	1s
Login to Amazon ECR	3s
Set up Docker Buildx	3s
Build & Push Image	31s
Post Set up Docker Buildx	0s
Post Login to Amazon ECR	0s
Post Configure AWS credentials	0s
Post Checkout	0s
Complete job	0s

Thanks