**Gitlab server, GitLab runner setup and maven CI pipeline to push the docker image to ECR**

Below are the high-level steps:

1. Install GitLab server
2. Install GitLab-runner
3. Register GitLab-runner using token
4. Create ECR repository
5. Import the project and run pipeline

**1.Install GitLab server:**

I am using the Amazon Linux Instance, but depending on your OS flavor you can choose the repo from below website:

<https://packages.gitlab.com/gitlab/gitlab-ce>

**Install the required dependencies :**

yum install curl policycoreutils-python openssh-server -y

**Download git repo and install it :**

curl https://packages.gitlab.com/install/repositories/gitlab/gitlab-ce/script.rpm.sh | sudo bash

EXTERNAL\_URL="`curl http://169.254.169.254/latest/meta-data/public-ipv4`" yum install -y gitlab-ce

**Note :**

Once you installed the GitLab package, you can execute the required configuration utility. This file provides automatic configurations, and you can modify it according to your need. Run the following edit of the GitLab configuration file.

$ sudo vim /etc/gitlab/gitlab.rb

Now, edit the configuration file to change hostname using external\_url variable so that, you can access them from other remote machine using the specified hostname and other parameters:

Run the following command to reconfigure the services of GitLab:

*$*sudo gitlab-ctl reconfigure

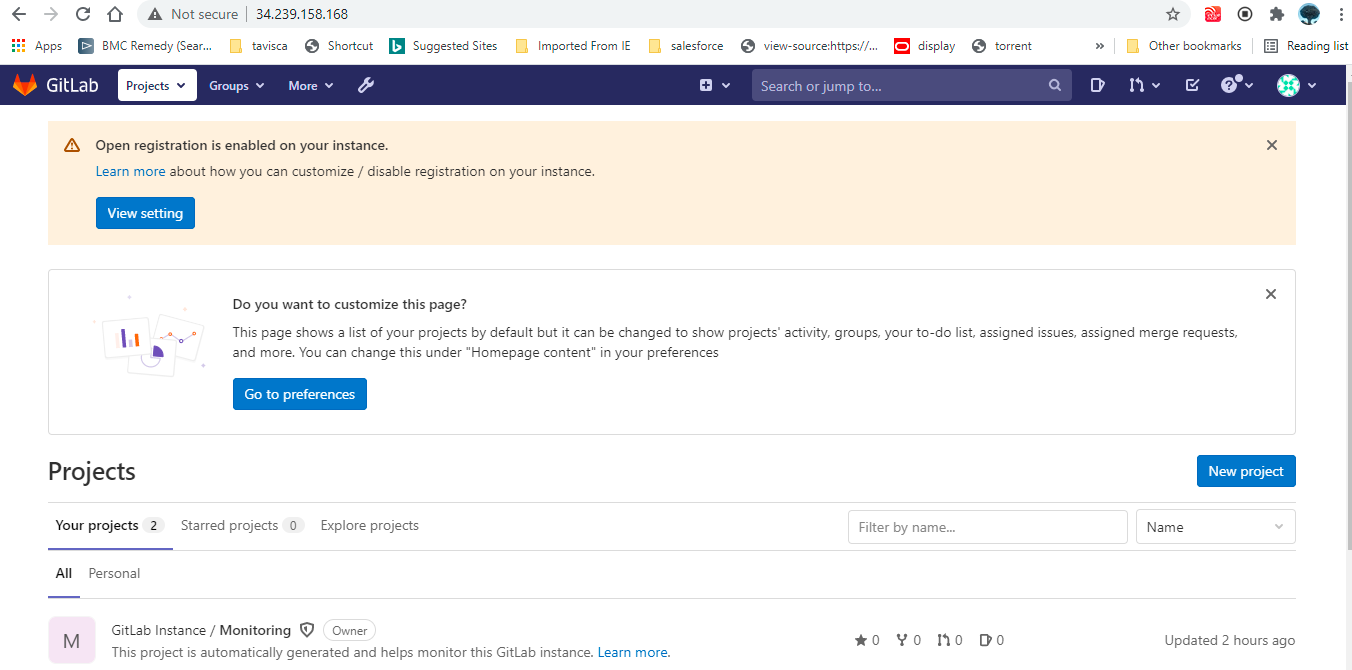
Open the port number 80 as gitlab is running on port 80.

Check the GitLab UI from browser:

<http://public_ip_of_ec2_instance> or <http://34.239.158.168>

To access the GitLab web portal that will ask you to set the username and password of root. Enter the new root password. After verifying then, click the ‘Change your password’ option.

Now, login with the username as root and then provide the password. You will see the following gitlab dashboard screen on your system.



**2.Install GitLab-runner**

https://docs.gitlab.com/runner/install/linux-repository.html#installing-the-runner

* *Add GitLab official repository:*

curl -L "https://packages.gitlab.com/install/repositories/runner/gitlab-runner/script.rpm.sh" | sudo bash

* *Install the latest version of GitLab Runner, or skip to the next step to install a specific version:*

export GITLAB\_RUNNER\_DISABLE\_SKEL=true; sudo -E yum install gitlab-runner

* To install a specific version of GitLab Runner:

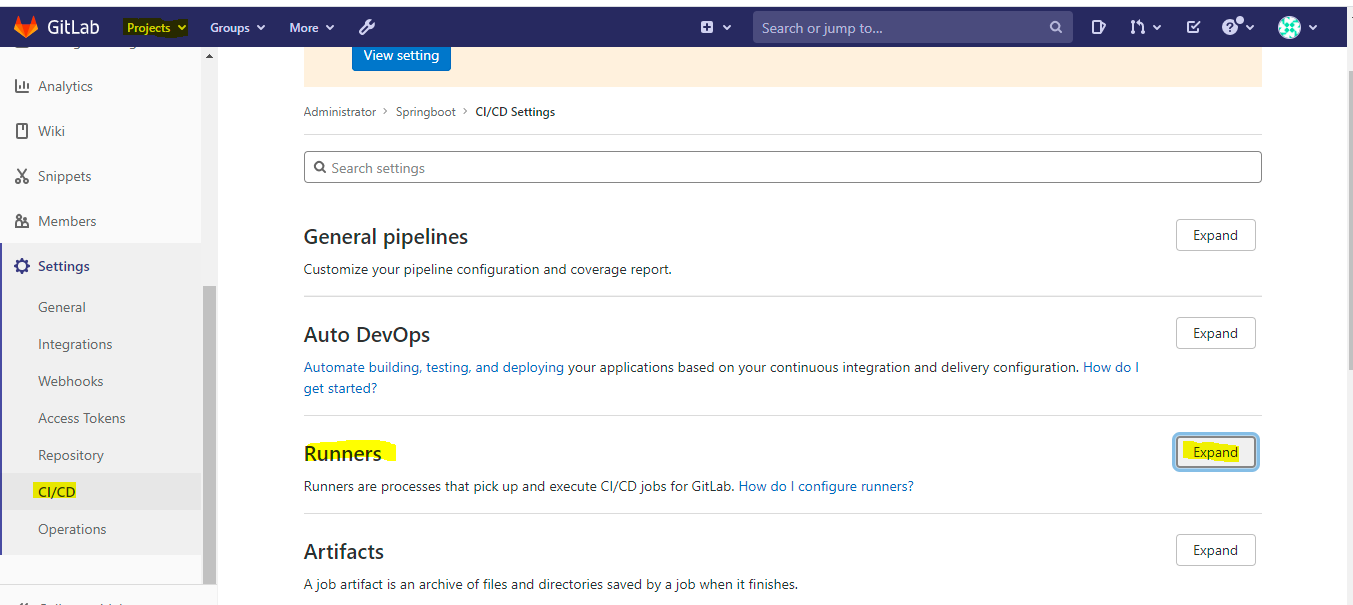
yum list gitlab-runner --showduplicates | sort -r

export GITLAB\_RUNNER\_DISABLE\_SKEL=true;sudo -E yum install gitlab-runner-10.0.0-1

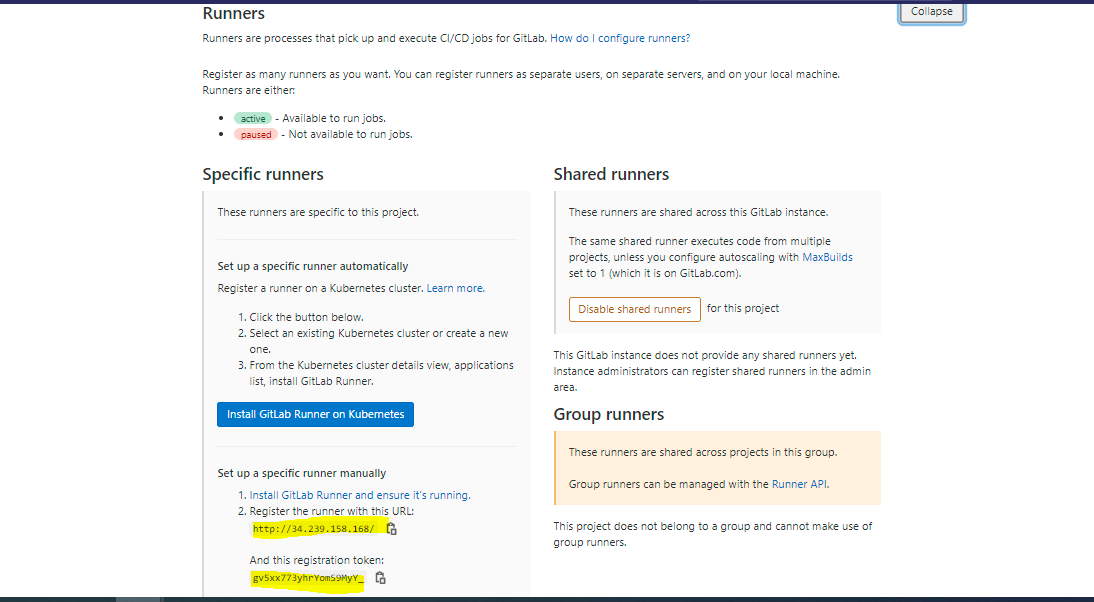
**3. Register runner**

For registering the runner, we need to get token from GitLab server first.

To get the token go to projects -> select project -> settings -> CI/CD 🡪Runners



**Copy token and url**



There are multiple types of runner executer like shell, docker, k8s. For our use case we will use docker executor.

First install Docker on the server where we are running GitLab Runner.

# yum install docker -y

# service docker start

## # chmod 666 /var/run/docker.sock

## sudo gitlab-runner register -n \

## --url http://34.239.158.168/ \

## --registration-token gv5xx773yhrYomS9MyY\_ \

## --executor docker \

## --description "My Docker Runner" \

## --docker-image "docker:stable" \

## --docker-privileged

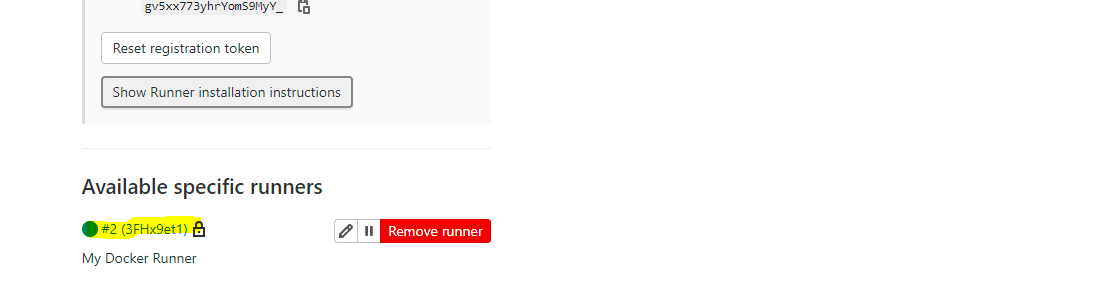
You can get the list of registered runners:

# gitlab-runner list

Runtime platform arch=amd64 os=linux pid=10131 revision=7f7a4bb0 version=13.11.0

Listing configured runners ConfigFile=/etc/gitlab-runner/config.toml

My Docker Runner Executor=docker Token=3FHx9et1jSzGpoCThz4Y URL=http://34.239.158.168/



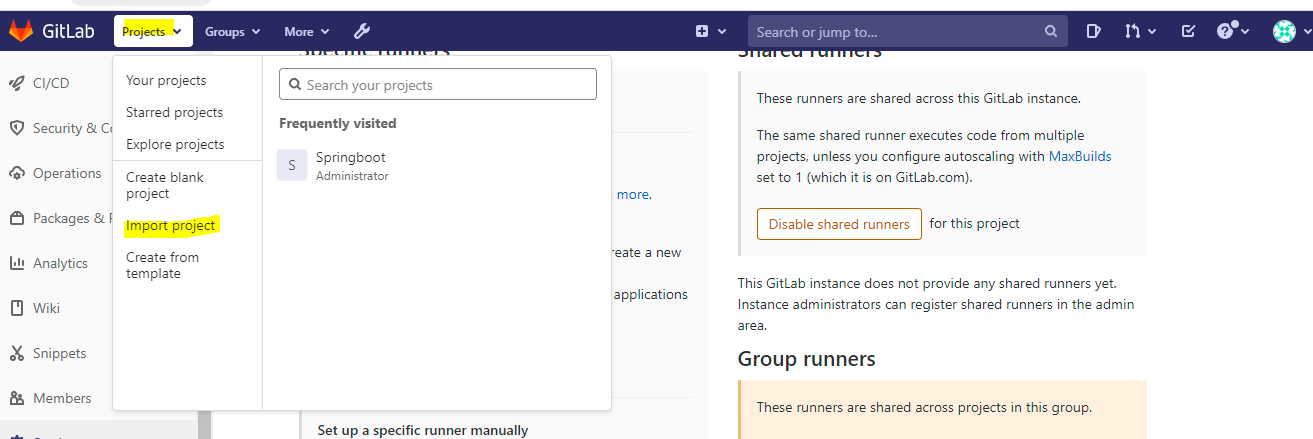
**4.Create ECR repository**

Goto ECR repository page and create repository with the name : springdemo

**5. Import the project and commit it**

Goto Projects in GitLab and do import project and specify the below git url to import repo.

<https://github.com/tushardashpute/springboot.git>



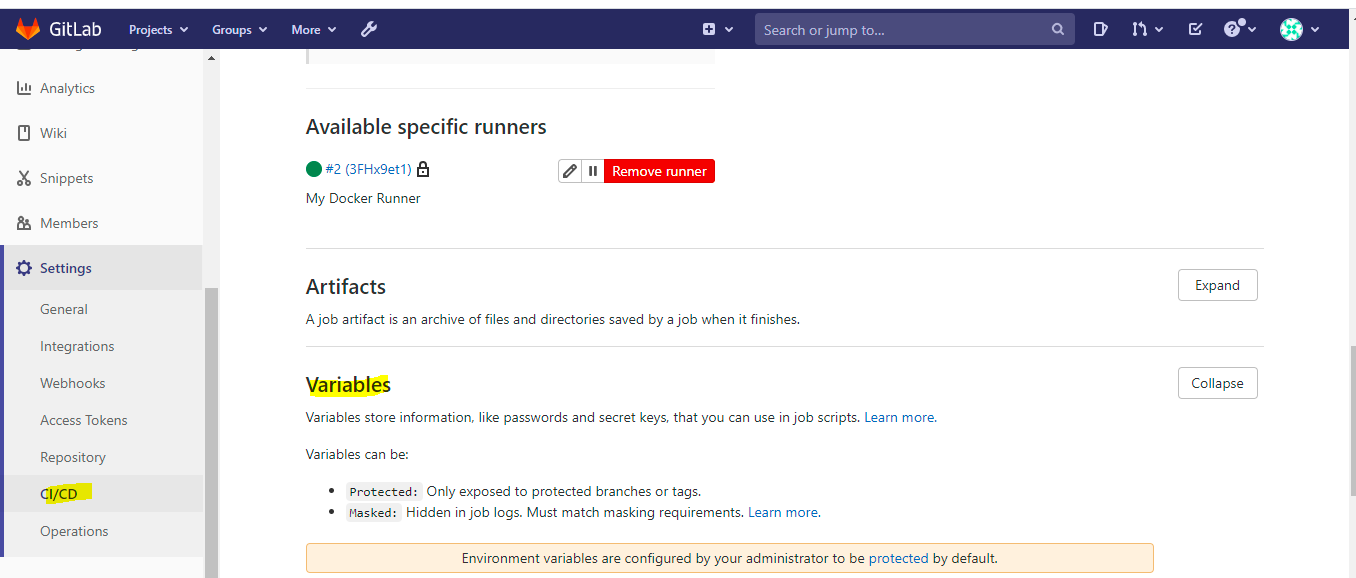
Now goto CI/CD 🡪 Variables 🡪 and add the following AWS variables to access the password of ECR and login to it.

**Variable Allowed Values**

AWS\_ACCESS\_KEY\_ID Any(access\_key\_id)

AWS\_DEFAULT\_REGION Any(us-east-1)

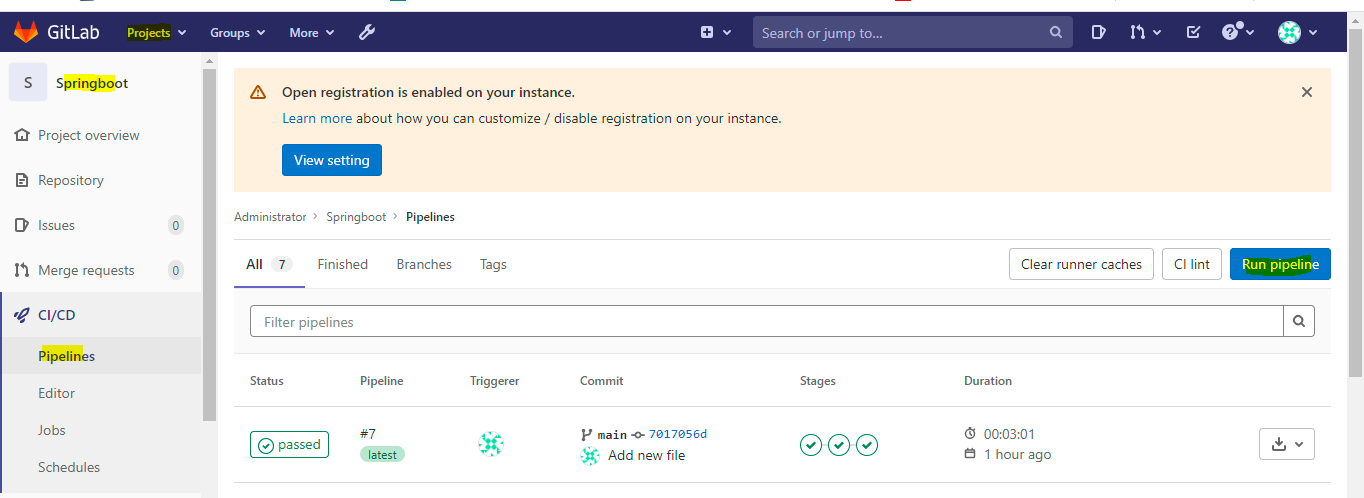
AWS\_SECRET\_ACCESS\_KEY Any(access\_key)



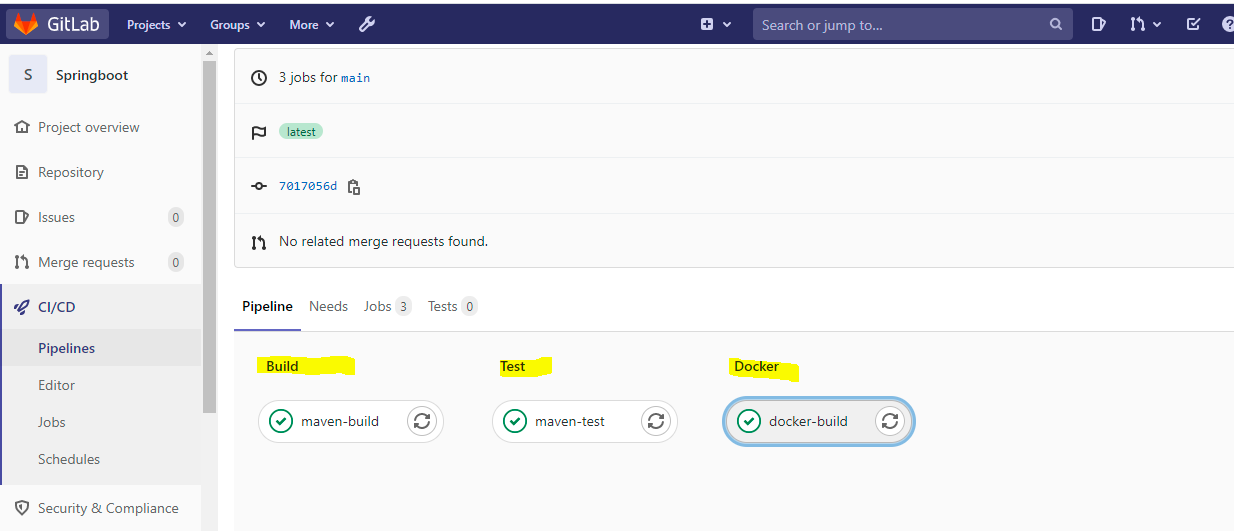
Once repo is imported, edit the file [.gitlab-ci.yml](https://github.com/tushardashpute/springboot/blob/main/.gitlab-ci.yml) update the DOCKER\_REGISTRY: with your ECR repo url and commit it. Once you commit the repo it will auto trigger the pipeline.

Or

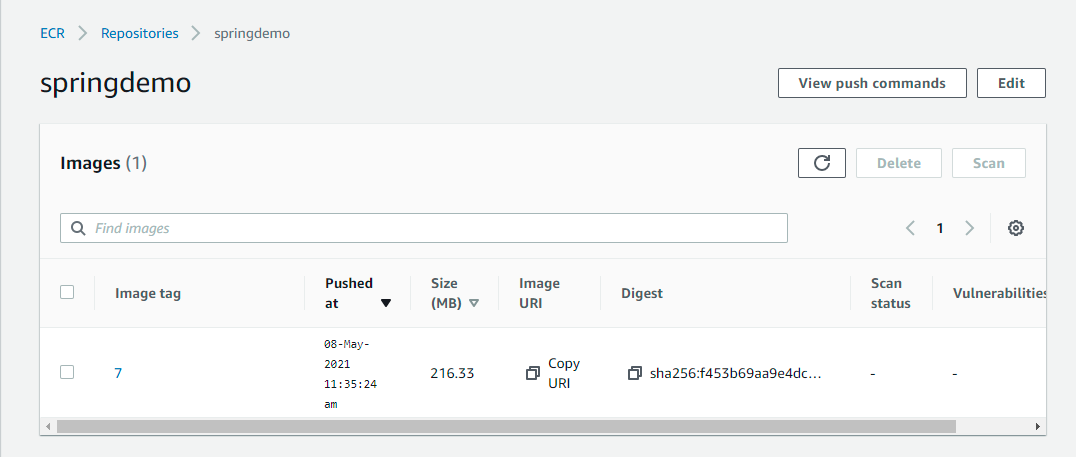
If you want you can goto projects 🡪 CI/CD 🡪 pipeline 🡪 run project.



You can see the stages and status as below :



Once pipeline is finished you can verify if image is pushed to your ECR repo or not.



To verify if image is created correctly or not, we can pull the image run the docker container using below command:

## # aws configure

AWS Access Key ID [None]: AKIAT6DSXXXXXXXXXXX

AWS Secret Access Key [None]: 5raxCB4Ymy8MXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Default region name [None]: us-east-1

Default output format [None]: json

[root@ip-172-31-67-134 opt]# aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin 270823544663.dkr.ecr.us-east-1.amazonaws.com

**Login Succeeded**

**# docker pull 270823544663.dkr.ecr.us-east-1.amazonaws.com/springdemo:7**

7: Pulling from springdemo

050382585609: Pull complete

a8c71082b2bb: Pull complete

abfe0cbfb3f2: Pull complete

Digest: sha256:f453b69aa9e4dcff39ccc1997841f5de02b7510bf458968cb89b6f457cb2da5b

Status: Downloaded newer image for 270823544663.dkr.ecr.us-east-1.amazonaws.com/springdemo:7

270823544663.dkr.ecr.us-east-1.amazonaws.com/springdemo:7

**# docker ps**

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

**[root@ip-172-31-67-134 opt]# docker images**

REPOSITORY TAG IMAGE ID CREATED SIZE

***270823544663.dkr.ecr.us-east-1.amazonaws.com/springdemo 7***  8689dfdd73ac 4 minutes ago 354MB

gitlab/gitlab-runner-helper x86\_64-7f7a4bb0 7ce33577de1e 19 minutes ago 70.2MB

docker 19-dind c0272ea5b8a2 7 days ago 236MB

docker dind dc8c389414c8 7 days ago 263MB

maven 3-jdk-8 87963037f00b 2 weeks ago 525MB

docker latest d2979b152a7d 3 weeks ago 246MB

**# docker run -it -p 8081:33333 -d 270823544663.dkr.ecr.us-east-1.amazonaws.com/springdemo:7**

**2986bf922257a1ccada7e20fba25054374b11f6ac13db837edada9fbd89dde8f**

**# docker ps**

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

33aec25f8298 270823544663.dkr.ecr.us-east-1.amazonaws.com/springdemo:7 "java -jar /springbo…" 7 minutes ago Up 7 minutes **0.0.0.0:8081->33333**/tcp gifted\_golick

Open port 8081 and access the <http://ip-address:8081/listallcustomers> url to check if image working properly or not.

