Demo 1: ECR

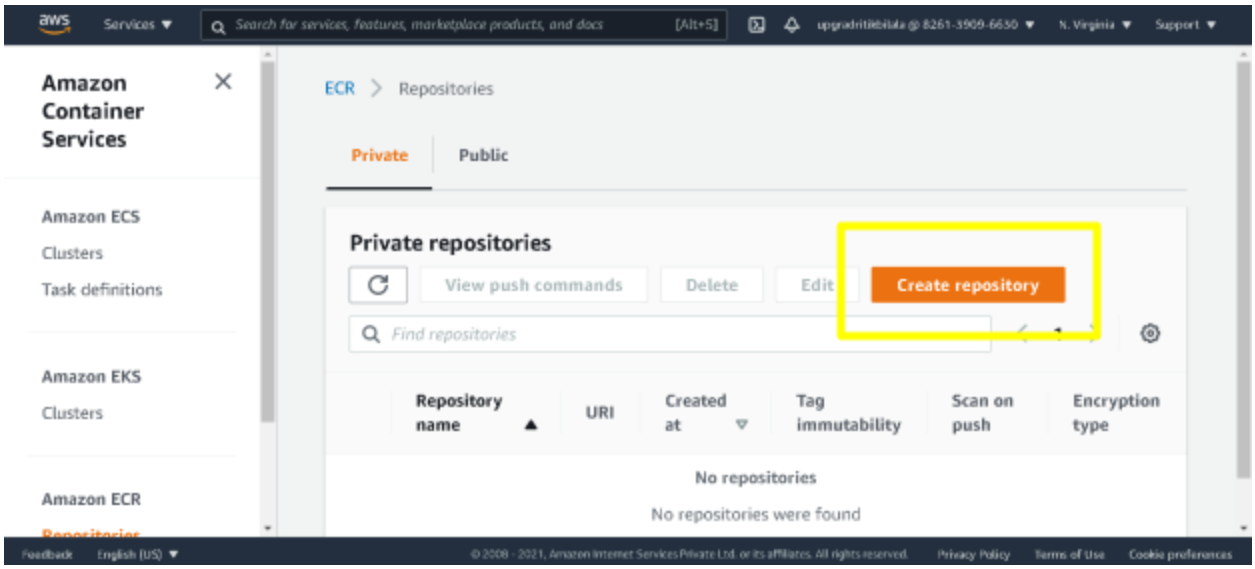
Introduction

Following are the learning objectives of this demonstration:

- Create an ECR repository
- Build an image and push it into the repository from an EC2 instance
- Set up the life cycle policy of ECR

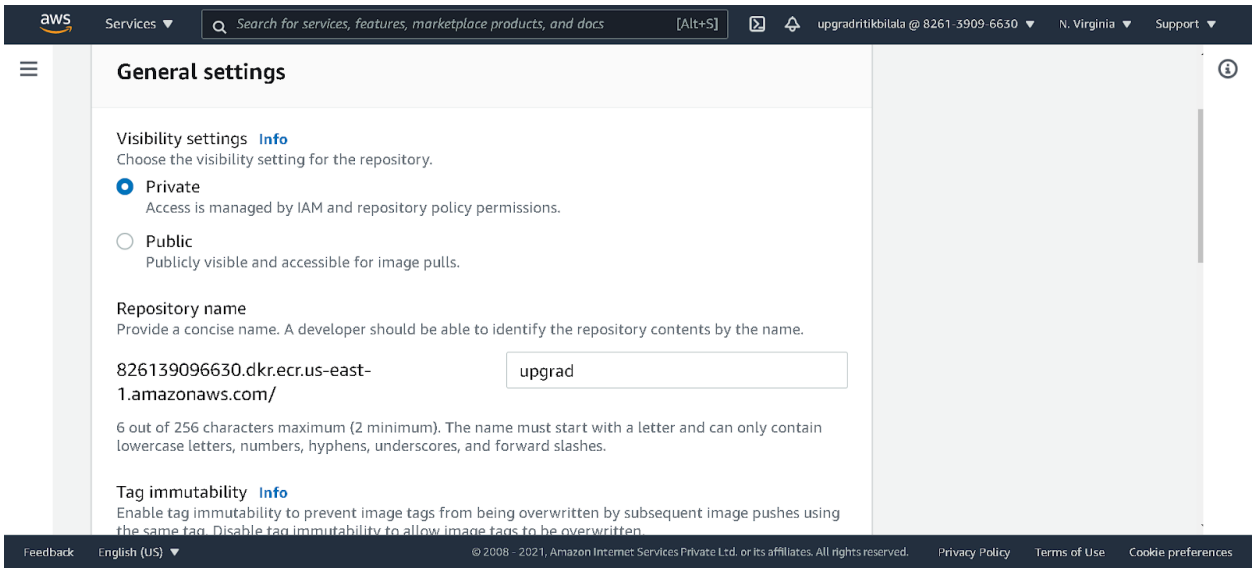
Create an IAM role with permission "AmazonEC2ContainerRegistryFullAccess" and attach it to the EC2 instance from which you will push images into the repository. If you are using a local machine outside AWS, you will need to configure an AWS account using AWS access keys.

1. Log in to the AWS console and visit the ECR page.

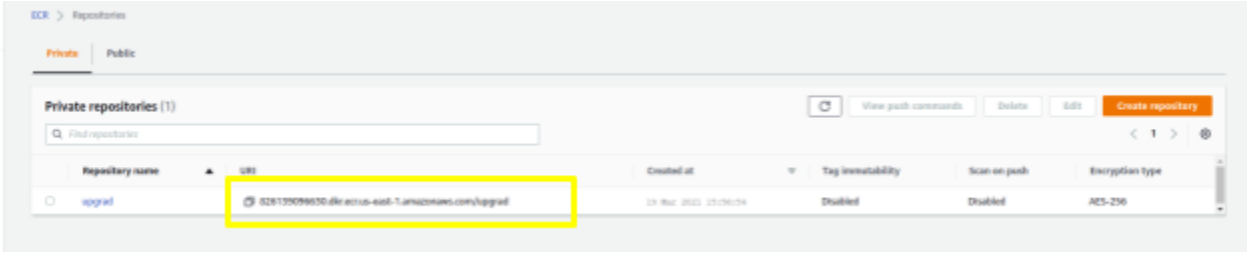


2. Click on create a repository.
3. Provide a repository name and click on the create button.

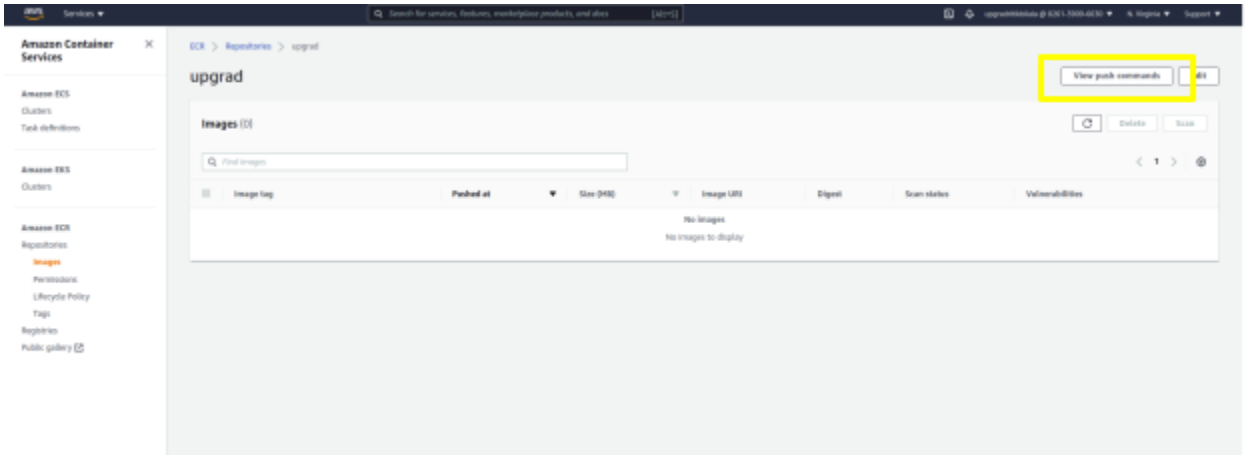
© Copyright 2020. upGrad Education Pvt. Ltd. All rights reserved



4. Copy your repository URL for the next steps.
   (My URL = **826139096630.dkr.ecr.us-east-1.amazonAWS.com)**

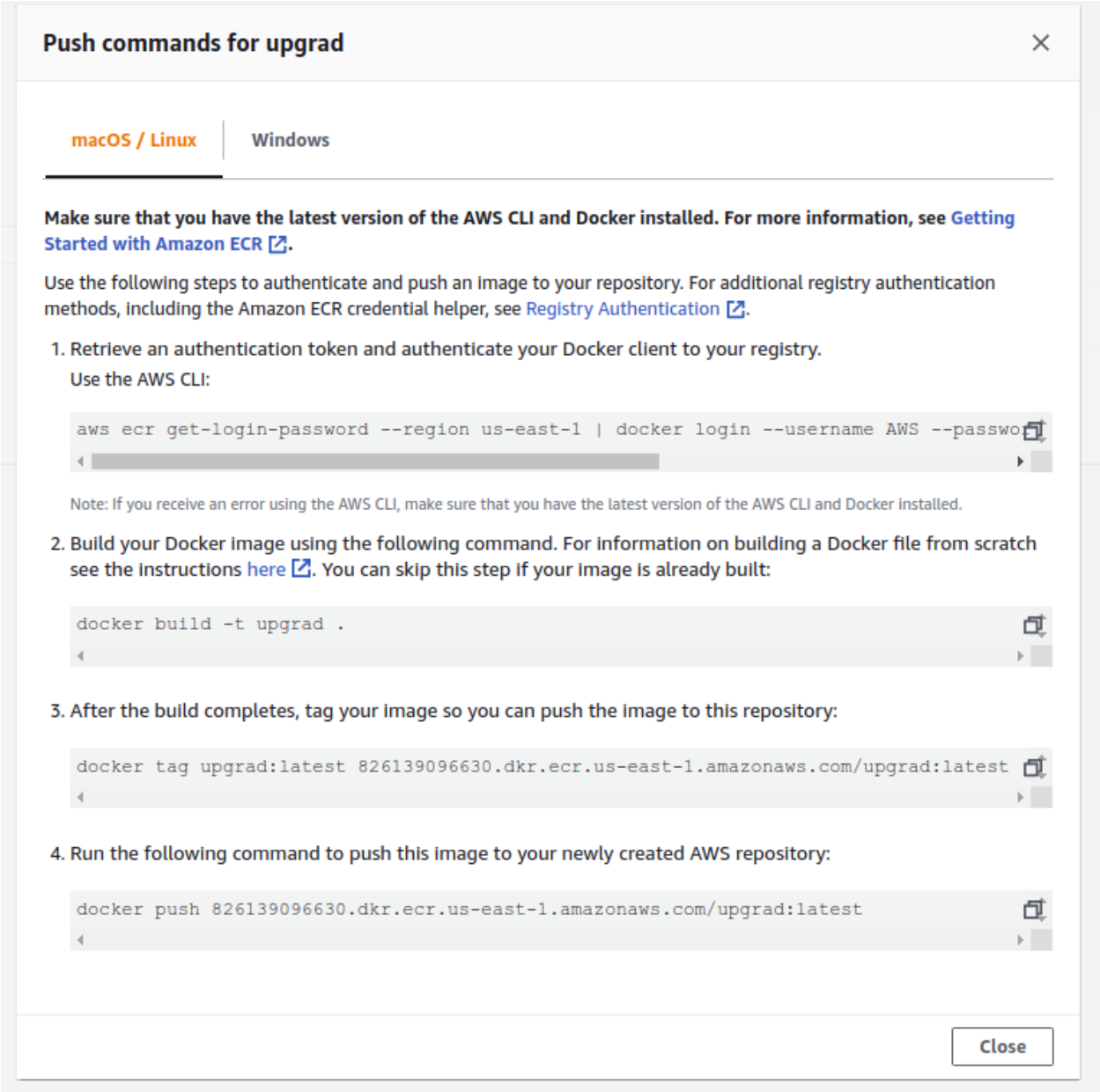


5. Now go to your repository details and click on the "**view push commands**" button.



6. After clicking the button, the following dialogue box will pop-up.

© Copyright 2020. upGrad Education Pvt. Ltd. All rights reserved



7. Now SSH into the EC2 instance and log in to the ECR repository using the above image's first command. Remember, your EC2 must have **AWScli** and **docker** installed.
AWS ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin 826139096630.dkr.ecr.us-east-1.amazonAWS.com

© Copyright 2020. upGrad Education Pvt. Ltd. All rights reserved

```
Last login: Fri Mar 19 13:24:26 2021 from 103.157.221.91
ubuntu@ip-172-31-30-37:~$ aws ecr get-login-password --region us-east-1 | docker login --user
WARNING! Your password will be stored unencrypted in /home/ubuntu/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
ubuntu@ip-172-31-30-37:~$
```

8. Now either create your dockerfile or pull your git repository, which contains the dockerfile.
9. Now build an image using **docker build -t <imagename:tag> .**

```
ubuntu@ip-172-31-30-37:~/NGINX$ docker build -t upgrad .
Sending build context to Docker daemon  65.54kB
Step 1/7 : FROM ubuntu
latest: Pulling from library/ubuntu
5d3b2c2d21bb: Pull complete
3fc2062ea667: Pull complete
75adf526d75b: Pull complete
Digest: sha256:b4f9e18267eb98998f6130342baacaeb9553f136142d40959a1b46d6401f0f2b
Status: Downloaded newer image for ubuntu:latest
 ---> 4dd97cefde62
Step 2/7 : RUN apt-get update && apt-get install -y nginx && rm -rf /var/lib/apt/li
 ---> Running in f337b975f5c1
Removing intermediate container 6bbd235053db
 ---> 87366dc14cae
Step 7/7 : EXPOSE 443
 ---> Running in 815cac12cbc8
Removing intermediate container 815cac12cbc8
 ---> e19967651a0a
Successfully built e19967651a0a
Successfully tagged upgrad:latest
ubuntu@ip-172-31-30-37:~/NGINX$
```

10. After the build completes, tag your image so you can push the image to this repository using the following commands:
    docker tag upgrad:latest \
    826139096630.dkr.ecr.us-east-1.amazonAWS.com/upgrad:latest

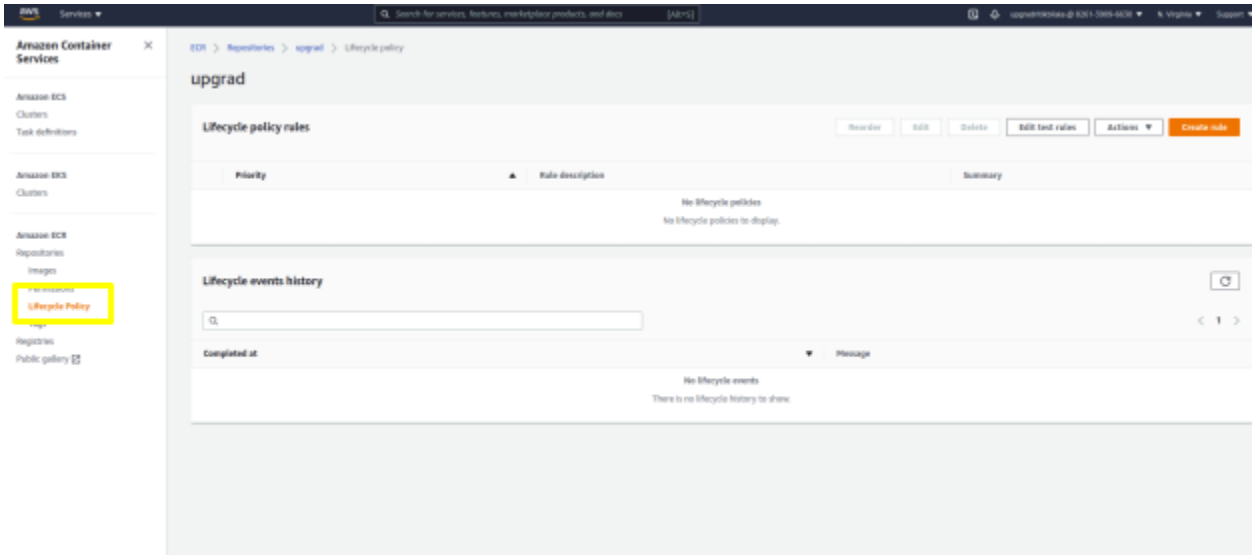
    Command *: (docker tag <repo name>:latest <repo url> :latest)*

11. Run the following command to push this image to your newly created AWS repository:
    docker push 826139096630.dkr.ecr.us-east-1.amazonAWS.com/upgrad:latest

© Copyright 2020. upGrad Education Pvt. Ltd. All rights reserved

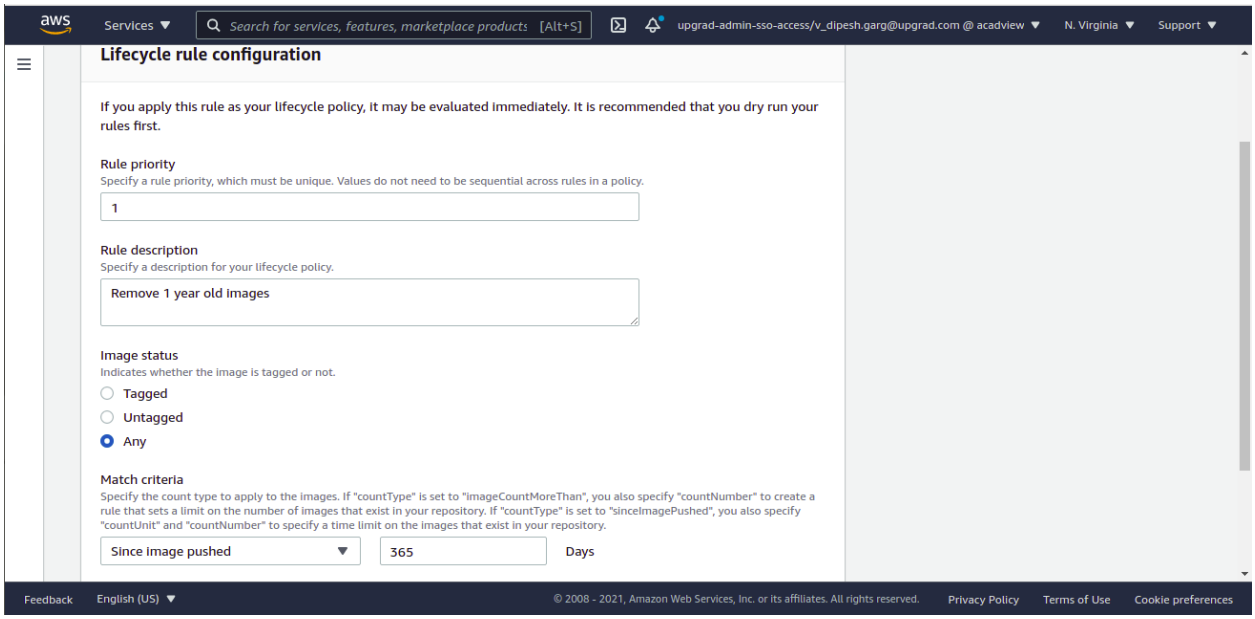
```
ubuntu@ip-172-31-30-37:~/NGINX$ docker tag upgrad:latest 826139096630.dkr.ecr.us-east-1.amazonaws.com/upgrad:latest
ubuntu@ip-172-31-30-37:~/NGINX$ docker push 826139096630.dkr.ecr.us-east-1.amazonaws.com/upgrad:latest
The push refers to repository [826139096630.dkr.ecr.us-east-1.amazonaws.com/upgrad]
03c18ce5e58c: Pushed
c20d459170d8: Pushed
db978cae6a05: Pushed
aeb3f02e9374: Pushed
latest: digest: sha256:17f2d26af3c0d50b16ae6ee1e59f6b81e15d358723f37518f957f71f1feeca71 size: 1155
ubuntu@ip-172-31-30-37:~/NGINX$
```

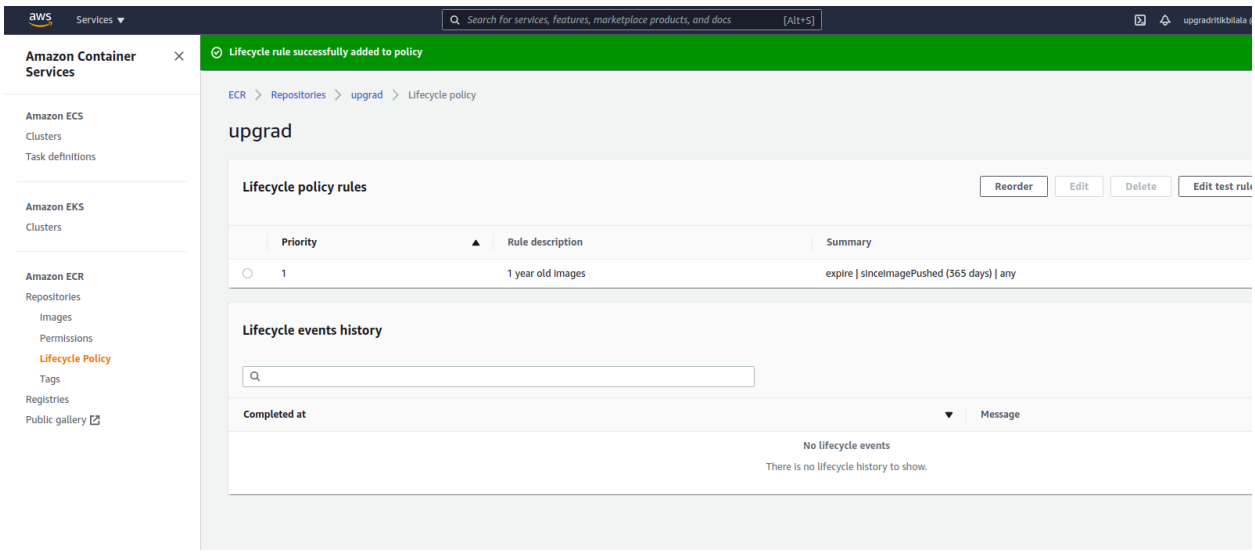
12. Go to the ECR console and check whether the image is pushed or not.
13. Now from the left panel, select “life cycle policy.”



14. Put an ECR lifecycle policy as given below so that images older than one year get deleted automatically.

© Copyright 2020. upGrad Education Pvt. Ltd. All rights reserved





Now images older than 365 days will get deleted from the repository.

© Copyright 2020. upGrad Education Pvt. Ltd. All rights reserved