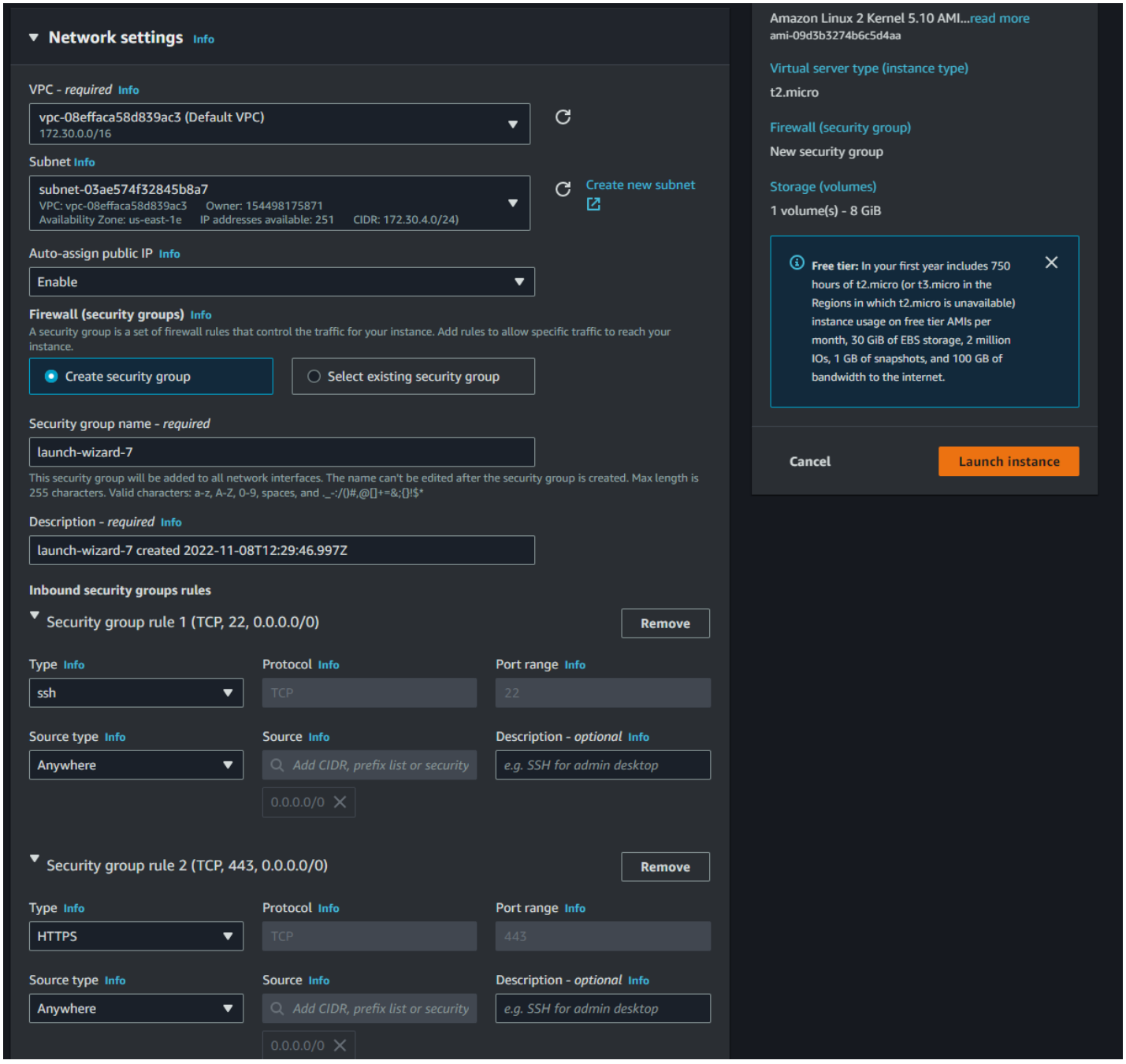
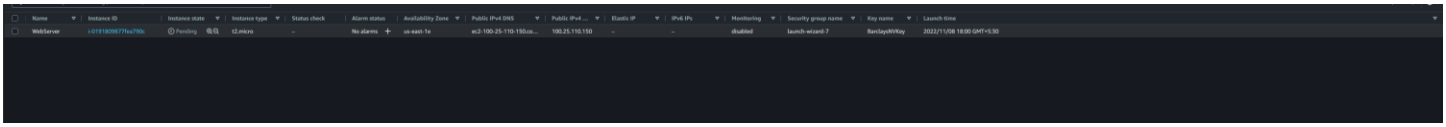
AWS Assignment

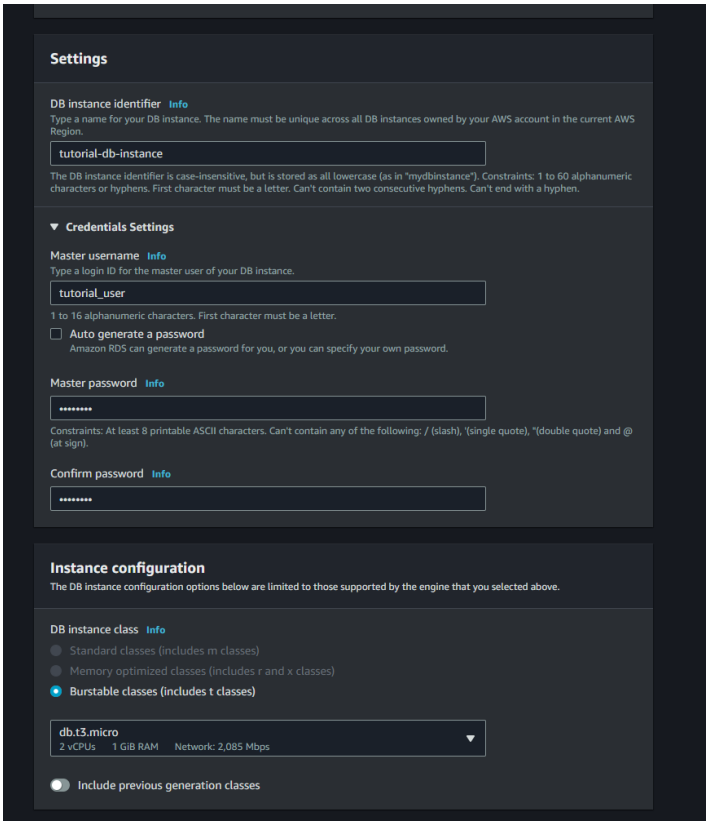
Create a web server and an Amazon RDS DB instance

1. Create an EC2 instance with HTTP inbound



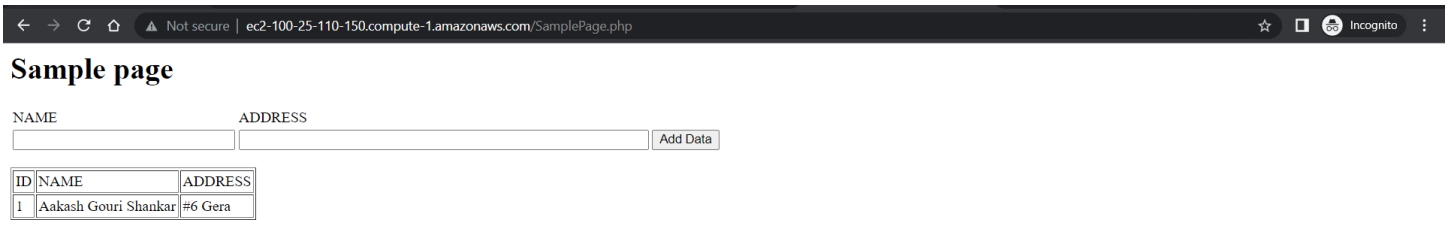


2. Create a DB instance and connect it to EC2 Instance



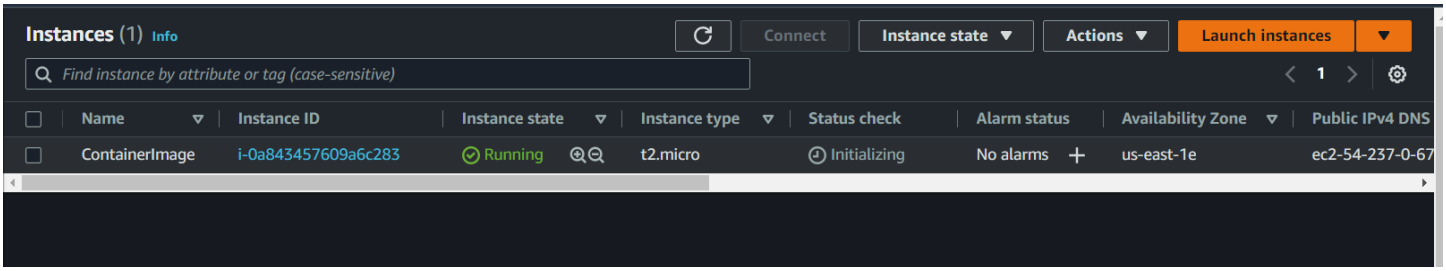
3. Install Web Server and Connect DB

```
sudo amazon-linux-extras install php8.0 mariadb10.5

sudo yum install -y httpd

sudo systemctl start httpd

sudo systemctl enable httpd
```



Creating a container image for use on Amazon ECS

1. Create an EC2 Instance and install Docker




```
1. sudo yum update -y

2. sudo amazon-linux-extras install docker
3. sudo service docker start
4. sudo systemctl enable docker
5. sudo usermod -a -G docker ec2-user
6. docker info
```

```
Last login: Wed Nov  9 07:11:05 2022 from ec2-18-206-107-28.compute-1.amazonaws.com

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-30-4-164 ~]$ docker info
Client:
 Context:    default
 Debug Mode: false

Server:
 Containers: 0
  Running: 0
  Paused: 0
  Stopped: 0
 Images: 0
 Server Version: 20.10.17
 Storage Driver: overlay2
  Backing Filesystem: xfs
  Supports d_type: true
  Native Overlay Diff: true
  userxattr: false
 Logging Driver: json-file
 Cgroup Driver: cgroupfs
 Cgroup Version: 1
 Plugins:
  Volume: local
  Network: bridge host ipvlan macvlan null overlay
  Log: awslogs fluentd gcplogs gelf journald json-file local logentries splunk syslog
 Swarm: inactive
 Runtimes: io.containerd.runc.v2 io.containerd.runtime.v1.linux runc
 Default Runtime: runc
 Init Binary: docker-init
 containerd version: 10c12954828e7c7c9b6e0ea9b0c02b01407d3ae1
```

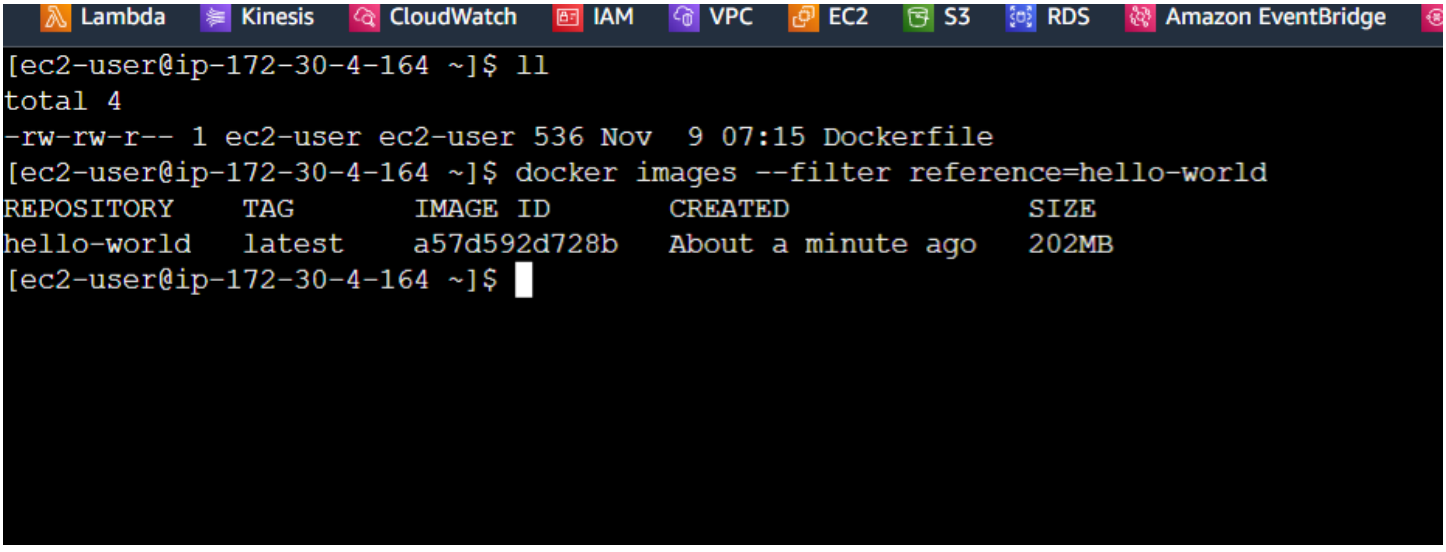
2. Create a Docker file

```
FROM ubuntu:18.04

# Install dependencies
RUN apt-get update && \
 apt-get -y install apache2

# Install apache and write hello world message
RUN echo 'Hello World!' > /var/www/html/index.html

# Configure apache
RUN echo '. /etc/apache2/envvars' > /root/run_apache.sh && \
 echo 'mkdir -p /var/run/apache2' >> /root/run_apache.sh && \
 echo 'mkdir -p /var/lock/apache2' >> /root/run_apache.sh && \
 echo '/usr/sbin/apache2 -D FOREGROUND' >> /root/run_apache.sh && \
 chmod 755 /root/run_apache.sh

EXPOSE 80

CMD /root/run_apache.sh
```

And build the docker file

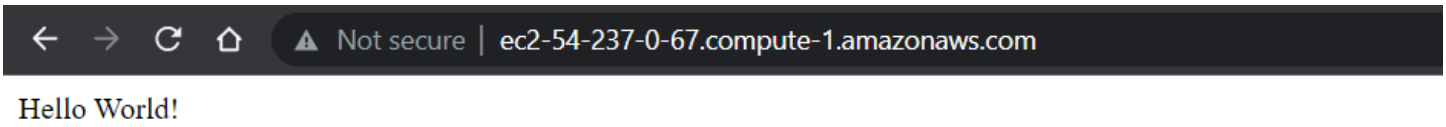
```
docker build -t hello-world .
```

```
docker images --filter reference=hello-world
```



Run the Docker

```
docker run -t -i -p 80:80 hello-world
```



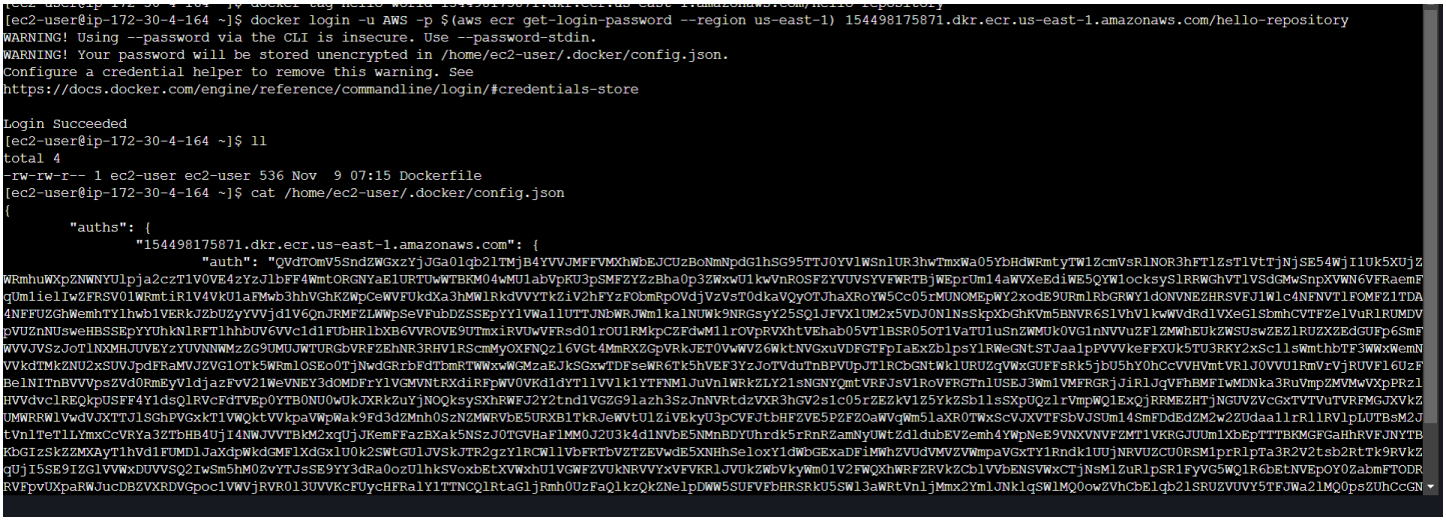
Push image to Elastic Container Registry

```
aws ecr create-repository --repository-name hello-repository --region region
```

```
[ec2-user@ip-172-30-4-164 ~]$ aws ecr create-repository --repository-name hello-repository --region us-east-1
{
    "repository": {
        "repositoryUri": "154498175871.dkr.ecr.us-east-1.amazonaws.com/hello-repository",
        "imageScanningConfiguration": {
            "scanOnPush": false
        },
        "encryptionConfiguration": {
            "encryptionType": "AES256"
        },
        "registryId": "154498175871",
        "imageTagMutability": "MUTABLE",
        "repositoryArn": "arn:aws:ecr:us-east-1:154498175871:repository/hello-repository",
        "repositoryName": "hello-repository",
        "createdAt": 1667978884.0
    }
}
[ec2-user@ip-172-30-4-164 ~]$
```

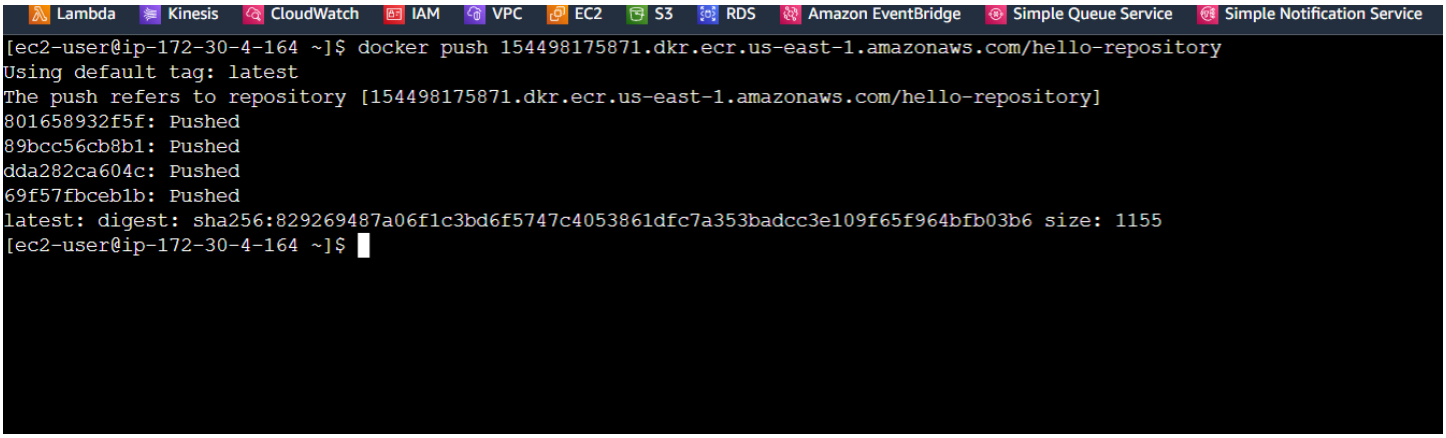
Tag image with Repository URI

```
docker tag hello-world 154498175871.dkr.ecr.us-east-1.amazonaws.com/hello-repository
```

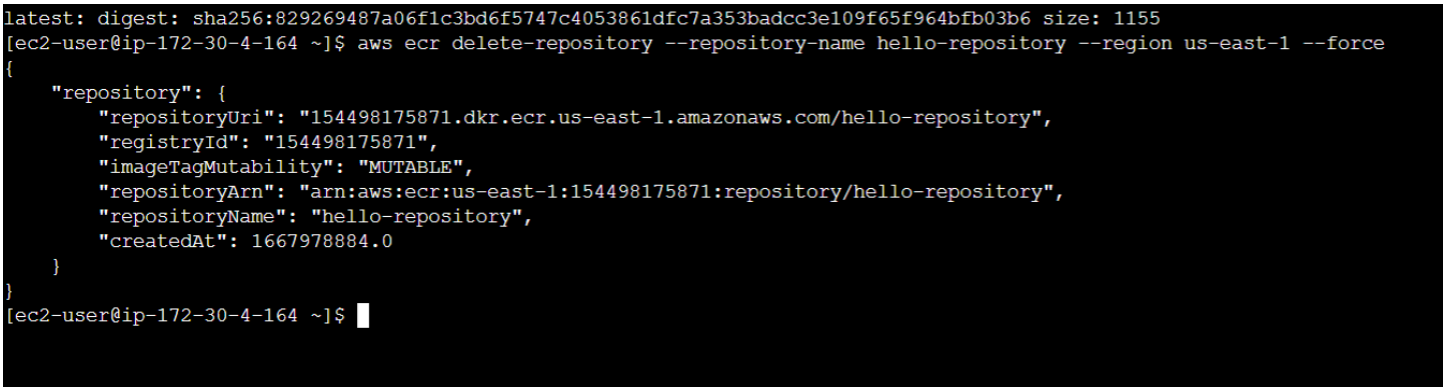


Push the Image to Amazon ECR

docker push 154498175871.dkr.ecr.us-east-1.amazonaws.com/hello-repository

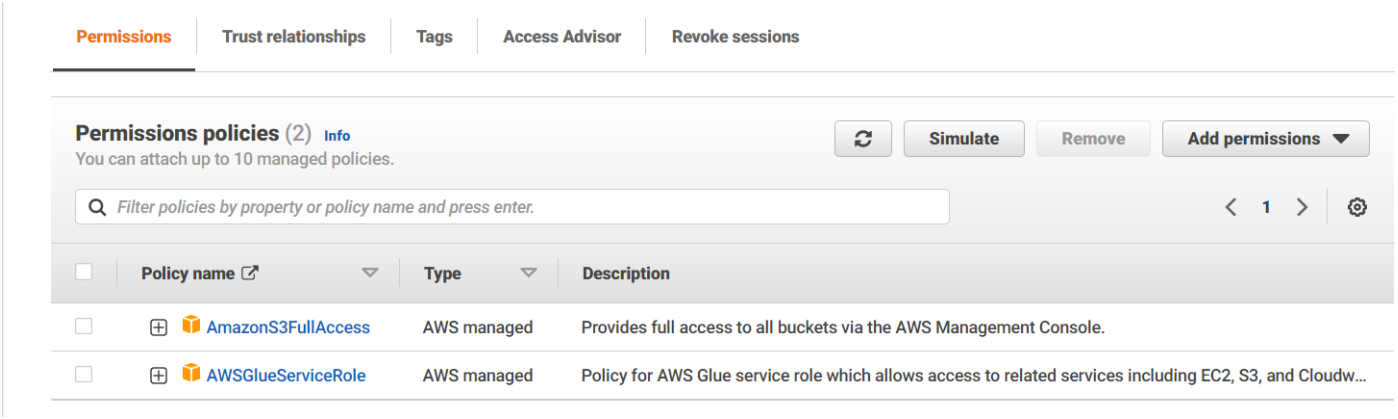


Clean Up:

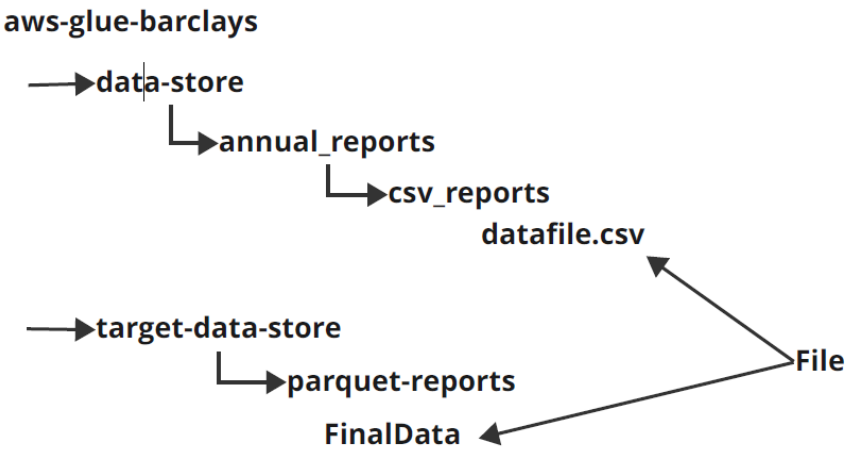


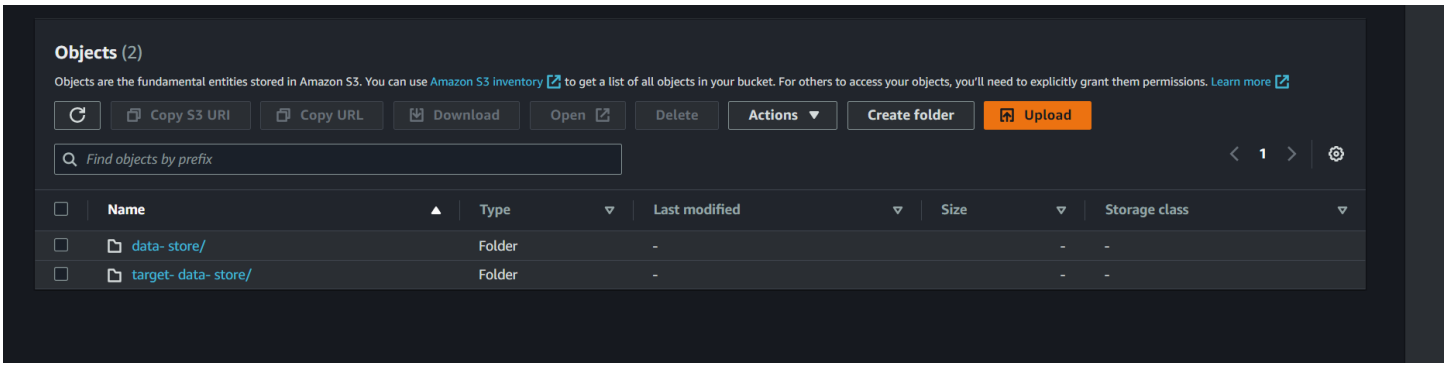
Glue Hands On

Create an IAM role with AWSGlueServiceRole and S3FullAccess

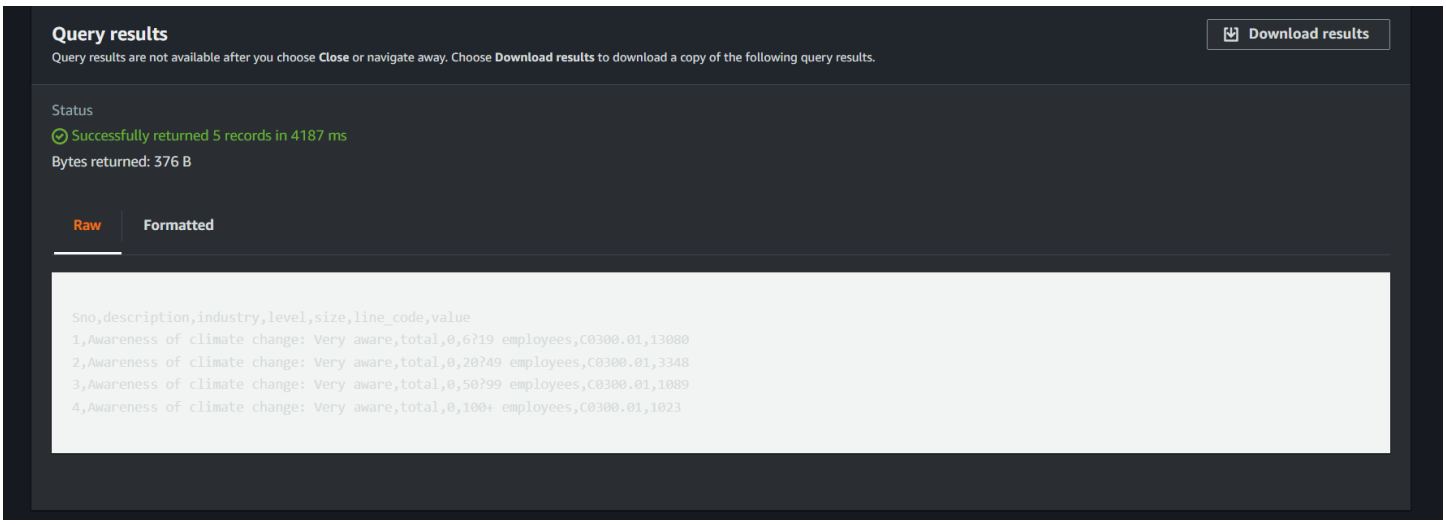


Create An S3 Bucket with the following Structure

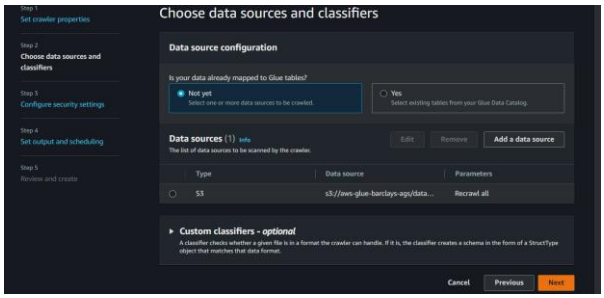


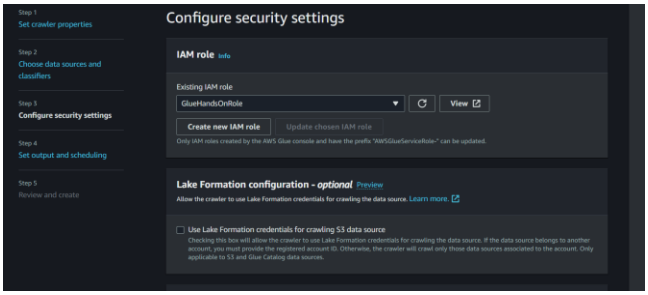


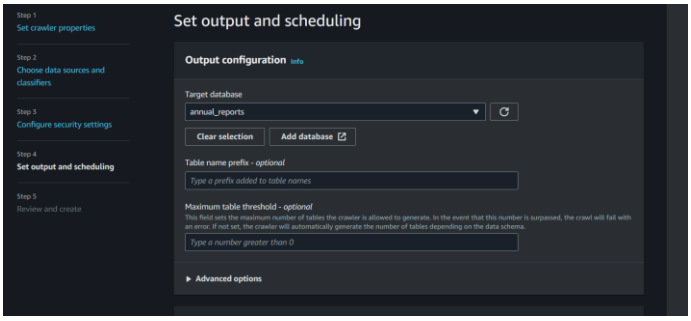
Data File:

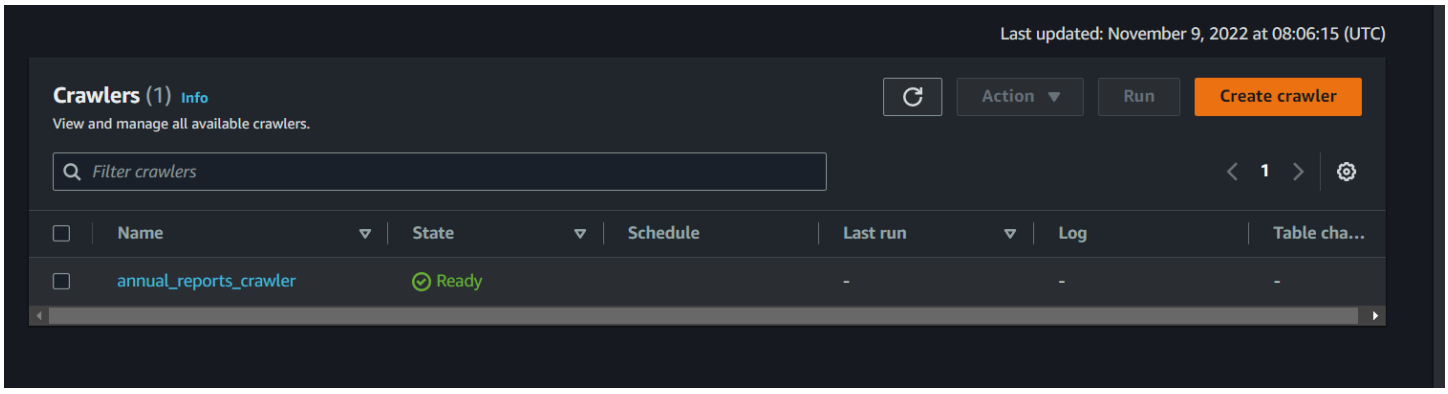


Glue: Configure DataBase information

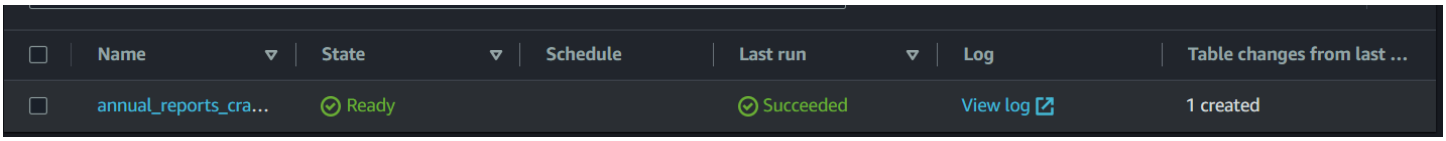




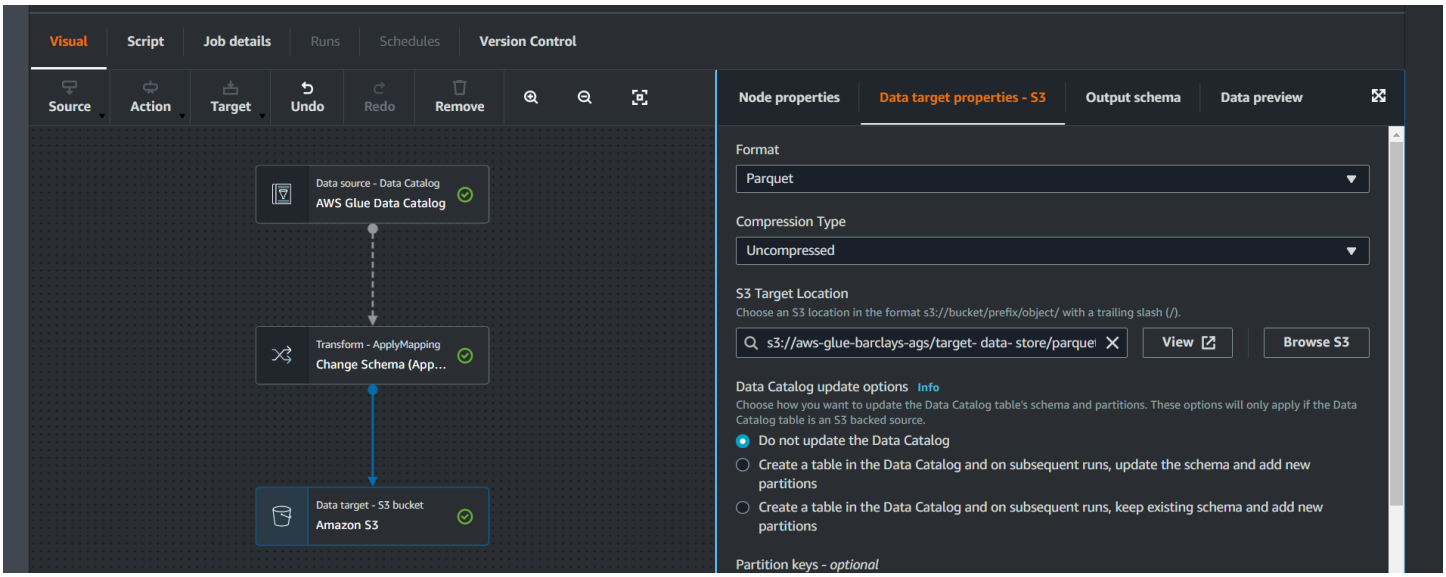




Run the Crawler



Create jobs from Glue Studio



Run the job and verify the output in s3