**Pre-Requirements:**

• Ec2 Instance

• Docker

• Aws-cli

**Step1:** **Connect to the server using the cmd or gitbash :**

• Ssh -i “fcra.pem” ubuntu@ip

Step2: Installation of Docker:

• apt-get install docker.io

• systemctl start docker

• systemctl enable docker

• systemctl status docker

•

**Step2:** **Installation of Aws Cli :**

• curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip" -o

"awscliv2.zip"

• unzip awscliv2.zip

• apt install unzip -y

• unzip awscliv2.zip

• sudo ./aws/install (To check the path where it installed)

• AWS –version

• AWS configure (enter the id and password details of the iam role which uhave

been created)

**Step3:**

**Create a role in AWS with the access of ec2 repository and admin access and create access**

**key:**

• for further requirements example below. \* \*

• Access key - XXXXXXXXXXXXX

• Secret access key: XXXXXXXXXXXXXX

• Region : ap-south-1

**Step4:**

**Once you completed the above steps use the Commands to generate the self-signed certificate:**

* openssl req -newkey rsa:4096 -x509 -sha256 -days 3650 -nodes -out protiviti.crt -keyout protiviti.key
* openssl pkcs12 -export -out protiviti.pfx -inkey protiviti.key -in protiviti.crt

**Step5**:

**Then include in the certificate path in docker example below.**

FROM mcr.microsoft.com/dotnet/aspnet:6.0-alpine AS base WORKDIR /app

EXPOSE 80

EXPOSE 443

RUN apk apk update && apk upgrade && \ apk add --no-cache openssl \

tzdata \

icu-libs>=67 \ krb5-libs>=1.18 \ libgcc>=10.3 \ libintl>=0.21 \ libssl1.1>=1.1.1 \ libstdc++>=10.3 \ zlib>=1.2.11

ENV TZ=Asia/Calcutta

ENV ASPNETCORE\_ENVIRONMENT=testing ENV DOTNET\_RUNNING\_IN\_CONTAINER=true

ENV DOTNET\_SYSTEM\_GLOBALIZATION\_INVARIANT=false

FROM mcr.microsoft.com/dotnet/sdk:6.0-alpine AS build WORKDIR /src

COPY ["FCRA.Web/FCRA.Web.csproj", "FCRA.Web/"]

COPY ["FCRA.Common/FCRA.Common.csproj", "FCRA.Common/"] COPY ["FCRA.Repository/FCRA.Repository.csproj", "FCRA.Repository/"] COPY ["FCRA.Models/FCRA.Models.csproj", "FCRA.Models/"]

COPY ["FCRA.ViewModels/FCRA.ViewModels.csproj", "FCRA.ViewModels/"] RUN dotnet restore "FCRA.Web/FCRA.Web.csproj"

COPY . .

WORKDIR "/src/FCRA.Web"

RUN dotnet build "FCRA.Web.csproj" -c Release -o /app/build FROM build AS publish

RUN dotnet publish "FCRA.Web.csproj" -c Release -o /app/publish /p:UseAppHost=false

//ARG CERT\_PASSWORD

//RUN dotnet dev-certs https -ep /app/ewraprotiviti.pfx -p ewraprotiviti@123

FROM base AS final WORKDIR /app

COPY --from=publish /app/publish .

//RUN mkdir -p /https

//COPY ewraprotiviti.pfx /app COPY ewraprotiviti.pfx /https

ENTRYPOINT ["dotnet", "FCRA.Web.dll"]

* docker build -t <nameof\_dockerfile> .
* docker build -t abcd .

**Step6**:

**Then run the application with environment variables in docker using the command below.**

docker run -d -p 80:80 -p 443:443 \

-e ASPNETCORE\_URLS="https://+;http://+" \

-e ASPNETCORE\_HTTPS\_PORT=443 \

-e ASPNETCORE\_Kestrel Certificates Default Password="protiviti@123" \

-e ASPNETCORE\_Kestrel Certificates Default Path="/https/protiviti.pfx" \

-e MetadataAddress="https://login.microsoftonline.com/ba04dd9d-19c9-423e-85c1- 63bc63f9ff4c/federationmetadata/2007-06/federationmetadata.xml?appid=d58bfe58-9cf7-4e27-84ec- 39fd728ab156" \

-e username="sa" \

-e password="fcra@123" \

-e host="13.201.123.96" \

-e port="" \

-e bucketname="" \

-e AWSRegion="ap-south-1" \

-e dbInstanceIdentifier="RISKDBADCB" \

-e engine="" \

-e IsThroughSMTP="N" \

-e From="" \

-e SMTPUsername="" \

-e SMTPPassword="" \

-e SMTPHost="" \

-e SMTPPort="587" \

-e RealmUrl="https://ec2-3-6-40-111.ap-south-1.compute.amazonaws.com/" \

-e certificatepath="" \

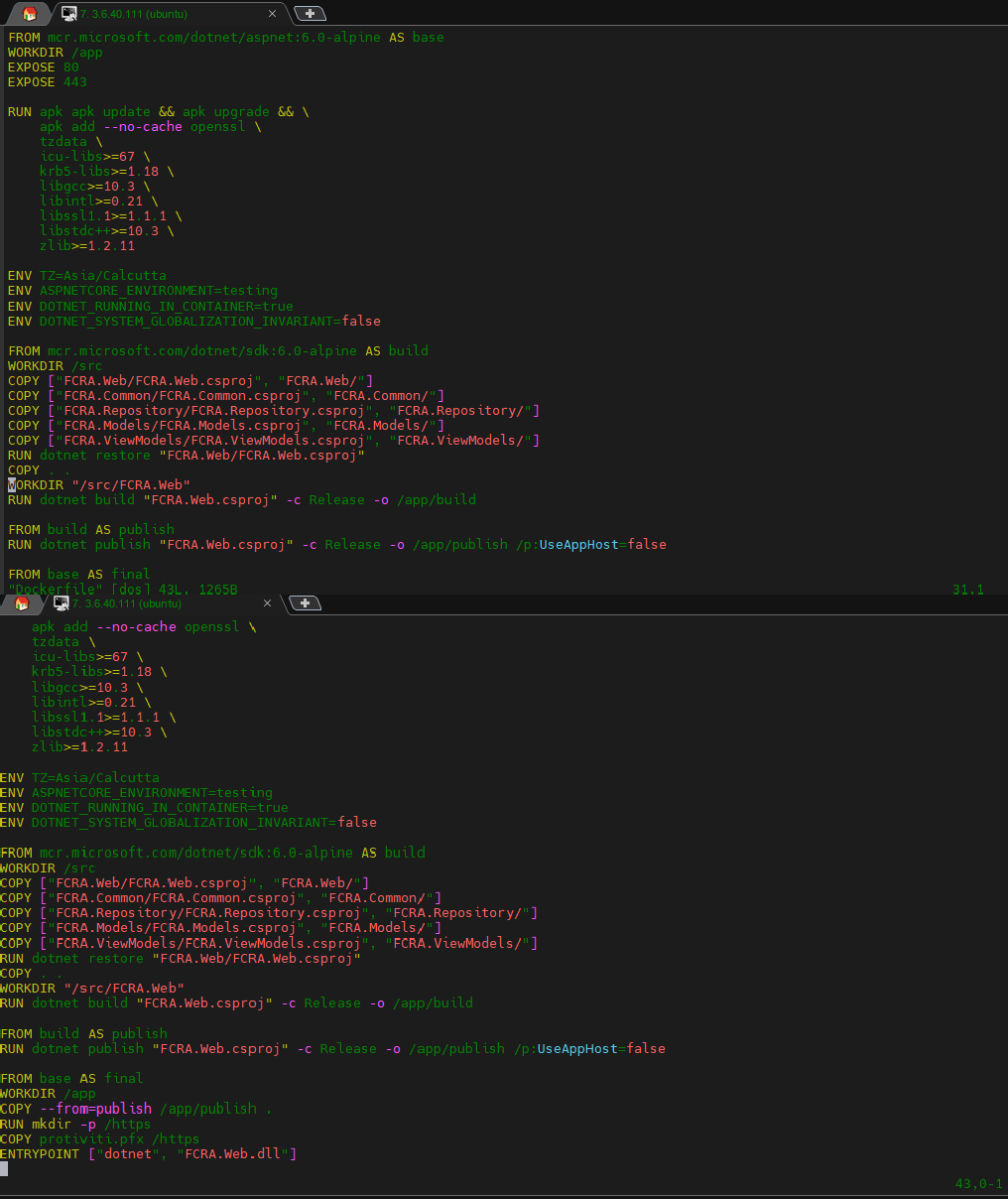
-e certificatepassword="" \

-e httpport="" \

-e httpsport="" \

-e IsSSOApplicable="Y" \

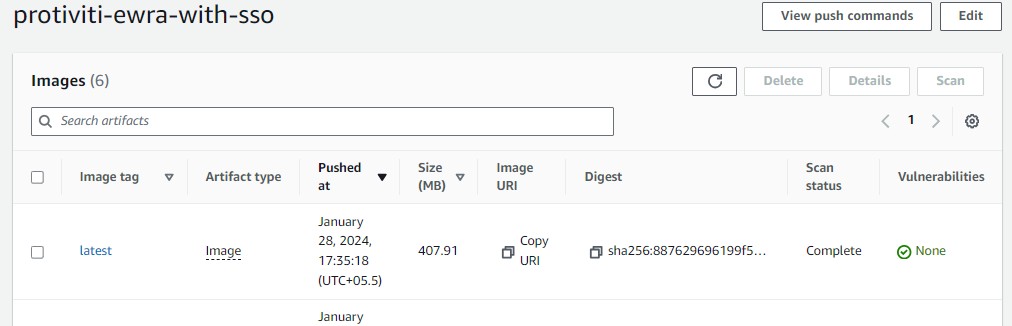
-e IsEnvironmentVariableApplicable="Y" \ Abcd



**Step7** :

**Push the image into ECR with the commands below.**

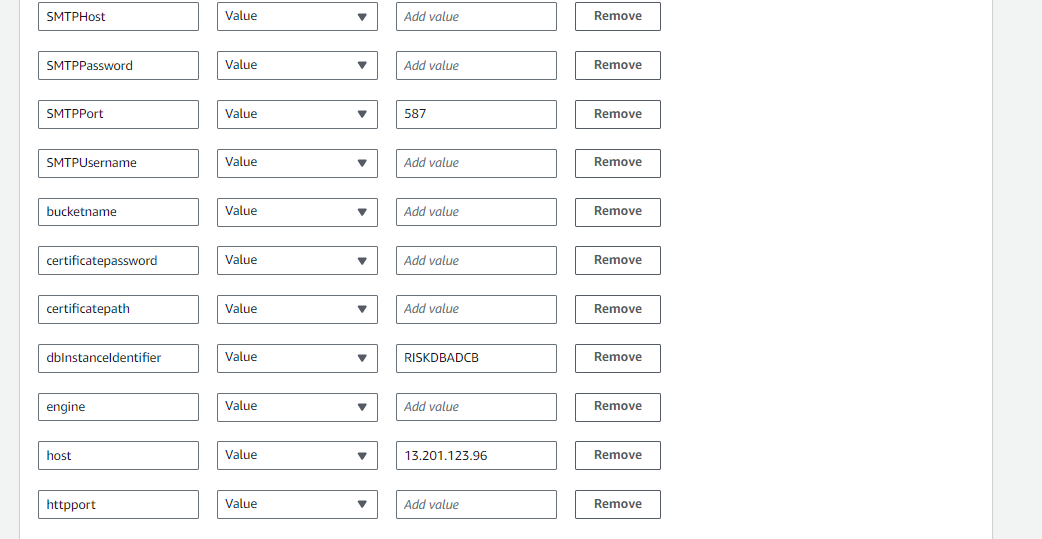
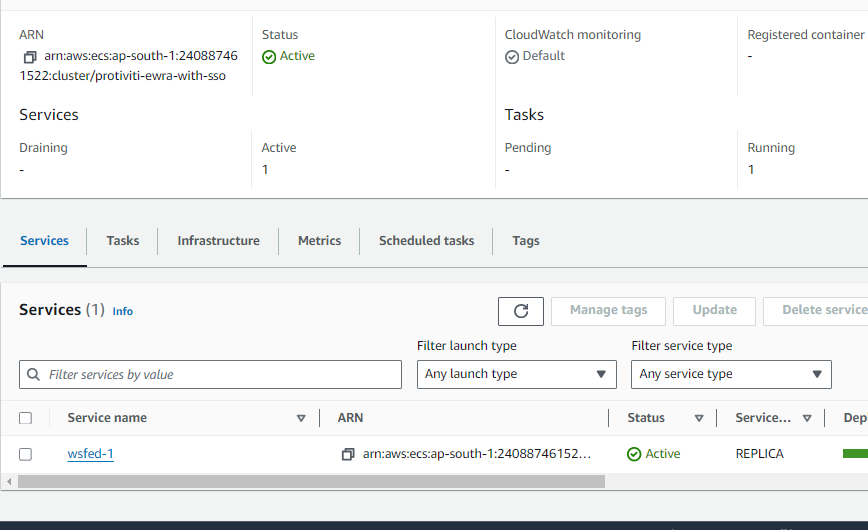
* aws ecr get-login-password --region ap-south-1 | docker login --username AWS --password-stdin 240887461522.dkr.ecr.ap-south-1.amazonaws.com
* docker tag abcd:latest 240887461522.dkr.ecr.ap-south-1.amazonaws.com/protiviti-ewra-with- sso:latest
* docker push 240887461522.dkr.ecr.ap-south-1.amazonaws.com/protiviti-ewra-with-sso:latest

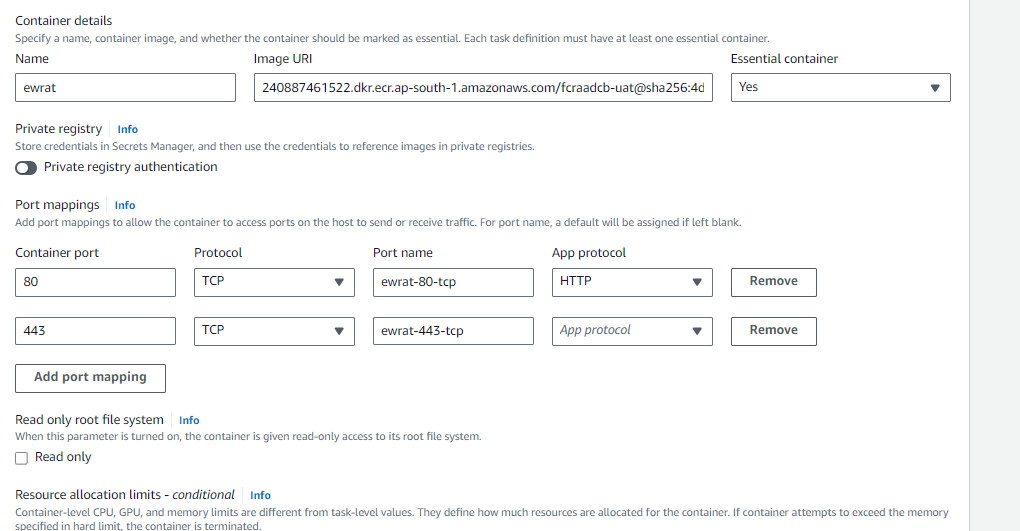


**Step8**:

**Then Create a task definition in AWS ECS and attach the environment variables with the values.**

* + Create a task definition with the below configurations.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

Step9:

**Then attach the created task for the Cluster shown in the below image**.

A screenshot of a computer

Description automatically generated





