

## Steps\_for\_the\_ssl\_and\_deploying\_in\_ECS:

### Step1:

#### 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
```

### Step2:

**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 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 .
```

### Step3 :

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

```
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
```

```
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

FROM base AS final
"dockerfile" [dos] 43L, 1265B 31.1
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

FROM base AS final
WORKDIR /app
COPY --from=publish /app/publish .
RUN mkdir -p /https
COPY protivity.pfx /https
ENTRYPOINT ["dotnet", "FCRA.Web.dll"] 43,0-1
```

Step4 :

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

protiviti-ewra-with-sso

View push commands

Edit

Images (6)

Q Search artifacts

< 1 >

<input type="checkbox"/>	Image tag ▾	Artifact type	Pushed at ▾	Size (MB) ▾	Image URI	Digest	Scan status	Vulnerabilities
<input type="checkbox"/>	latest	Image	January 28, 2024, 17:35:18 (UTC+05.5)	407.91	<div> Copy URI</div>	<div> sha256:887629696199f5...</div>	Complete	<div> None</div>
			January					

Step5 :

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

- Create a cluster and attach that task to it.

```

38         "name": "AWSRegion",
39         "value": "ap-south-1"
40     },
41     {
42         "name": "SMTPPort",
43         "value": "587"
44     },
45     {
46         "name": "password",
47         "value": "fcra@123"
48     },
49     {
50         "name": "SMTPPassword",
51         "value": ""
52     },
53     {
54         "name": "engine",
55         "value": ""
56     },
57     {
58         "name": "bucketname",
59         "value": ""
60     },
61     {
62         "name": "httpport",
63         "value": ""
64     },
65     {
66         "name": "host"

```

```

1  {
2      "taskDefinitionArn": "arn:aws:ecs:ap-south-1:240887461522:task-definition/protiviti-ewra-with-sso:12",
3      "containerDefinitions": [
4          {
5              "name": "protiviti-ewra-with-sso",
6              "image": "240887461522.dkr.ecr.ap-south-1.amazonaws.com/protiviti-ewra-with-sso:latest",
7              "cpu": 0,
8              "portMappings": [
9                  {
10                     "name": "protiviti-ewra-with-sso-80-tcp",
11                     "containerPort": 80,
12                     "hostPort": 80,
13                     "protocol": "tcp",
14                     "appProtocol": "http"
15                 },
16                 {
17                     "name": "protiviti-ewra-with-sso-443-tcp",
18                     "containerPort": 443,
19                     "hostPort": 443,
20                     "protocol": "tcp"
21                 }
22             ],
23             "essential": true,
24             "environment": [
25                 {
26                     "name": "IsSSOApplicable",
27                     "value": "Y"

```

JSON

```
175     {
176         "name": "ecs.capability.task-eni"
177     },
178     {
179         "name": "com.amazonaws.ecs.capability.docker-remote-api.1.1.29"
180     }
181 ],
182 "placementConstraints": [],
183 "compatibilities": [
184     "EC2",
185     "FARGATE"
186 ],
187 "requiresCompatibilities": [
188     "FARGATE"
189 ],
190 "cpu": "1024",
191 "memory": "3072",
192 "runtimePlatform": {
193     "cpuArchitecture": "X86_64",
194     "operatingSystemFamily": "LINUX"
195 },
196 "registeredAt": "2024-01-28T12:08:07.696Z",
197 "registeredBy": "arn:aws:iam::240887461522:user/sachchidanandjha",
198 "tags": []
199 }
```

Health and metrics

Tasks

Logs

Deployments

Events

Configuration and networking

Tags

Deployment configuration Info

View pipelines

Deployment status

Deployment type

Platform version

Min and max running tasks

Completed

ECS

LATEST

100% min and 200% max

► Deployment failure detection

► Task placement strategy and constraints

Deployments (1) Info

Filter deployments

< 1 >

Start date

Status

Failed tasks

Tasks

Version

Task definition

January 26, 2024 at 10:09 (UTC+5:30)

Primary 100%

0

1 Running | 0 Pending | 1 Desired

1.4.0

[ewrat-adcb-fcra](#)