

Protiviti Project Using ECS and Faregate:

Pre-requirement make sure the dotnet project is running or not in the local do all needful.

Step1:

Create an RDS named database1 and then set the username and password and public access should be yes.

Ex:

Database service url: **database-1.cvdr8z18g9n4.ap-southeast-2.rds.amazonaws.com**

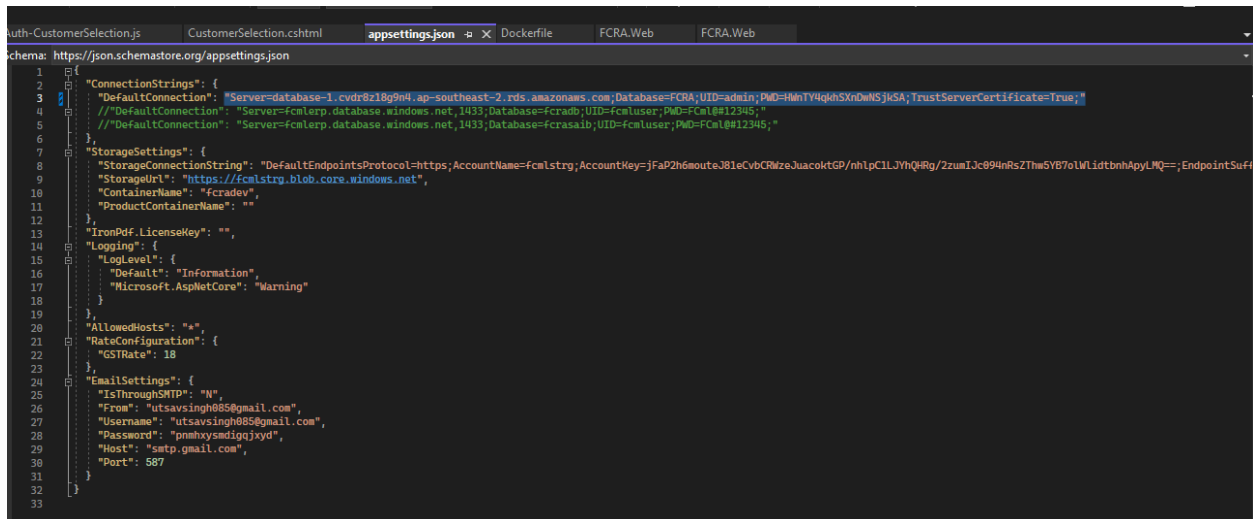
Admin

Shashi@123

Step2:

Goto appsetting.json file and add rds. details as shown below.

"DefaultConnection": "Server=database-1.cvdr8z18g9n4.ap-southeast-2.rds.amazonaws.com;Database=FCRA;UID=admin;PWD=HmNTY4qkhSXnDwNSjkSA;TrustServerCertificate=True;"

A screenshot of a code editor showing the 'appsettings.json' file. The file contains configuration settings for a .NET application. The 'DefaultConnection' string is highlighted in blue, showing the RDS connection details: "Server=database-1.cvdr8z18g9n4.ap-southeast-2.rds.amazonaws.com;Database=FCRA;UID=admin;PWD=HmNTY4qkhSXnDwNSjkSA;TrustServerCertificate=True;". The editor also shows other settings like 'StorageSettings', 'Logging', and 'EmailSettings'. The file is open in a window titled 'appsettings.json' with a schema link to 'https://json.schemastore.org/appsettings.json'.

Step3:

Create a Ec2 server in ubuntu .

Step4:

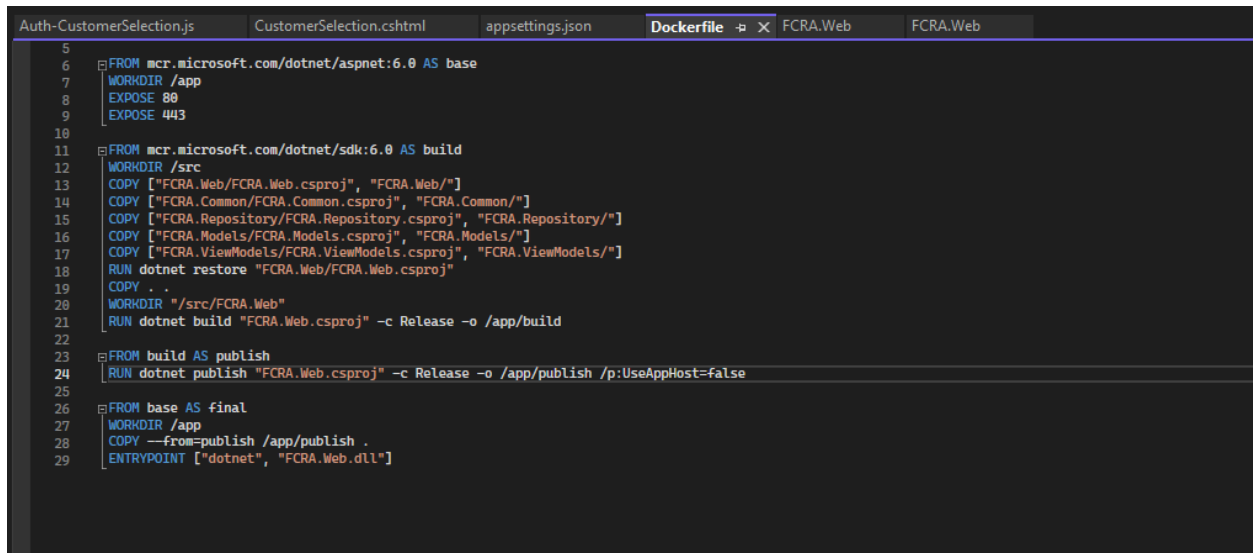
Pre-requirements are docker.io and git.

Step5:

Git clone <project>

Step6:

Visual studio docker file copy and create a docker file in ec2.



```
5
6 FROM mcr.microsoft.com/dotnet/aspnet:6.0 AS base
7 WORKDIR /app
8 EXPOSE 80
9 EXPOSE 443
10
11 FROM mcr.microsoft.com/dotnet/sdk:6.0 AS build
12 WORKDIR /src
13 COPY ["FCRA.Web/FCRA.Web.csproj", "FCRA.Web/"]
14 COPY ["FCRA.Common/FCRA.Common.csproj", "FCRA.Common/"]
15 COPY ["FCRA.Repository/FCRA.Repository.csproj", "FCRA.Repository/"]
16 COPY ["FCRA.Models/FCRA.Models.csproj", "FCRA.Models/"]
17 COPY ["FCRA.ViewModels/FCRA.ViewModels.csproj", "FCRA.ViewModels/"]
18 RUN dotnet restore "FCRA.Web/FCRA.Web.csproj"
19 COPY . .
20 WORKDIR "/src/FCRA.Web"
21 RUN dotnet build "FCRA.Web.csproj" -c Release -o /app/build
22
23 FROM build AS publish
24 RUN dotnet publish "FCRA.Web.csproj" -c Release -o /app/publish /p:UseAppHost=false
25
26 FROM base AS final
27 WORKDIR /app
28 COPY --from=publish /app/publish .
29 ENTRYPOINT ["dotnet", "FCRA.Web.dll"]
```

And then make sure check app.json file in ec2 that have given the server path or not.

Step7:

Create a docker image using below commands :

- `docker build -t dockerfcraimage .`
- `docker tag dockerfcraimage shashikumar023/dockerfcraimage`
- `docker push shashikumar023/dockerfcraimage`

Step8:

Create a ecs cluster and task-definition and attach the docker img name on task-definition.

Step9:

Create a service and tasks.



Add admin@admin.com

Pass: you know