Installation document for Programming Task

Contents

[Objective 2](#_Toc70569312)

[Prerequisite 2](#_Toc70569313)

[Current Design 2](#_Toc70569314)

[Assumptions: 2](#_Toc70569315)

[Build 2](#_Toc70569316)

[Run 10](#_Toc70569317)

[Clean up 10](#_Toc70569318)

[Ideal design 10](#_Toc70569319)

[Troubleshooting 10](#_Toc70569320)

# Objective

Create a docker container/image that performs the following:

  Implements a webserver which accepts requests to manage a PostgreSQL DB

                1. Connects to a PostgreSQL DB

                2. Create a table with columns:  id, timestamp, JSONB

                3. Insert an entry into the table

                5  List the contents of the DB

# Prerequisite

1. Install **aws cli** on your laptop <https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2-windows.html#cliv2-windows-install>
2. Install **python 3.6 or higher** on your laptop <https://www.python.org/downloads/>
3. Install **Docker Desktop** on your laptop <https://docs.docker.com/docker-for-windows/install/>
4. **Personal AWS Account log in Preferred**. We will be cleaning up resources with single click after work is completed.
5. Install Git client <https://git-scm.com/downloads>
6. Windows Powershell
7. Postman <https://www.postman.com/downloads/>

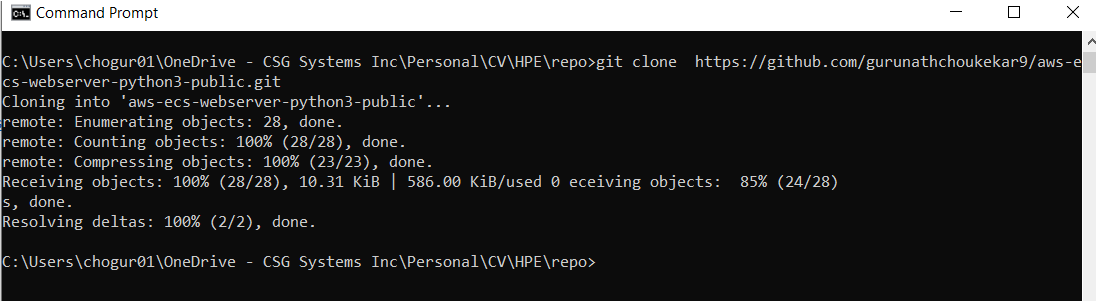
# Current Design

Need to upload visio diagram here

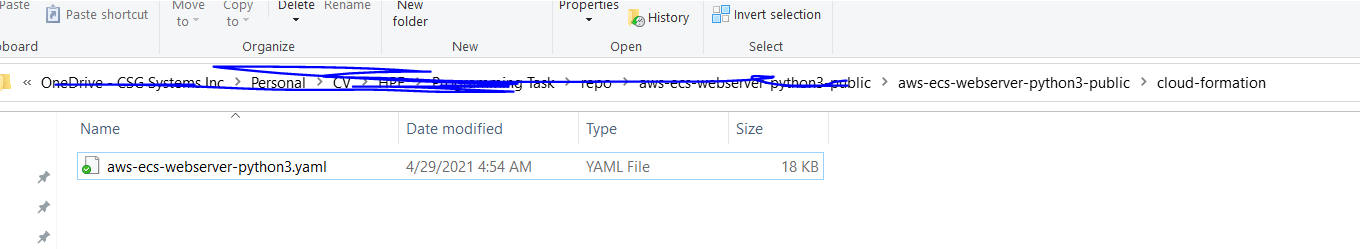
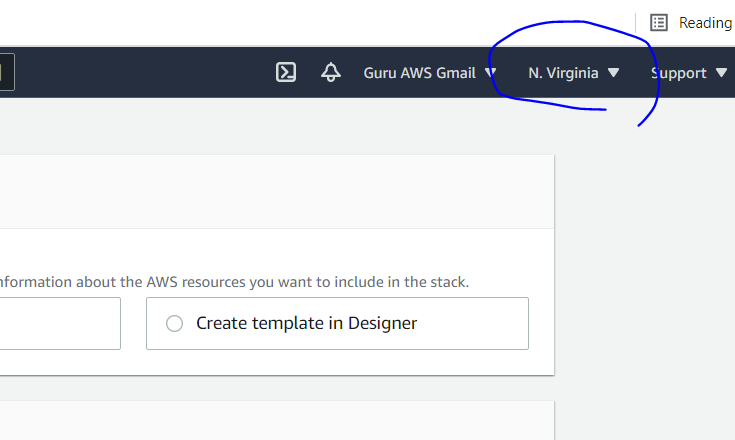
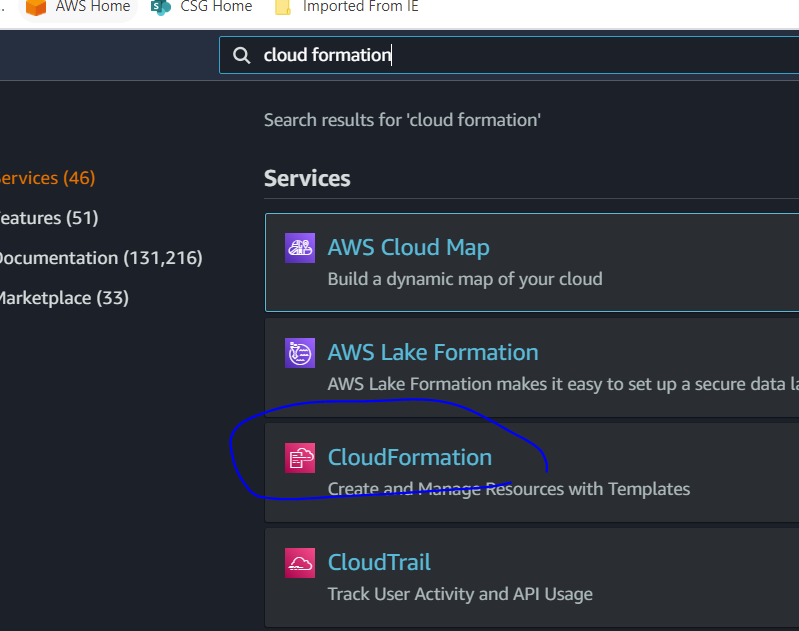
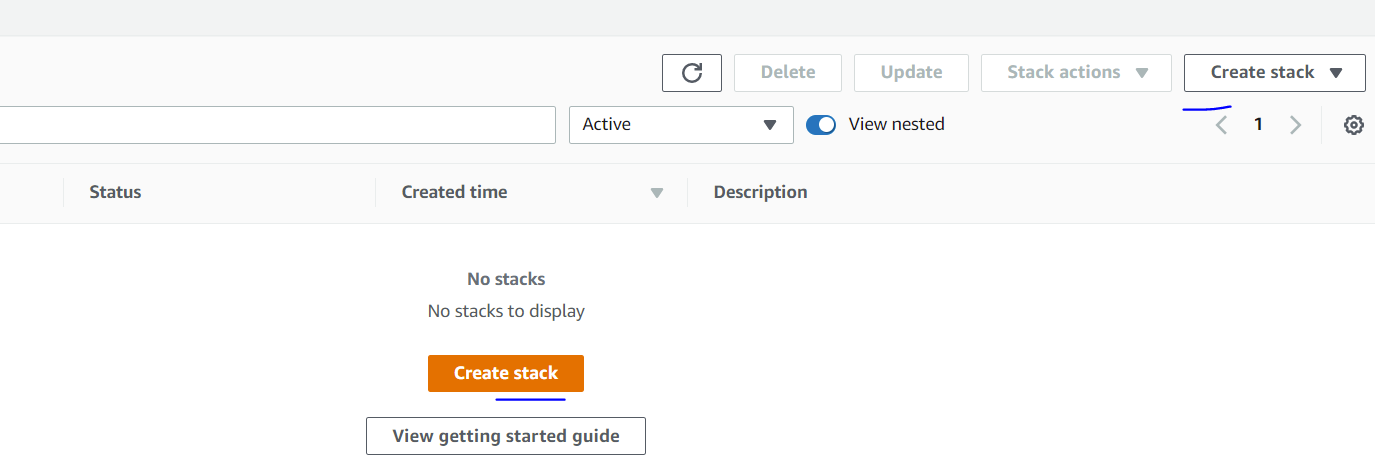
# Assumptions:

1. Prerequisite software installation is completed.
2. **Personal AWS Account log in Preferred**. We will be cleaning up resources with single click after work is completed.
3. **Use AWS Region us-east-1 (N Virginia) ONLY .** Cloud Formation template will not support other region this time

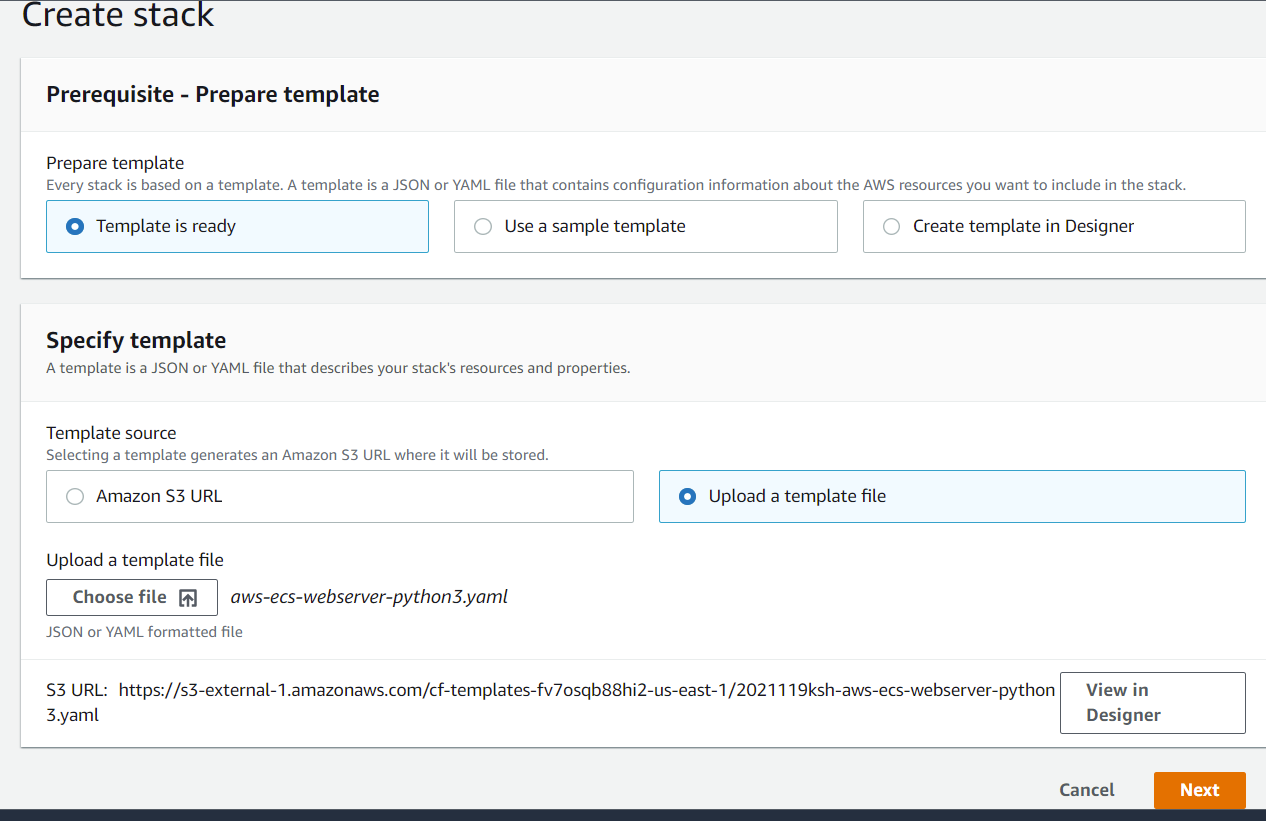
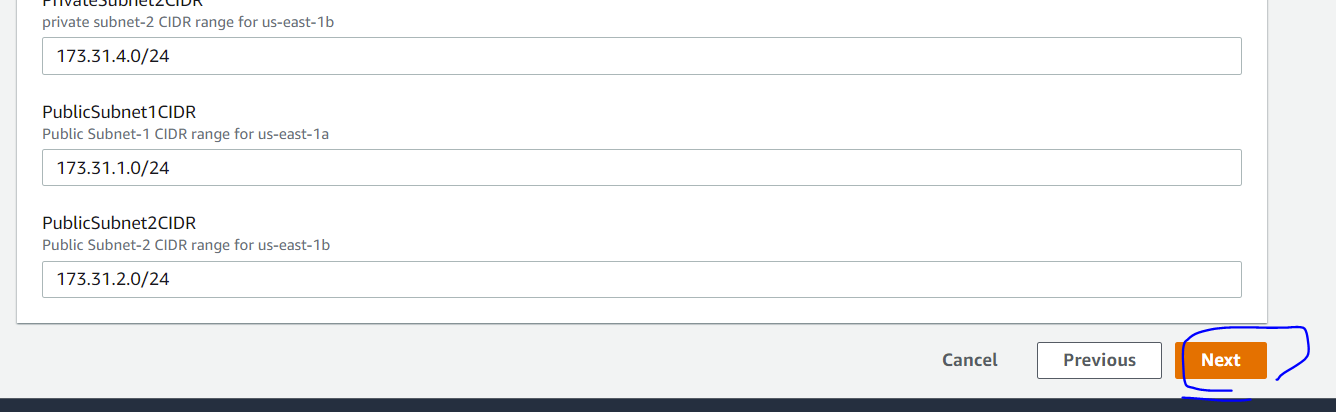
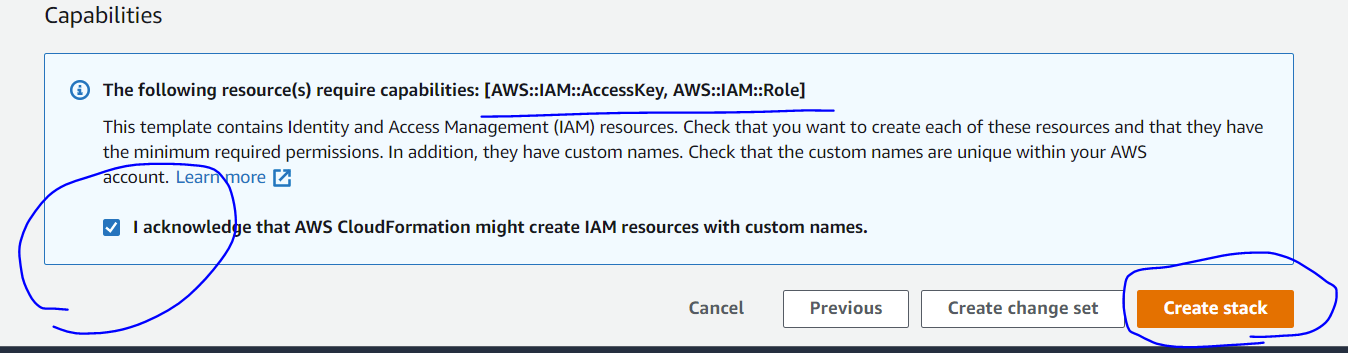
# Build

1. We need to run Cloud formation template from git repo here
2. download git code on your laptop <https://github.com/gurunathchoukekar9/aws-ecs-webserver-python3-public.git>
   1. git clone <https://github.com/gurunathchoukekar9/aws-ecs-webserver-python3-public.git>
   2. 
3. Go to cloud-formation folder

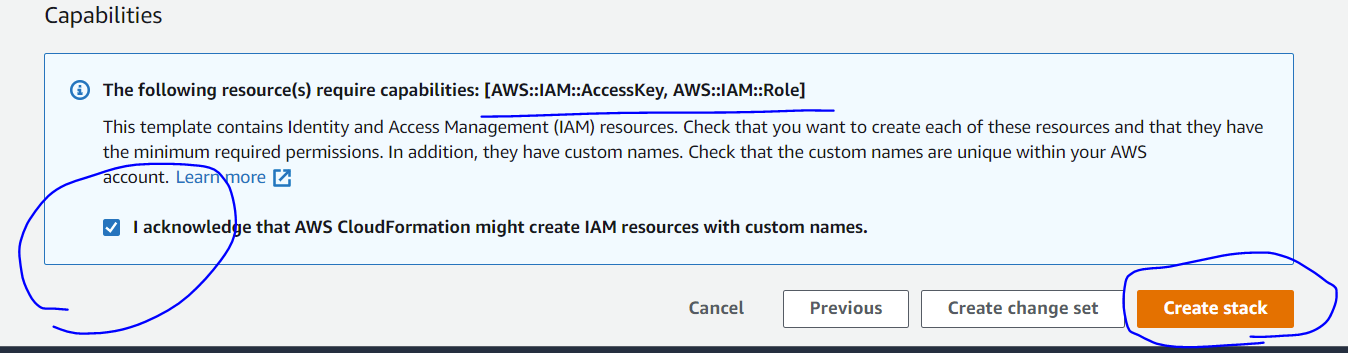
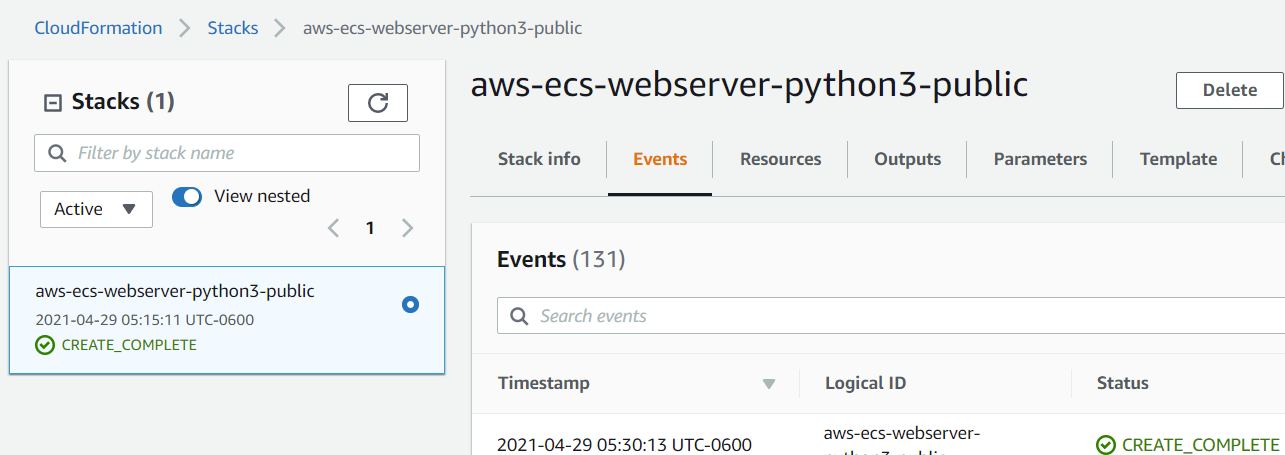
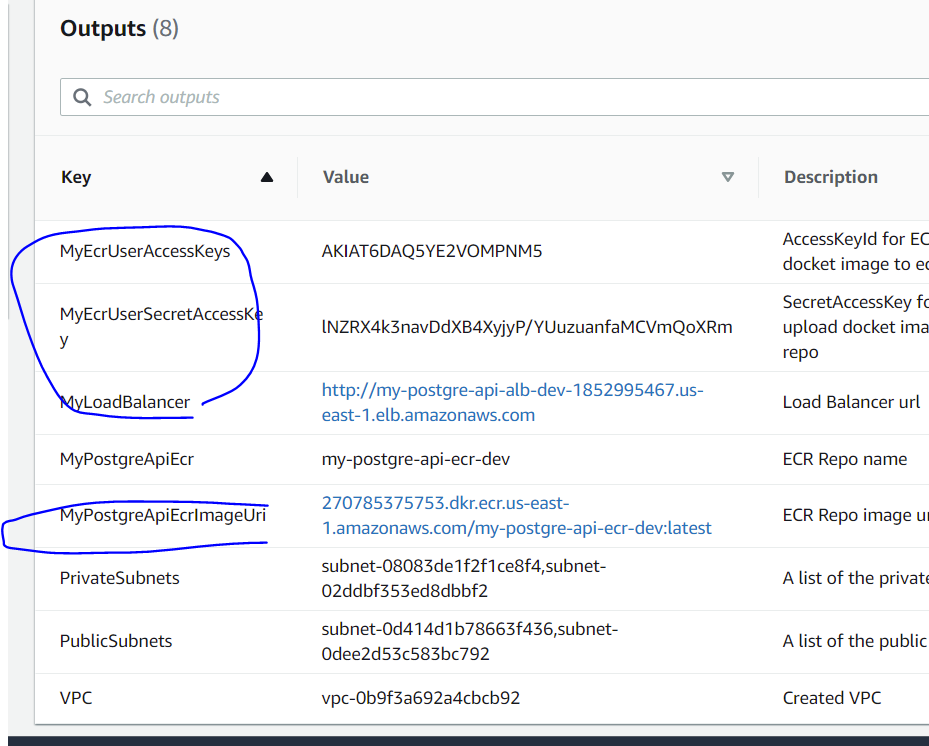
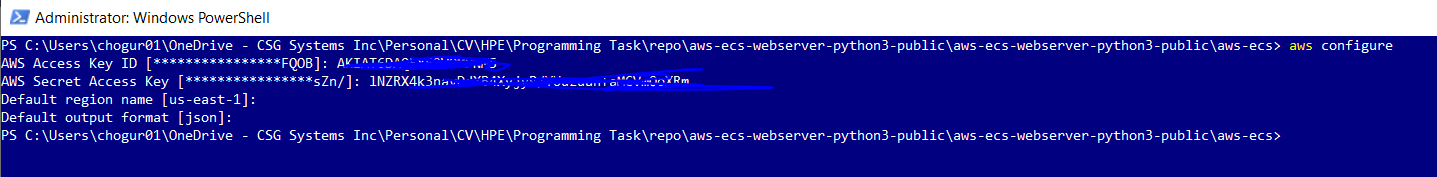
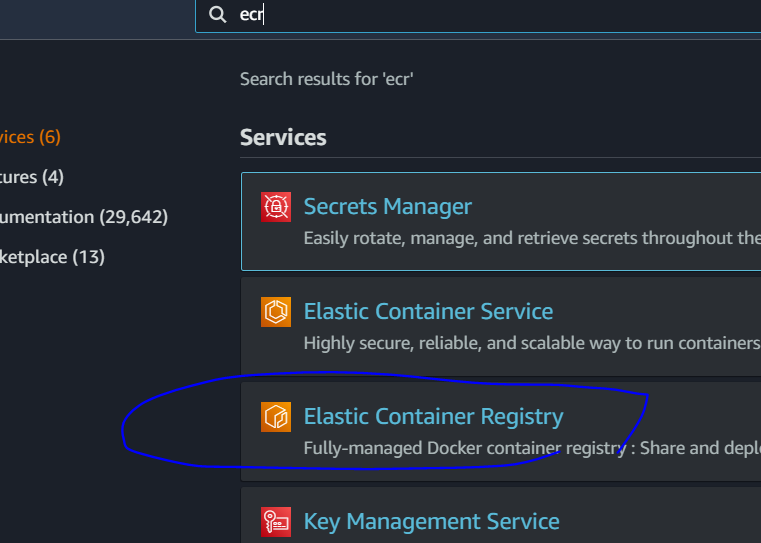
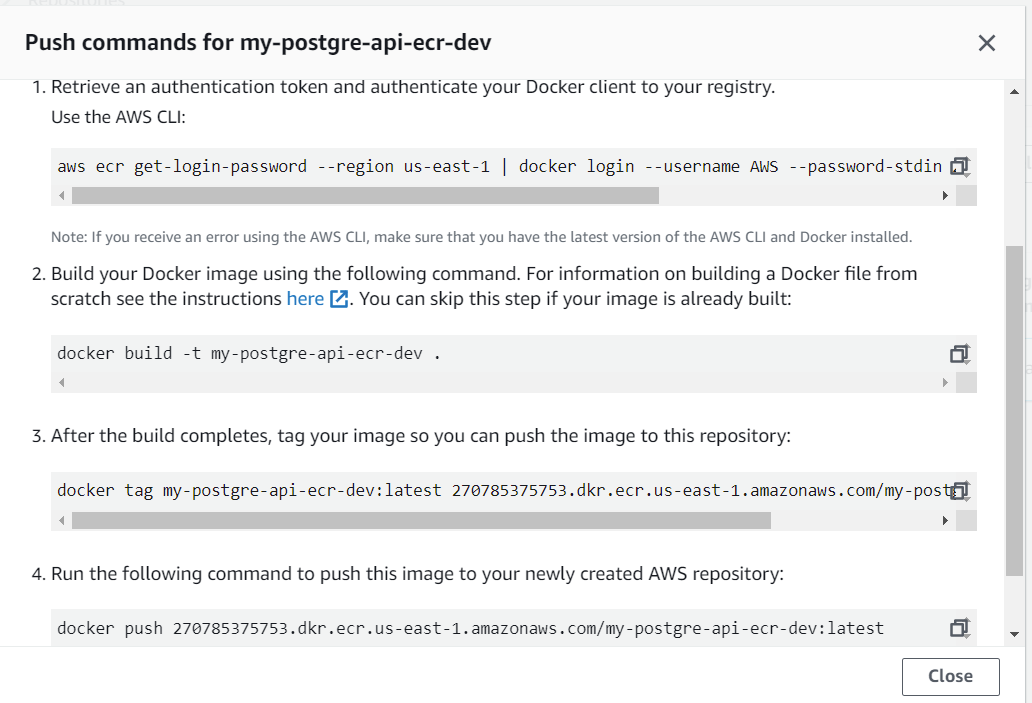
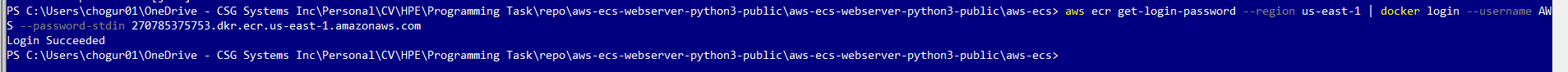
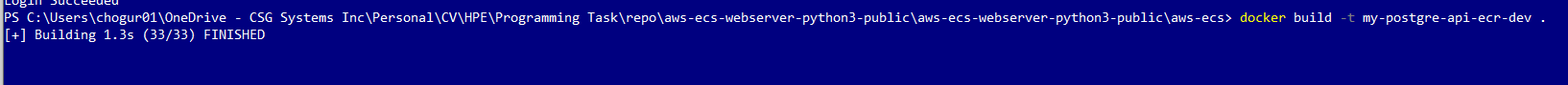
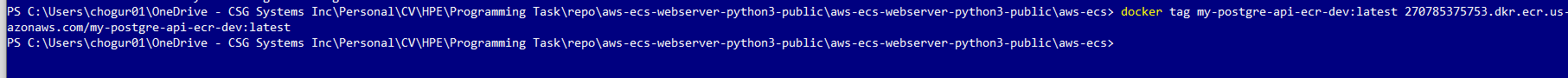
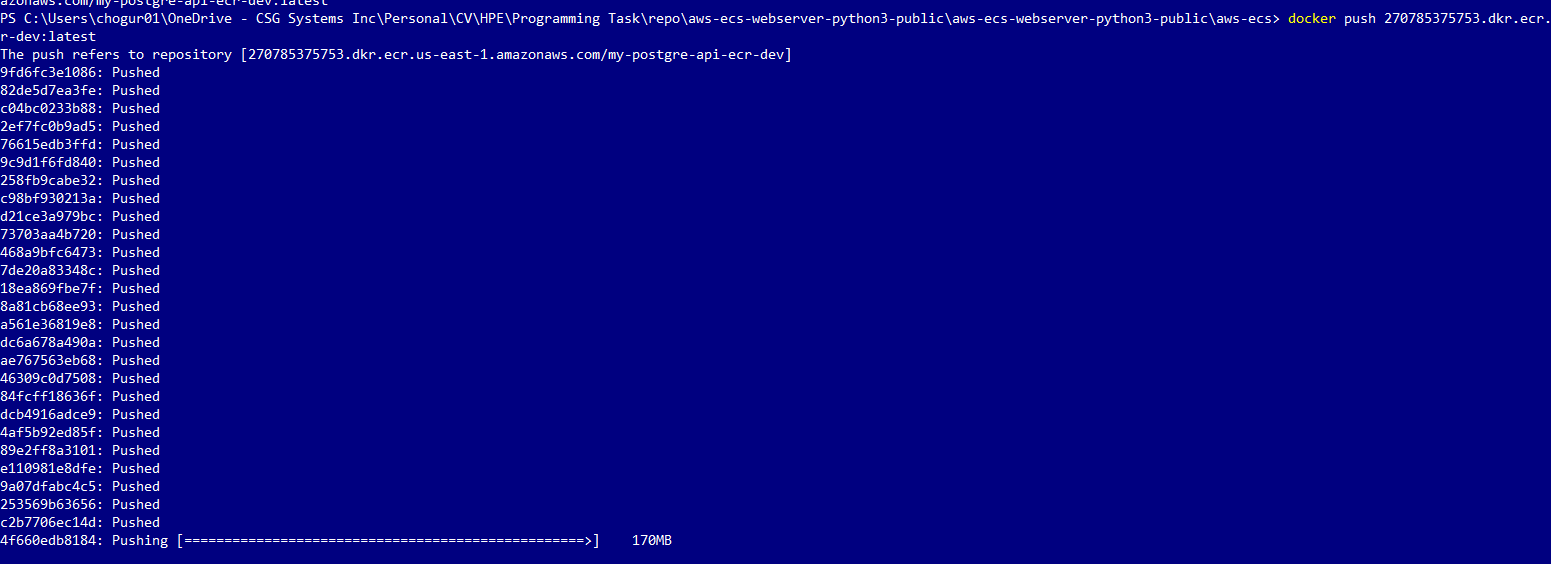
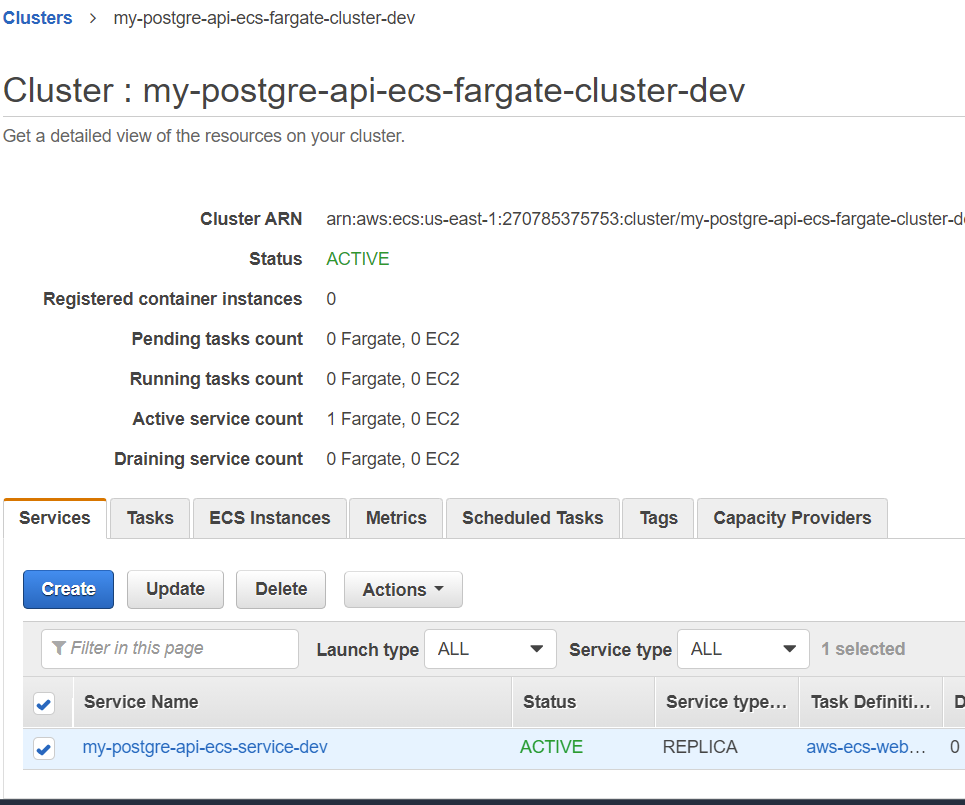
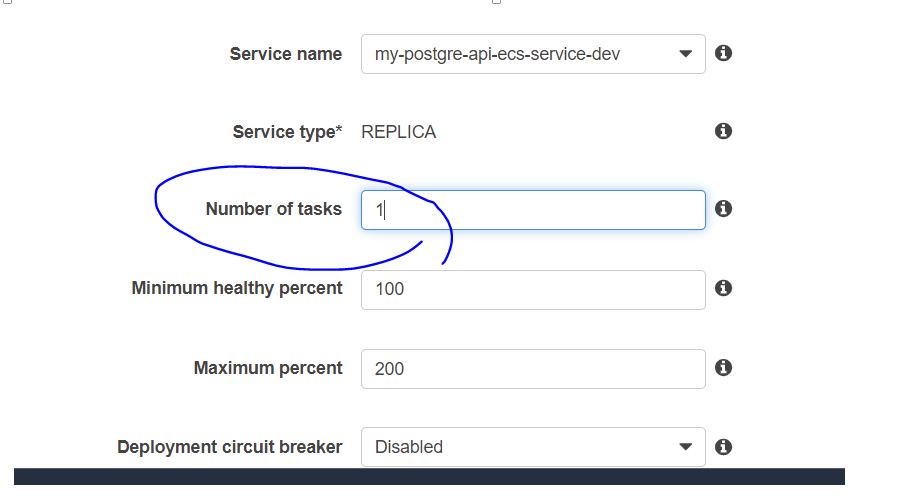
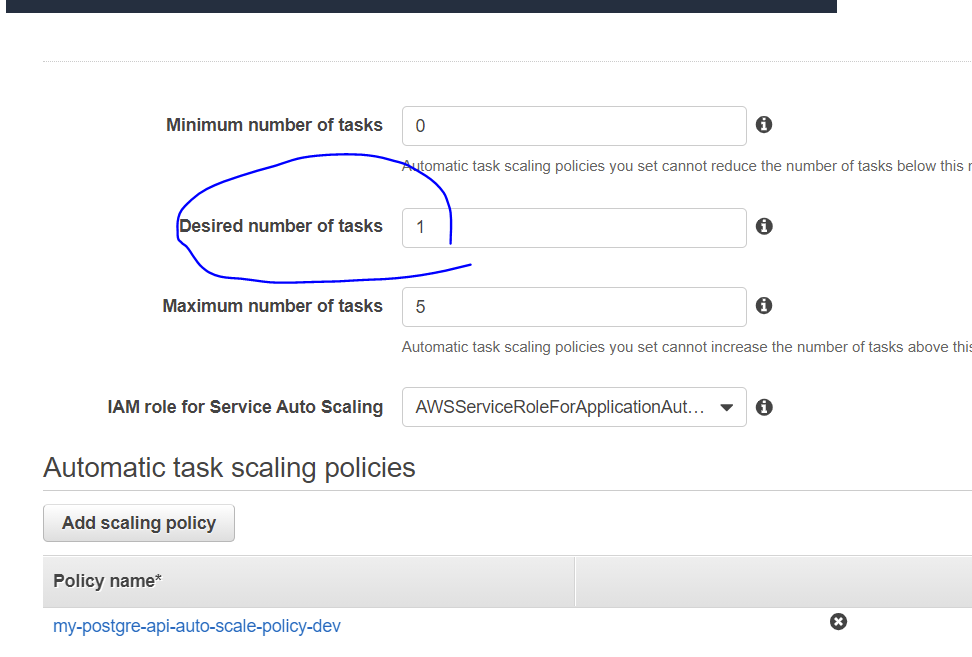
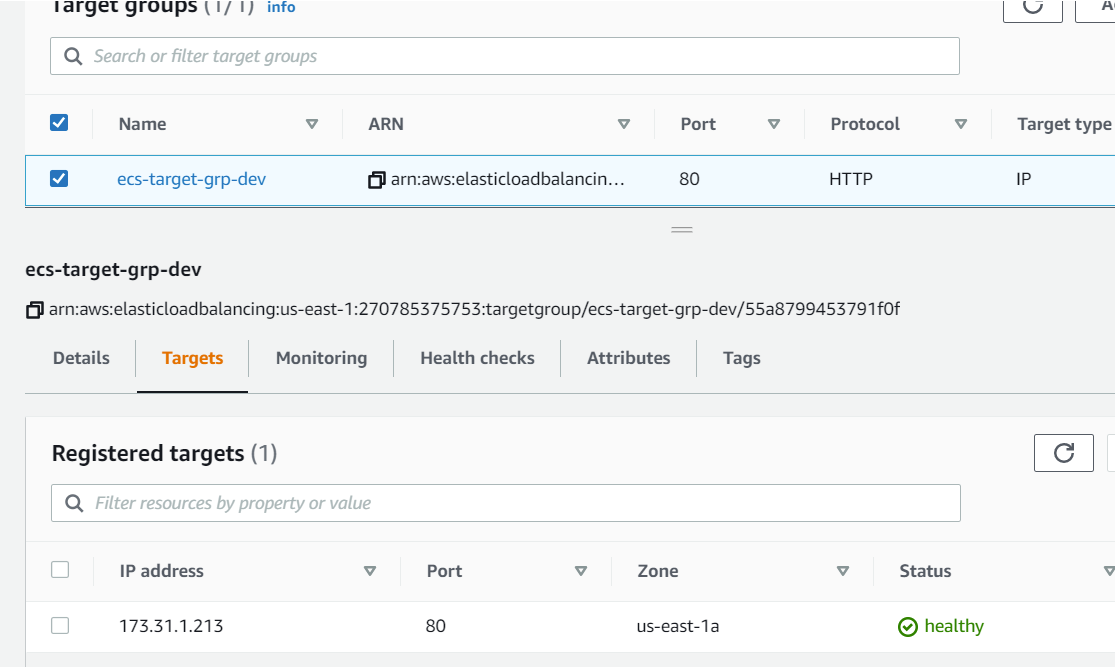
~\aws-ecs-webserver-python3-public\aws-ecs-webserver-python3-public\cloud-formation

1. 
2. Cloud formation template will take care of **resource creation like VPC,PostgreSQL RDS , Task, ECS Cluster, ECR empty repository , ECS Service with ASG , Application Load Balancer**
3. Log on to your AWS Account with us-east-1 (N Viriginia) region
4. 
5. Go to Cloud Formation
6. 
7. Create Stack
8. 

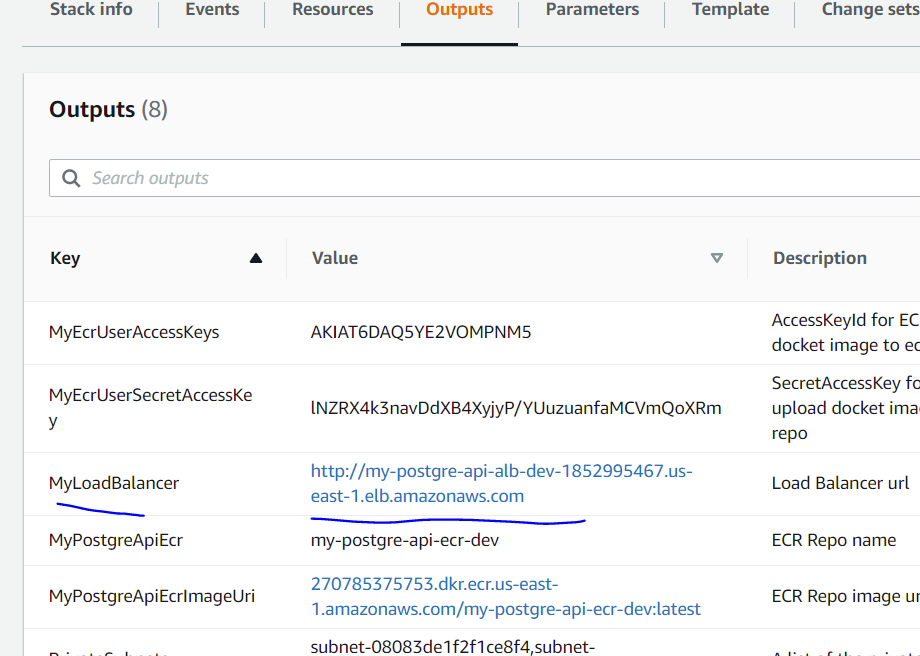
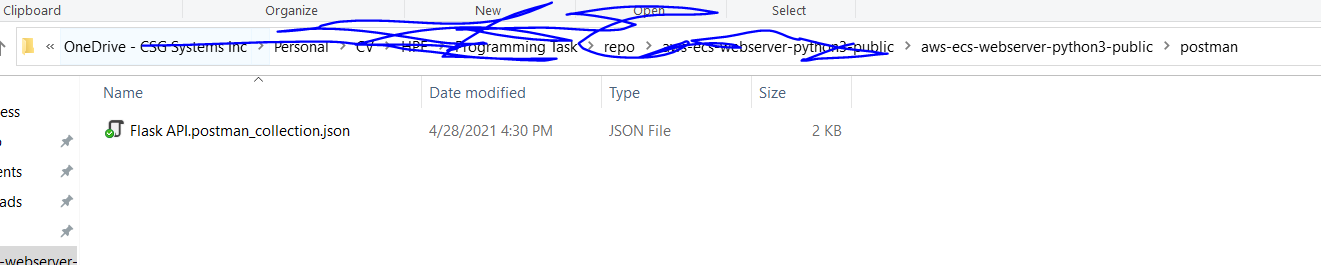
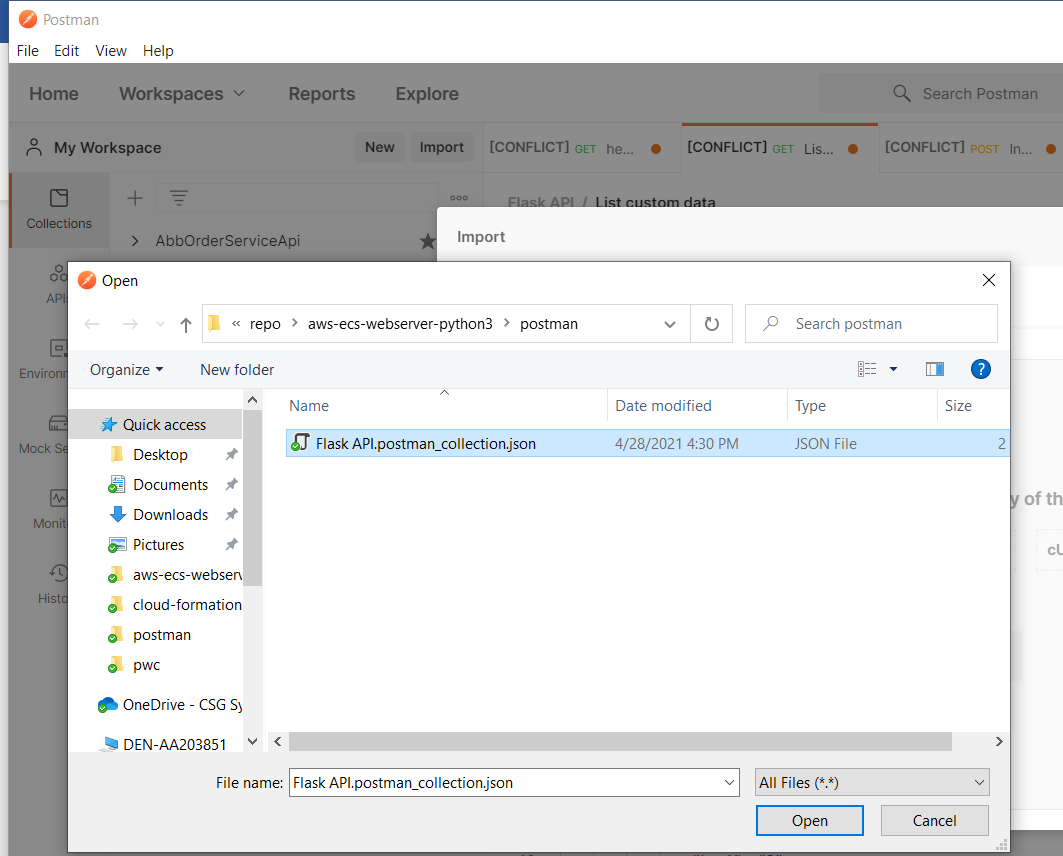
Upload Cloud formation template from downloaded git repo ~\aws-ecs-webserver-python3-public\aws-ecs-webserver-python3-public\cloud-formation

1. 
2. Enter stack name as “**aws-ecs-webserver-python3-public**” or anything you like
3. 
4. Hit Next button
5. Hit Next button again.
6. 
7. Click on Require capabilities [AWS::IAM::AccessKey, AWS::IAM::Role]

Internally Cloud formation is creating IAM user for ECR image upload. With no other access

1. ****
2. Cloud formation template will take **around 15 min to Complete.**
3. 
4. After Cloud Formation is completed go to Outputs tab
5. And Note down **MyEcrUserAccessKeys , MyEcrUserSecretAccessKey , MyLoadBalancer ,** **MyPostgreApiEcr , MyPostgreApiEcrImageUri** output values
6. 
7. Open Windows powershell in run as amin mode and go to git repo folde ~\ aws-ecs-webserver-python3-public\aws-ecs
8. Perform **aws configure** with **MyEcrUserAccessKeys , MyEcrUserSecretAccessKey** received from Cloud formation output variables
9. 
10. Make sure Docker Desktop is running on your laptop . we need to build and upload docker image to ECR
11. Go to AWS ECR
12. 
13. Get the push command from ECR for repo **MyPostgreApiEcr = my-postgre-api-ecr-dev**
14. 
15. Open Windows powershell in run as amin mode and go to git repo folde ~\ aws-ecs-webserver-python3-public\aws-ecs
16. Log on to aws ecr
    1. aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin {your\_aws\_account\_id}.dkr.ecr.us-east-1.amazonaws.com
17. 
18. Build container image
    1. docker build -t my-postgre-api-ecr-dev .
    2. 
19. Tag container image
    1. docker tag my-postgre-api-ecr-dev:latest {your\_aws\_account\_id}.dkr.ecr.us-east-1.amazonaws.com/my-postgre-api-ecr-dev:latest
    2. 
20. Push container impage to aws ecr
    1. docker push your\_aws\_account\_id}.dkr.ecr.us-east-1.amazonaws.com/my-postgre-api-ecr-dev:latest
    2. 
    3. This will take around 5 min to upload 170MB docker image to aws ecr
21. Go to AWS ECS 🡪 Update Service 🡪 **we need to update Number of Tasks from 0 to** 1
    1. 
    2. 
    3. 
22. Go to AWS EC2 🡪 Target Groups 🡪 ecs-target-grp-dev
    1. 
    2. You will see Service Tasks getting recognized soon with “healthy” green color
23. Next steps are in “Run” section of this document

# Run

1. Go to Cloud formation Outputs and get MyLoadBalancer uri which we can use it in postman project provided in git repo
   1. Ex 🡪 http://my-postgre-api-alb-dev-{your\_aws\_account\_id}.us-east-1.elb.amazonaws.com
2. 
3. Go to downloaded git repo ~\postman
4. 
5. Open Postman software 🡪 File 🡪 Import🡪 provide above postman collection json file
6. 

# Clean up

# Things which can be Improved

# Troubleshooting