Day 4: SNS SQS Beanstalk and SSM

Project marks - 70/80

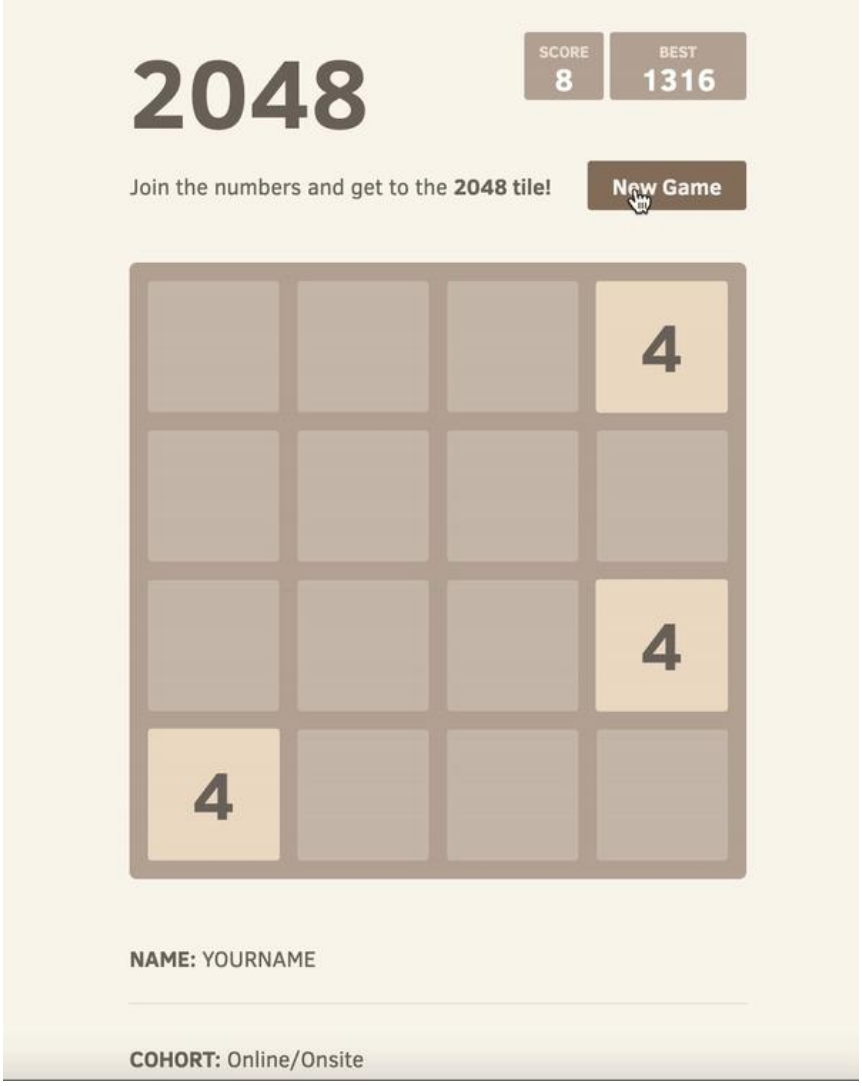
MCQ marks- 10/80

Deadline- 24th 9:00 AM AST

Submission link -

https://docs.google.com/forms/d/e/1FAIpQLSfEkvOP83BB6cOnGLzBT9xz6TFT4K8m4-wxSrF6b66VBezghQ/viewform

Explanation video-



Day 4: SNS SQS Beanstalk and SSM

Description:
Writing and maintaining automation works is an important part of day-to-day activities of DevOps engineers. While AWS Elastic Beanstalk is dedicated to making the process of deploying and managing web applications and services into AWS as easy as possible. At the same time AWS Systems Manager (SSM) gives customers visibility and control of their infrastructure on AWS. Systems Manager provides a unified user interface so customers can view operational data and issues from multiple AWS services and allows them to automate operational tasks across their AWS resources.

In the following questions you have to finish the tasks related to AWS Elastic Beanstalk and AWS SSM. The end objective is to reduce mean time to launching of the infrastructure and resolution for operational issues by providing contextual information and automation runbooks.

Update:
Hello world code -

1. Simple EB. - 25

    1.1. Launch the 2048 application on EB us-east-1 php **- –single instance**. You must change make the following changes in the code.

    ```
    You have to open `index.html` line no. 76 and 80 and
    give your name and your cohort name in the place of
    player's name.
    ```

Day 4: SNS SQS Beanstalk and SSM

Paste screenshot of game playing with your name
Paste game link

1.2. Now You must change make the following changes in the code.

You have to open `style/main.css` line no. 5 and change color `background: #000000;` to `background: #faf8ef;` as blue-green deployment.

And create version-2 version of the app.

Paste new game link

Day 4: SNS SQS Beanstalk and SSM

Paste new game screenshot with your name

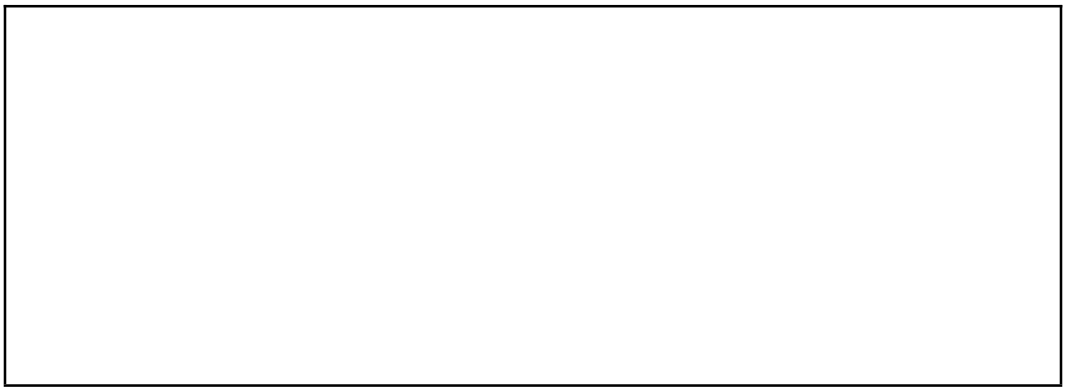
1.3. Now perform the blue green deployment using url swap.

Paste Elastic beanstalk environment page screenshot.
Paste link-1 version-1
Paste link-2 version-2

1.4. Terminate the environment.

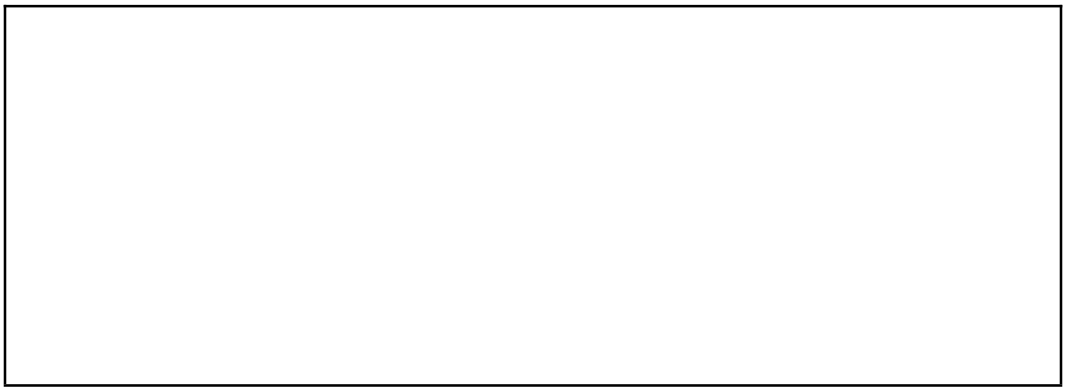
Day 4: SNS SQS Beanstalk and SSM

Paste screenshot of console.



1.5. Delete the application.

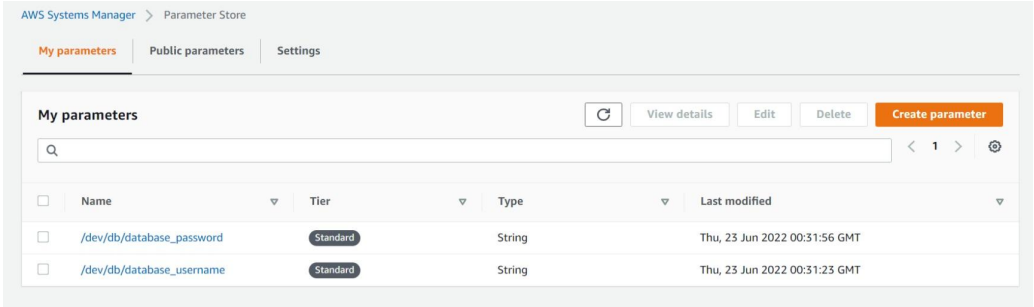
Paste screenshot of console.



2. Use aws ssm parameters to store two credentials. **/dev/db/database_username** and **/dev/db/database_password**. Print that using AWS CLI or python - 10

Paste screenshot of console with both parameters.

Day 4: SNS SQS Beanstalk and SSM



Print that using AWS CLI. Use the commands below in python code or use cli commands.

```
import json

import boto3

client = boto3.client('ssm')

parameterUsername = client.get_parameter(Name='/dev/db/database_username')

parameterPassword = client.get_parameter(Name='/dev/db/database_password',
WithDecryption=True)

print(parameterUsername['Parameter']['Value'])

print(parameterPassword['Parameter']['Value'])
```

Screenshot of the output of both parameter1:

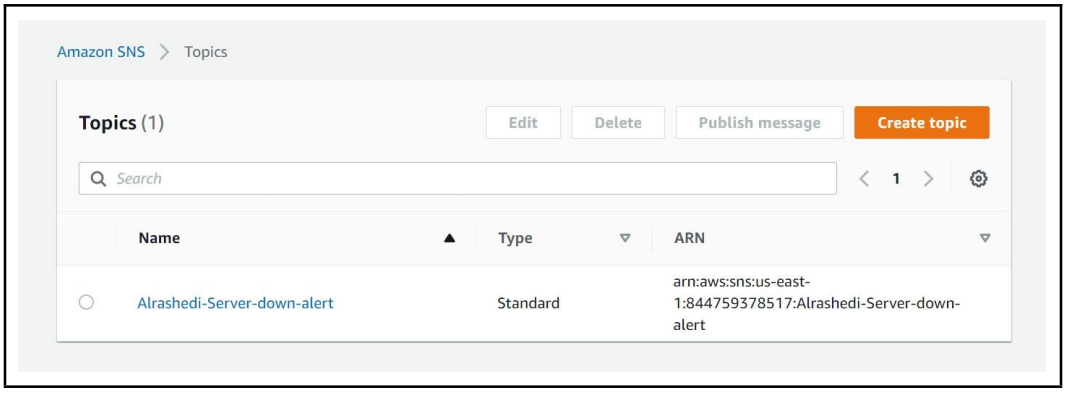
Day 4: SNS SQS Beanstalk and SSM

Screenshot of the output of both parameter2:

3. SNS - 10

3.1. Create a topic “YOURNAME-Server-down-alert” in SNS

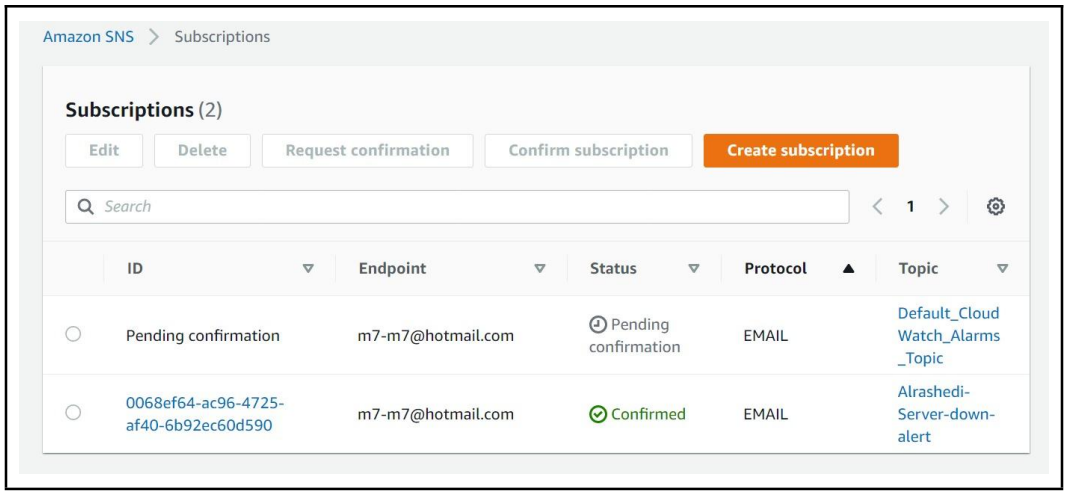
Paste screenshot of console.



3.2. Create a subscription “YOURNAME-report-admin-email” with subscription type EMAIL

Paste screenshot of console.

Day 4: SNS SQS Beanstalk and SSM



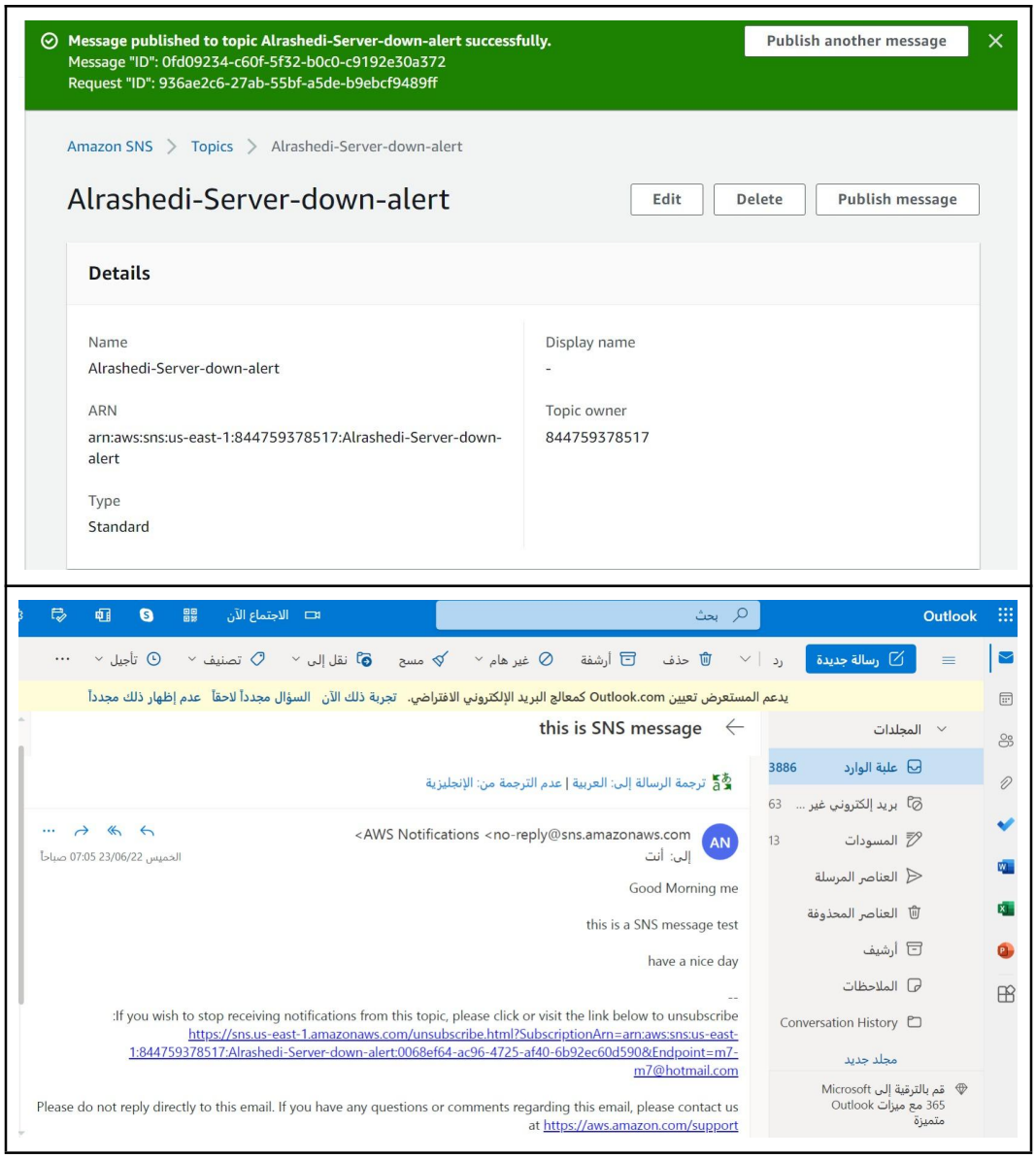
3.3. Confirm the subscription in the email that you receive from AWS.

Paste screenshot of console.

3.4. Publish a sample message to SNS.

Paste screenshot of console.

Day 4: SNS SQS Beanstalk and SSM



4. SQS - 10

4.1. Create an SQS queue with name “YOURNAME-analytics-done”

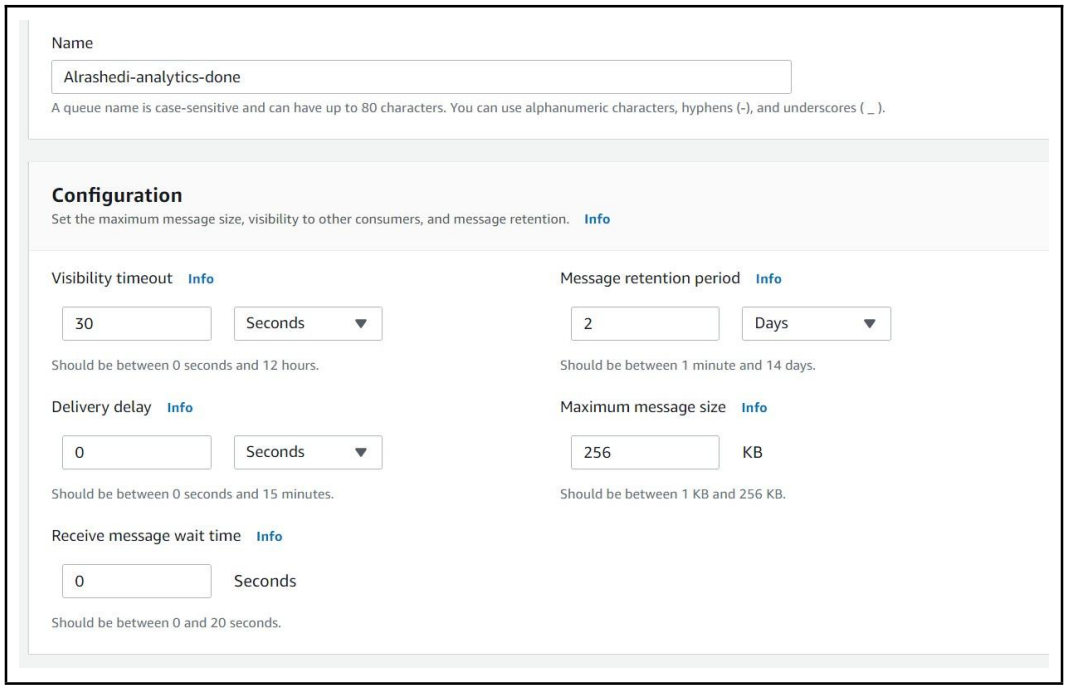
Keep parameters as this

Visibility timeout - 30 sec.

Message retention period - 2 days

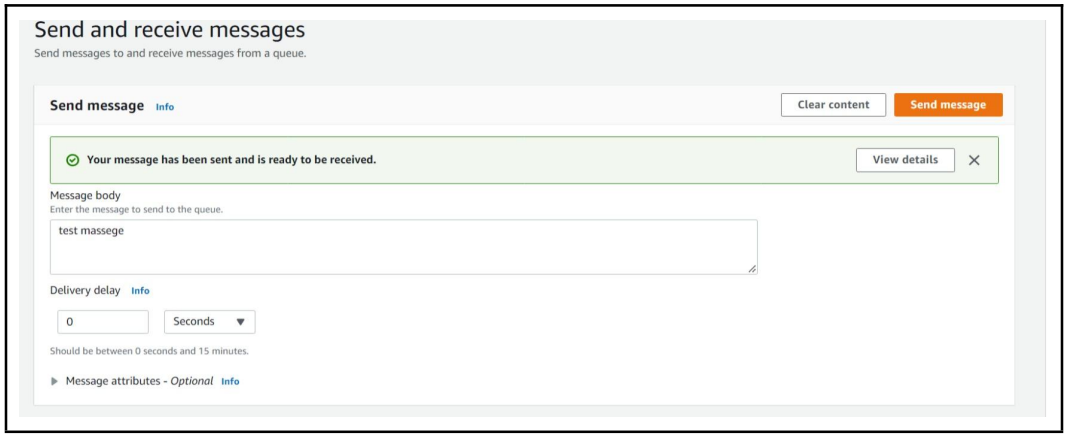
Day 4: SNS SQS Beanstalk and SSM

Paste screenshot of console.

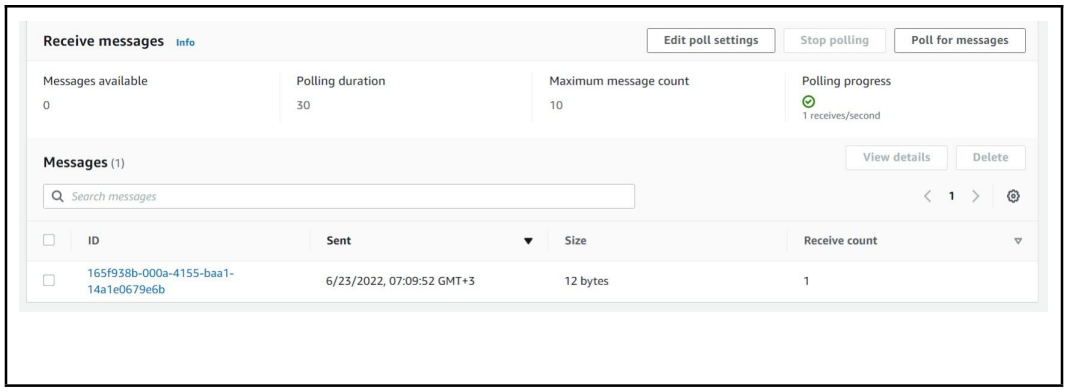


4.2. Send a test message. Paste screenshot of the AWS console.

Paste screenshot of console.



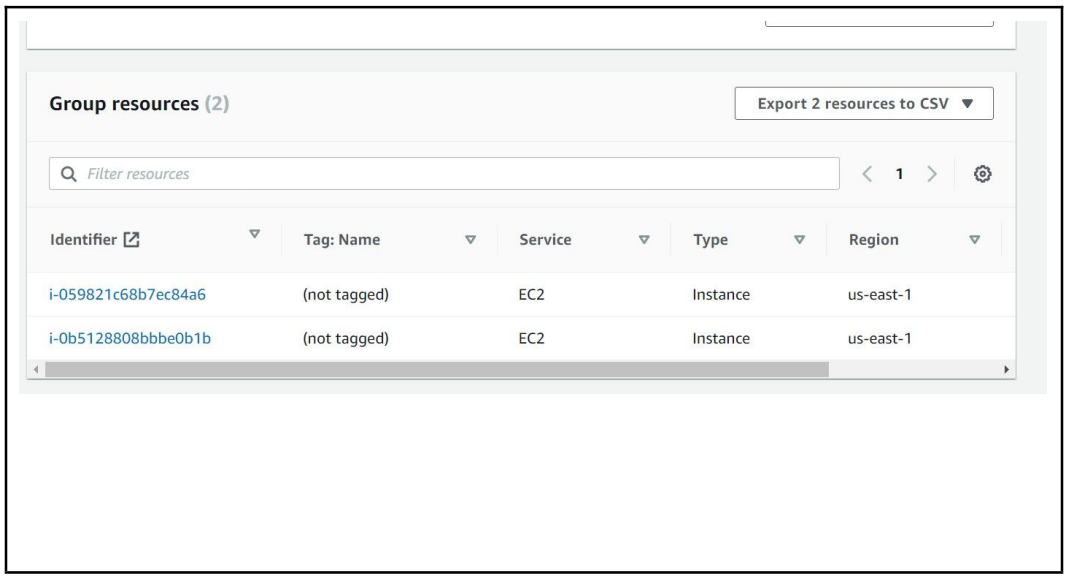
Day 4: SNS SQS Beanstalk and SSM



5. AWS SSM

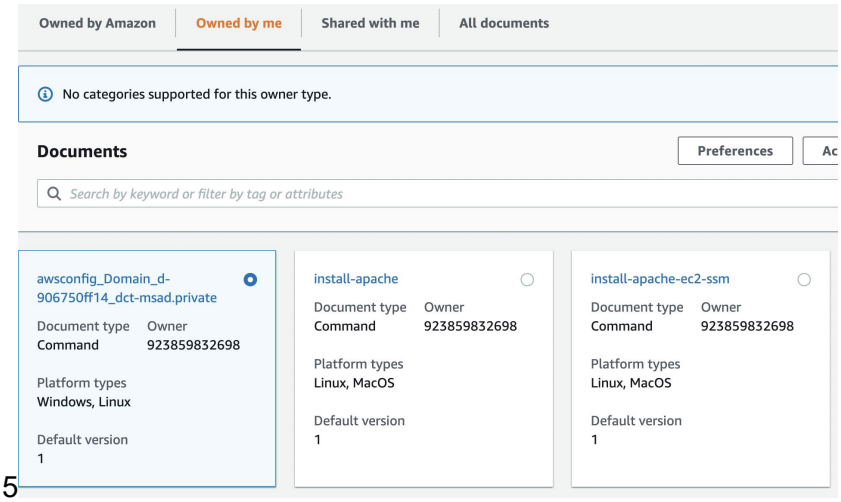
5.1. Create 2 two Amazon linux2 servers and tag one of them as **Dev** and Another **Prod.** Now create resource group **OnPremise and** Add these servers in a **OnPremise** resource group - 10

Paste screenshot of console.



Day 4: SNS SQS Beanstalk and SSM

5.2. Create document to install apache as given in the screenshot below and now install apache in the **OnPremise** resource group. -



5

Paste screenshot of console.

YML document code.

```
---

schemaVersion: '2.2'

description: YAML template which will install Apache on an EC2 instance

parameters:

 InitialWebText:

   type: "String"

   description: "Initial message"

   default: "Welcome to instance "

mainSteps:

- action: aws:runShellScript

 name: configureApache

 inputs:

   runCommand:

   - 'sudo yum update -y'
```

Day 4: SNS SQS Beanstalk and SSM

```
  - 'sudo yum install -y httpd'

  - 'sudo systemctl start httpd.service'

  - 'sudo systemctl enable httpd.service'

  - 'echo "{{InitialWebText}} $(hostname -f) , running Apache, installed
with AWS Systems Manager" > /var/www/html/index.html'
```