

Task6

Script File (.sh or .py)

CloudFormation Deployment Script

```
#!/bin/bash
# Deploy the CloudFormation file
aws cloudformation deploy --region ap-south-1 \
  --template-file ./main.yaml \
  --stack-name ec2forinterview \
  --tags madeFromCLI=yes anotherTagForAllStackResources=yes \
  --capabilities CAPABILITY_NAMED_IAM
```

Explanation of the Deployment Script

This script builds and deploys the CloudFormation template. The template contains the user data for the EC2 instance.

User Data Script

```
#!/bin/bash

dnf update -y
dnf install httpd git python3 python3-pip -y
yum install docker -y

systemctl start docker
systemctl enable docker
```

```
usermod -aG docker $USER
```

```
mkdir -p /usr/local/lib/docker/cli-plugins
```

```
curl -SL
```

```
https://github.com/docker/compose/releases/latest/download/docker-compose-linux-x86_64 -o /usr/local/lib/docker/cli-plugins/docker-compose
```

```
chmod +x /usr/local/lib/docker/cli-plugins/docker-compose
```

```
cd /home/ec2-user
```

```
git clone https://github.com/AryanGitHub/a-very-simple-webapp-for-assignment.git 2> error.log
```

```
cd a-very-simple-webapp-for-assignment
```

```
bash bash.sh 2> error_bash.log
```

Explanation of the User Data Script

This script performs the following actions:

1. System Update and Package Installation:

- Updates the system and installs `httpd`, `git`, `python3`, `python3-pip`, and `docker`.

2. Docker Setup:

- Starts and enables the Docker service.
- Adds the current user to the Docker group to allow running Docker commands without `sudo`.

3. Docker Compose Installation:

- Downloads and installs Docker Compose.

4. Clone the Web Application Repository:

- Clones the specified GitHub repository containing the web application.

5. Run the Web Application:

- Navigates to the cloned repository and executes the `bash.sh` script.

Bash Script in the Web Application Folder

```
#!/bin/bash
```

```
python -m venv .venv  
source .venv/bin/activate  
pip install -r ./requirements.txt  
pip install prometheus-fastapi-instrumentator  
uvicorn main:app --host 0.0.0.0 --port 80 --reload
```

Explanation of the Bash Script

This script performs the following actions:

- 1. Create a Python Virtual Environment:**

- Sets up a virtual environment in the current directory.

- 2. Activate the Virtual Environment:**

- Activates the virtual environment to isolate package installations.

- 3. Install Required Packages:**

- Installs the packages listed in `requirements.txt` and the `prometheus-fastapi-instrumentator` package.

- 4. Run the Web Application:**

- Starts the web application using `uvicorn`, listening on all interfaces at port 80 with auto-reload enabled.