Assignment #02

Due Date

Section	Due Date	Grading Deadline
Saturday & Online	06:00pm on 06/14/2019	06:00pm on 06/21/2019

Getting Help

Info

Ask all your questions on Piazza. Assign **assignment2** tag to your posts.

Assignment Weightage

Individual Assignment Weightage on Course Grade of this assignment is **10%**.

Objectives

Amazon Web Services Identity & Access Management (IAM) Setup

Configure AWS Identity and Access Management (IAM) [https://aws.amazon.com/iam/] for your account.

Create Groups

Danger

No one except you should be able to modify resources in your environment.

1. Create following groups
   a. csye6225-su19-team
   b. csye6225-su19-staff
2. These groups should have read-only access to AWS services and resources in your account. Search for policy pre-defined read-only policy **arn:aws:iam::aws:policy/ReadOnlyAccess** [https://console.aws.amazon.com/iam/home?region=us-east-1#/policies/arn:aws:iam::aws:policy/ReadOnlyAccess$serviceLevelSummary] and attach it to the groups.

Create Users

1. Create users for each of your team member and teaching assistants.
2. Use first name as the username. Do not configure credentials for the users.
3. Leave the default setting **Autogenerated password** checked and copy the generated password. **Autogenerated passwords are not emailed out by AWS. You need to manually send the email with password.**

AWS Billing Alarm

Create billing alarm so that you are notified if/when AWS usage exceeds $10/month.

Install and Setup AWS Command Line Interface

You must install and setup AWS Command Line Interface (CLI) on your development virtual machine. See Install the AWS Command Line Interface on Linux [https://docs.aws.amazon.com/cli/latest/userguide/awscli-install-linux.html] for detailed instructions.

Infrastructure as Code

We are going to start setting up our AWS infrastructure. This assignment will focus on setting up our networking resources such as Virtual Private Cloud (VPC), Internet Gateway, Route Table and Routes. We will use both AWS Command Line Interface and AWS CloudFormation for infrastruture setup and tear down.

> **Naming Resources**
>
> Wherever supported, you must provide unique names to the resources. You may be asked to create multiple networks as part of assignment demo in the same account.

Infrastructure as Code with AWS Command Line Interface

Here what you need to do for networking infrastructure setup:

1. Create a Virtual Private Cloud (VPC) [https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-ec2-vpc.html].
2. Create subnets [https://docs.aws.amazon.com/vpc/latest/userguide/working-with-vpcs.html#AddaSubnet] in your VPC. You must create `3` subnets, each in different availability zone in the same region under same VPC.
3. Create Internet Gateway [https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-ec2-internetgateway.html] resource.
4. Attach the Internet Gateway [https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-ec2-vpc-gateway-attachment.html] to the created VPC.
5. Create a public Route Table [https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-ec2-route-table.html]. Attach all subnets created above to the route table.

6. Create a public route [https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-ec2-route.html] in the public route table created above with destination CIDR block `0.0.0.0/0` and internet gateway creted above as the target.

For this objective, you must complete following tasks:

1. Install and setup AWS command line interface.
2. Create shell script `csye6225-aws-networking-setup.sh` to create and configure required networking resources using AWS CLI. Script should take all required values as parameter and should not contain environment specific hardcoded values.
3. Create shell script `csye6225-aws-networking-teardown.sh` to delete networking resources using AWS CLI. Script should take all required values as parameter and should not contain environment specific hardcoded values.
4. Should one of the resource creation fail, your script should print out proper error message and exit gracefully. Your scripts are not required to rollback already created resources.
5. Resources are only created if the previous command is successful.

Infrastructure as Code with AWS CloudFormation

Here what you need to do for infrastructure setup:

1. Create a Virtual Private Cloud (VPC) [https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-ec2-vpc.html].
2. Create subnets [https://docs.aws.amazon.com/vpc/latest/userguide/working-with-vpcs.html#AddaSubnet] in your VPC. You must create `3` subnets, each in different availability zone in the same region under same VPC.
3. Create Internet Gateway [https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-ec2-internetgateway.html] resource.

4. Attach the Internet Gateway [https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-ec2-vpc-gateway-attachment.html] to the VPC.
5. Create a public Route Table [https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-ec2-route-table.html]. Attach all subnets created above to the route table.
6. Create a public route [https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-ec2-route.html] in the route table with destination CIDR block `0.0.0.0/0` and the internet gateway as the target.

For this objective, you must complete following tasks:

1. Install and setup AWS command line interface.
2. Create shell script `csye6225-aws-cf-create-stack.sh` to create and configure required networking resources using AWS CloudFormation. Script should wait until the stack is terminated and print message on script if it was success or failure.Your shell script may assume that the CloudFormation template is in the same directory as your script.
3. Create shell script `csye6225-aws-cf-terminate-stack.sh` to delete CloudFormation stack. Script should wait until the stack is terminated and print message on script if it was success or failure. Your shell script may assume that the CloudFormation template is in the same directory as your script.
4. Create CloudFormation template `csye6225-cf-networking.json` or `csye6225-cf-networking.yaml` that can be used to setup required networking resources. Both scripts must accept `stack name` among other things as parameter.

Folder Structure for Infrastructure Code

1. Create a folder structure similar to the one shown below. You can refer to template repository [https://github.com/tejasparikh/csye6225-summer2019-template].

2. Your shell scripts for *Infrastructure as Code with AWS Command Line Interface* must be stored in `scripts` directory and CloudFormation templates and shell script created in *Infrastructure as Code with AWS CloudFormation* must be in `cloudformation` directory.
3. `README.md` in each directory must document instructions on running the scripts contained in the folder.

```
.
├── infrastructure
│   ├── aws
│   │   ├── cloudformation
│   │   │   ├── README.md
│   │   │   ├── csye6225-aws-cf-create-stack.sh
│   │   │   ├── csye6225-aws-cf-terminate-stack.sh
│   │   │   ├── csye6225-cf-networking.json
│   │   ├── codedeploy
│   │   │   └── README.md
│   │   └── scripts
│   │       └── README.md
```

Web Application

RESTful API Endpoints To Be Implemented

HTTP Method	Endpoint	Authenticated Endpoint	Description
GET	`/book/{id}/image/{idImage}`	Yes	Get image for the book
DELETE	`/book/{id}/image/{idImage}`	Yes	Delete image for the book
PUT	`/book/{id}/image/{idImage}`	Yes	Update image for the book
POST	`` `/book/{id}/image ``	Yes	Add image for the book

API Specifications

API specifications [https://app.swaggerhub.com/apis-docs/csye6225/csye6225-summer2019/2.0.0]

User Stories

1. As a user, I want to add, update or delete images (cover page) for books. I should be authenticated & authorized to be able to perform these operations.
    a. As a user, I want to add a images for the book. Only `.png` , `.jpg` ,, and `.jpeg` formats are supported.
    b. As a user, I want to update book image. Updating images should replace existing image.
    c. As a user, I want to delete image for the book.
2. Only one image can be uploaded per book.
3. ~~As a user, I expect image to be stored in Amazon S3 bucket when the application is running on cloud (when running in EC2 instance).~~ **you are not implementing this feature in this assignment**.

4. As a user, I expect image to be stored locally in some directory when running application is running locally on developer's machine.
5. Metadata about book image must be stored in RDBMS such as MySQL or Postgres.
    a. When the application is running locally, it must use a local database instance.
    b. ~~When the application is running on cloud, it must use the AWS RDS service.~~ **you are not implementing this feature in this assignment**.

Documentation

AWS Command Line Interface

- Install AWS CLI [http://docs.aws.amazon.com/cli/latest/userguide/installing.html]
- Configure AWS CLI [http://docs.aws.amazon.com/cli/latest/userguide/cli-chap-getting-started.html]

AWS CloudFormation

- AWS CloudFormation [https://aws.amazon.com/cloudformation/]
- AWS CloudFormation Intrinsic Function Reference [http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/intrinsic-function-reference.html]
- Virtual Private Cloud [https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-ec2-vpc.html]
- Internet Gateway [https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-ec2-internetgateway.html]

- Route Table [https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-ec2-route-table.html]
- Route [https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-ec2-route.html]
- Attaches a gateway to a VPC [https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-ec2-vpc-gateway-attachment.html]

Submission

> **Danger**
>
> Assignment will be considered late if commits are made to `master` and `feature` branch after due date.

1. All work for this assignment must be done on **assignment2** feature branch and merged to master when you are dev complete.
2. All team member's feature and master branches must be in-sync.

Grading Guidelines

> **Warning**
>
> Following guidelines are for information only. They are subject to change at the discretion of the instructor and TA.

Previous Assignment Objectives

- TAs must verify that students have resolved all open issues from previous assignment(s).

IAM (10%)

- Check if all users (team and TAs) are created and assigned to correct group.
- Verify users have console access and not programmatic access. For this check the email students have sent to you. Did the email contain password or credentials for programmatic access or just username?
- Check the policy attached to the group is **arn:aws:iam::aws:policy/ReadOnlyAccess** [https://console.aws.amazon.com/iam/home?region=us-east-1#/policies/arn:aws:iam::aws:policy/ReadOnlyAccess$serviceLevelSummary].
- Verify that AWS billing alarm is setup. Dollar amount does not matter.
- Verify that you (TA) have received both username and password from student for IAM console. Login and verify that you can access their AWS account and your account can only READ. Try deleting their networking stack and it should not allow you to do that.

Infrastructure as Code with AWS CloudFormation & Command Line Interface (50%)

Info

All demo's must be performed by downloading code from student's GitHub repository's master branch through browser.

- Students should demo creation of CloudFormation stack using `csye6225-aws-cf-create-stack.sh` shell script.
- Students should demo termination of CloudFormation stack using `csye6225-aws-cf-terminate-stack.sh` shell script.
- Students should demo `csye6225-aws-networking-setup.sh` shell script.
- Students should demo `csye6225-aws-networking-teardown.sh` shell script.

- Verify that CloudFormation templates and shell scripts do not contain any hardcoded values.
- Try scripts with invalid parameters.
- **Create multiple VPCs using CLI and CloudFormation without any conflicts.**

Git Forking Workflow (15%)

- All team members must use the Github forking workflow and their repositories (master branch which must include code for this assignment) must be in-sync. Check this by asking students to create pull request between their `master` branch and their scrum master's `master` branch. There should be no changes. Verify that all assignment changes are in `master` branch.
- Students must show pull requests raised for their code changes contribution. A student who has not raised any pull request for the assignment gets `0` points for the whole assignment.
- Added TAs and instructor as collaborator to the GitHub repository.
- Verify that students have README.md file in their git repository and it contains instructions on how to build, test and deploy their application including any pre-requisites for programming language, frameworks and third-party libraries.
- Verify that dev environment is not setup in `Downloads` folder.
- Git repositories should be cloned locally using `git/ssh` protocol and not `https`. Verify this by running `git remote -v` command in the cloned repository in the VM.
- Ask students to perform `git pull` from scrum master's repo and run the `git status` command. This must be done from terminal.

Git Repository Content Check

- Check the repository for any AWS credentials
- Check the repository for any IDE specific files. IDE configuration files must not be in repository. Verify their `.gitignore` configuration.

- Check the repository for build artificats such as `.class`, `.jar`, `.war` files and `build` directory. None of these should be checked into the repostiory.
- Check for dependencies. Dependencies from Maven repository should not be committed to the git repository.

Web Application (25%)

1. Demo all API implemented. Check the output is in JSON format and correct HTTP response code is returned for successful API calls.
2. Demo error handling in APIs and validate that correct error HTTP response code is returned as defined in user stories above.
3. Students should demo their application running with **default** or **dev** application profile
   a. Book images must be stored on local disk.
   b. Local RDBMS database instance must be used