Install and Configure AWS Command Line Interface

* Install and configure AWS Command Line Interface (CLI) on your development machine (laptop). See [Install the AWS Command Line Interface on Linux](https://docs.aws.amazon.com/cli/latest/userguide/awscli-install-linux.html) for detailed instructions.
* Create dev profile for your dev AWS account and demo profile for your production AWS account. Do not set up a default profile.
* Both dev and demo AWS CLI profiles should be set to use the us-east-1 region or the region closest to you.

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 Tables, and Routes. We will use AWS CloudFormation for infrastructure setup and tear down.

**Note: Wherever is supported, you must provide unique names to the resources. You may be asked to create multiple networks in the same account during the demo.**

Create & Setup GitHub Repository for AWS Infrastructure

1. [Create](https://help.github.com/articles/create-a-repo/) a new private GitHub repository in the GitHub organization you created.
2. GitHub repository name must be infrastructure.
3. Grant TAs access to your GitHub repository.
4. Update README.md in your repository with instructions for setting up your infrastructure using CloudFormation.
5. Fork the GitHub repository in your namespace. You will do all development work on your fork.
6. All CloudFormation templates should be in this repository.
7. Add appropriate .gitignore to your repository. A collection of useful .gitignore templates can be found [here](https://github.com/github/gitignore).

AWS Networking Setup

Here is what you need to do for networking infrastructure setup:

1. Create [Virtual Private Cloud (VPC)](https://docs.aws.amazon.com/vpc/latest/userguide/what-is-amazon-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 a different availability zone in the same region in the same VPC.
3. Create an [Internet Gateway](https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Internet_Gateway.html) resource and attach the Internet Gateway to the VPC.
4. Create a public [route table](https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Route_Tables.html). Attach all subnets created to the route table.
5. Create a public route in the public route table created above with destination CIDR block 0.0.0.0/0 and internet gateway created above as the target.

Infrastructure as Code with CloudFormation

For this objective, you must complete the following tasks:

1. Install and set up AWS command-line interface.
2. Create CloudFormation template *csye6225-infra.json*or *csye6225-infra.yml* that can be used to set up required networking resources.
3. Values should not be hardcoded in your CloudFormation template.
4. You must be able to use the same CloudFormation template in the same AWS account and region to create multiple VPCs including all of its resources (listed in the “AWS Networking Setup” section) such as subnets, internet gateway, route table, etc.

Documentation

AWS Command Line Interface

* [AWS Command Line Interface](https://aws.amazon.com/cli/)
  + [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)
* [Tools for Amazon Web Services](https://aws.amazon.com/tools/)

CloudFormation

* [AWS CloudFormation](https://aws.amazon.com/cloudformation/)
* [CloudFormation Intrinsic Function References](https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/intrinsic-function-reference.html)
* [AWS CloudFormation Templates](https://aws.amazon.com/cloudformation/resources/templates/)
* [AWS CloudFormation Documentation](https://docs.aws.amazon.com/cloudformation/index.html)
* [AWS resource and property types reference](https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-template-resource-type-ref.html)

Submission

The assignment will be considered late if commits are made to the **main** branch after the due date.

1. All work for this assignment must be done on the feature branch in your fork and merged to main when you are dev complete.
2. The feature and main branches must be in sync.
3. Submit your code from **all** repositories in this assignment. **Read the instructions carefully to create your zip file.**
   1. Create a folder with the naming convention **firstname\_lastname\_neuid**
   2. In the **firstname\_lastname\_neuid**clone all of your GitHub (organization) repositories with the **git clone**command. It is important that you clone the repos so that your commit history and branches are preserved.
   3. Once you have cloned all of your repositories, you will create a zip of the **firstname\_lastname\_neuid\_a\_##** directory. The zip file should be **firstname\_lastname\_neuid\_a\_##.zip** where **##** is the assignment number.
   4. Now unzip the zip file in some other directory and confirm the content of the zip files.
   5. Upload the Zip to this assignment.
4. You are allowed to resubmit. If you think there may be an issue with the ZIP file, feel free to submit it again. Only the latest will be used grading.

Grading Guidelines

The following guidelines are for information only. They are subject to change at the discretion of the instructor and TA.

10% penalty will be imposed if the application crashes due to unhandled exceptions/errors.

Previous Assignment Objectives

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

Git Repository Content Check (10% Penalty)

* Check the repository for any IDE-specific files. IDE configuration files must not be in the repository.
  + Verify their .gitignore configuration.
* Check the repository for build artifacts such as .class, .jar, .war files and build, node\_modules directory. None of these should be checked into the repository.
* Check for dependencies. Dependencies from the Maven repository or npm should not be committed to the git repository.

Pre-requisite for Demo (5% Penalty)

* Delete all VPCs you may have created in your demo AWS account.

Infrastructure as Code with CloudFormation (100%)

**All demos must be performed by downloading code from the student’s GitHub repository’s main branch through the browser.**

* Students should demo the creation of networking resources using AWS CLI & CloudFormation template.
* Students should demo the cleanup of networking resources using AWS CLI & CloudFormation template.
* Verify that the CloudFormation template does not contain any hardcoded values.
* **Create multiple VPCs (and resources) without any conflicts in the same AWS account & same region.**
* **Create another VPCs (and resources) without any conflicts in different AWS Region.**