Capstone Project #3
Cloud Application

This capstone project will provide an opportunity to design and build a multi-faceted solution that combines multiple technologies covered to date. Intended to be more substantial than a lab, the capstone aims to simulate the type of work more closely that you might encounter as part of a development team working on larger software engineering projects. The capstone project is considered "open book", self-paced, and to be completed on your own time. You may refer to the relevant videos and reference material (as well as use internet searches) as part of unit completion. Additionally, learners are encouraged to work as teams to complete. Having said that, each learner is still expected to submit an individual solution for the project to ensure that all team members are gaining necessary competency in the target technologies. Prior to beginning development, each learner is expected to submit a set of bullets outlining intended approach including a "checklist" for the set and order of tasks planned for implementation.

You will be using AWS CloudFormation to build the following set of resources in AWS:

EC2 and NodeJS/Express

- Using CloudFormation, create a new stack template including an EC2 instance with elastic IP address (start with the template provided by AWS and update according to instructions outlined below)
- For "InstanceType" parameter, limit allowed values to only t1 and t2 sizes; make sure you update the mappings as well
- Add a new parameter called "DepartmentTagValue" that will be a string between 10 and 20 characters, inclusive
- Add a new parameter called "EnvironmentTagValue" that will be a string allowing selection from a dropdown that includes "Development", "QA", and "Production"; set the default to "Development"
- Include a tag with key of "Department" on the EC2 instance and use the "DepartmentTagValue" parameter for tag value
- Include a tag with key of "Environment" on the EC2 instance and use the "EnvironmentTagValue" parameter for tag value
- Save the CloudFormation template (**you will submit this as part of your solution**)
- Use the CloudFormation template to create a new "t2.micro" instance; for SSH IP address use your own public IP address to limit and secure remote connectivity
- Confirm ability to connect over SSH via putty (or similar) to the EC2 instance (**capture a screen shot of that connection for submission as part of your solution**)
- While SSH'ed into your EC2 instance, install node and git
- Use git to pull down the repo at https://github.com/KernelGamut32/capstone3-resources onto your EC2 instance file system
- Launch the node application using the server.js file
- This app includes a single route of "/" that returns the string "Hello from NodeJS on EC2" using port 43000

- Make the necessary updates to the security group to allow inbound traffic for HTTP to that port from your public IP address
- Verify connectivity by accessing the "/" route remotely in a browser from your machine and confirm return of the expected text (**capture a screen shot of the browser for submission as part of your solution**)
- OPTIONAL CHALLENGE: Leverage a combination of UserData and/or cfn-init/AWS::CloudFormation::Init to automatically install NodeJS, install Git, and pull down the GitHub repo to your EC2 instance on initial provision
- When complete, post the following to a GitHub repo and send the link for review:
    - CloudFormation template
    - Screen shot demonstrating SSH connectivity to the EC2 instance
    - Screen shot demonstrating HTTP connectivity to the NodeJS application running in EC2
- Make sure you delete all AWS resources after completion to avoid incurring charges