Assignment #03

Due Date

Section	Due Date	Grading Deadline
Saturday & Online	09:00pm on 06/21/2019	09:00pm on 06/28/2019

Getting Help

Info

Ask all your questions on Piazza. Assign **assignment3** tag to your posts.

Assignment Weightage

Individual Assignment Weightage on Course Grade of this assignment is **5%**.

Objectives

Domain Name System (DNS) Setup

Register Domain Name

Namecheap Domains

> Only **.me** TLD domains are free for students.

1. Register a domain name with Namecheap [https://www.namecheap.com/domains/registration.aspx] as your registrar. You will be able to get a domain for free with Github Student Developer pack [https://education.github.com/pack].
2. Domain name must be of format `csye6225-su19-husky_id.me` where `husky_id` is your Northeastern husky id without the dot in it. For example, if you husky email id is `doe.j@husky.neu.edu`, your domain will be `csye6225-su19-doej.me`.

Configuring Amazon Route 53 For DNS Service

1. Create a public hosted zone [http://docs.aws.amazon.com/Route53/latest/DeveloperGuide/CreatingHostedZone.html] in Amazon Route 53 [https://aws.amazon.com/route53/].
2. Configure Namecheap to use custom nameservers [https://www.namecheap.com/support/knowledgebase/article.aspx/767/10/how-can-i-change-the-nameservers-for-my-domain] provided by Amazon Route 53 [http://docs.aws.amazon.com/Route53/latest/DeveloperGuide/GetInfoAboutHostedZone.html].
3. Create a type TXT [http://docs.aws.amazon.com/Route53/latest/DeveloperGuide/ResourceRecordTypes.html#TXTFormat] record for your domain with **TTL** [http://docs.aws.amazon.com/Route53/latest/DeveloperGuide/resource-record-sets-values-basic.html#rrsets-values-basic-ttl] of `10 seconds`. Type `TXT` record should contain the text value `"csye6225-spring2019"`.

Setup GitHub Repository for AMIs

1. Create a GitHub repository for assignments [https://help.GitHub.com/articles/create-a-repo/]. This must be a private repository that only your team and TAs can access. Make sure to create empty

repository. Just like we learned in the lab, one member should create the repo and other's should fork this repo.

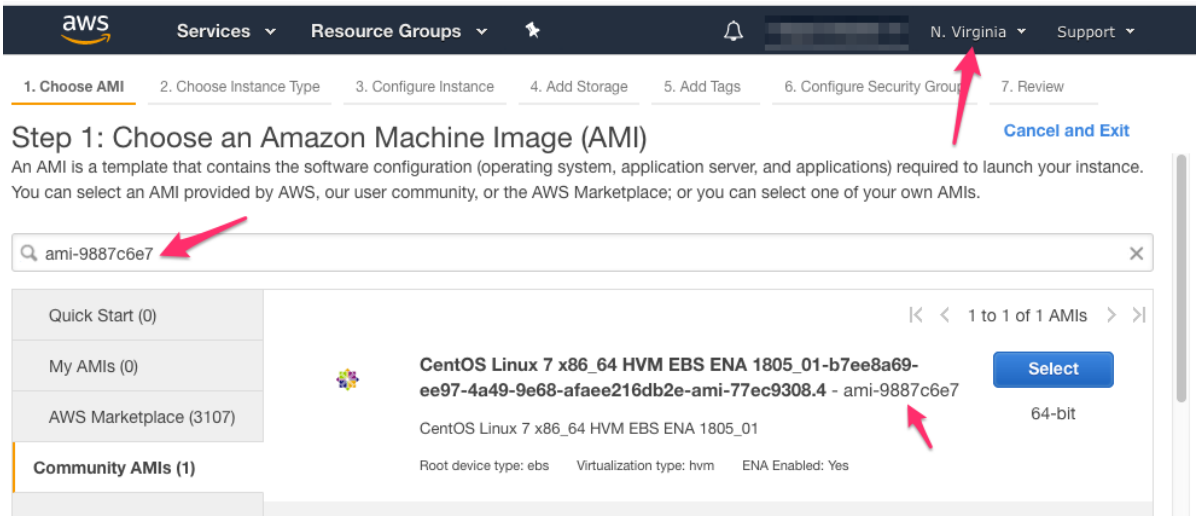
2. GitHub repository name must be **csye6225-su19-ami**.
3. Add all TAs to your GitHub repository as collaborators. TAs emails and GitHub IDs can be found on home page.
4. Update **README.md** in your repository with team member information.

Subscribe to CentOS in AWS Marketplace

Before you can use CentOS AMIs, you may have to manually subscribe for it at https://aws.amazon.com/marketplace/pp/B00O7WM7QW [https://aws.amazon.com/marketplace/pp/B00O7WM7QW].

Building Custom AMI using Packer

Create a AMI using packer. Use **CentOS Linux 7 (AMI ID ami-9887c6e7)** as your source image. This AMI should be private so that only you can deploy EC2 instances from it.



Documentation

Packer & AWS AMIs

- Amazon Machine Images (AMI) [https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AMIs.html]
- HashiCorp Packer [https://www.packer.io/]

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 **assignment3** 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).

Domain Name Setup (25%)

- Check if student's name meets the naming requirement.

- Check if students have configured `Name Server` in Namecheap to use Amazon Route 53. Command to use is `dig NS DOMAIN_NAME`.
- Query for `TXT` record for student's domain and see if the value returned is `"csye6225-spring2019"`. Use the command `dig TXT DOMAIN_NAME` to verify.

Building AMIs (60%)

Students must demo following:

1. Build AMI using packer.
2. Launch EC2 instance from the AMI
3. Deploy application on the EC2 instance. Students can do manual install of database, schema, and application configuration.
4. Verify application APIs can be accessed from the IP address of EC2 instance. Security group may be manually configured to permit access to the application.

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.