# HW 2 - Configuration Management

Automatically configuring a computing environment will be one of the most valuable skills for you to master in your career. You have gained some experience using ansible to automatically run configuration tasks on a virtual machines.

You will start with a starter code base and modify it to fulfill the homework criteria.

## Setup

Please do the following before you start the homework.

### Prepare your GitHub Repo.

Sign into NCSU's GitHub.

1. Create a *private* repo called HW2-DevOps.
2. Go to Settings, Collaborators and Teams, and add the TAs and instructor as a collaborator (using their unity id).

Samim Mirhosseini Ghamsa smirhos@ncsu.edu, Jeremiah Percy Dsouza jdsouza@ncsu.edu, Christopher Parnin cjparnin@ncsu.edu

**Do not create any content, yet**

### Clone and set-url

Clone this repo:

```
git clone https://github.com/CSC-DevOps/CM-Template
```

Create a private `HW2-DevOps` GitHub repo, change the remote url of the repo you just cloned, and push:

```
git remote set-url origin https://github.ncsu.edu/<unityid>/HW2-DevOps
git push -u origin master
```

Read the instructions on CM-Template for details about the two commands provided:

- `cm setup`
- `cm playbook`

# Basic Requirements

In this homework assignment, you will master these skills by completing the following tasks:

**Using ansible**, be able to automatically configure a server running mattermost.

Perform the following tasks listed in the installation guide for mattermost.

- Installing Ubuntu Server 18.04 LTS (This should be mostly done, you already have image! But can still run update/upgrade).
- Installing MySQL Database Server (5.7.x is recommended)
- Installing Mattermost Server

Your goal is to translate those instructions into ansible playbooks and roles.

## Demonstrating mattermost server

- Finalize configuring the mattermost server by browsing http://192.168.33.80:8065.
- Create a team and users.
- Demonstrate you can actually use mattermost by posting some messages.

## Constraints

Follow the following constraints when creating your server:

- Use best practices (use modules to do work, avoid shell/commands, be idempotent, use your own roles when sensible, manage secrets)
- Everything is setup automatically. No manual steps.
- **Limited use ansible galaxy roles**: You are limited to using galaxy roles for setting up mysql. Note you will need to update your ansible server setup script to install galaxy roles on your ansible-srv.

# Extra Requirements

You can also extend your implementation to meet the following extra requirements for more credit:

- Automate the creation of teams and other mattermost server configuration using the mattermost CLI (5 points)
- Configure the ability to send email notifications (5 points).
- Complete the section "Configuring NGINX as a proxy for Mattermost Server" in mattermost instalation instructions (10 points).
- Complete the section "Configuring NGINX with SSL and HTTP/2". Note you can setup a local hosts file for enabling temporary testing of your ssl configuration. (5 points)

Reminder that limited technical assistance will be provided by teaching staff when attempting the extra requirements. Make sure you have completed all the basic requirements before attempting any extra requirements.

## Screencast (10)

Create a screencast of your assignment:

- Demonstrate running your code to setup your environment and running your customization and post-configuration steps (`cm setup`), and running your mattermost playbook (`cm playbook cm/mattermost.yml cm/inventory.ini`). Demonstrate your mattermost server running on your browser. Demonstrate any extra requirements fulfilled. Note some extra requirements supercede the basic requirements.

For guidelines, software, and recommendations see Screencasts.

## Progress

You can check your progress by running:

```
opunit verify -i test/inventory.yml
```

You will want to be able to pass these basic checks:

```
Checks

        Basic checks for ansible server

        version check
            ✓   ansible --version: 2.9.4 > ^2.9.x => true
        reachable check
            ✓   [/bakerx] status: true
```

```
            ✓   [/bakerx/cm/inventory.ini] status: true
        contains check
                Checking if you have MSDOS style newlines in your bash scripts. Fix
with dos2unix
            ✓   [...run-ansible.sh] does not contain [\r] status: true message: NA
        contains check
            ✓   [...server-init.sh] does not contain [\r] status: true message: NA

Checks

        Basic checks for dependencies

        version check
            ✕   mysql --version: bash: mysql: command not found > ^5.7.x => false
        availability check
            ✕   [vagrant] http://192.168.33.80:8065/ expected: 200 actual:
ECONNREFUSED
        reachable check
            ✕   [/opt/mattermost/data] status: false
            ✕   [/lib/systemd/system/mattermost.service] status: false
        service check
            ✕   [mattermost] expected: active actual: null
        contains check
            ✕   [...config.json]  contains ["DriverName" : "mysql"] status: false
message: Error: file doesn't exist

Summary

        0.0% of all checks passed.
        0 passed · 6 failed
```

# Submission

Please [submit your repo here]

The assignment is due Wednesday, Feburary 12th before midnight.

# Evaluation

- 50% Basic requirements.
- 30% Quality of scripts and following best practices.
- 20% Screencast and following instructions.

Max possible score: 125/100.