CIS4010 Assignment 2: One-Click Deployment

Your task is to automate deployment of VMs to either an AWS or Azure platform. On these VMs you can also load multiple Docker images and start some of them. Your goal is to make this as close to a one-click action as possible.

Pre-Conditions

You may establish pre-conditions before your "one-click" script is activated. Everyone will have the first pre-condition but others are possible as long as you document them and they do not require onerous amounts of work (*i.e.* do not try to game the assignment – most of the work is done in the script).

Pre-Condition 1: Deployment description file

- Use a freely available editor, spreadsheet, etc. to create a file(s) in csv, JSON, or YMAL format. Items to be described include:
    - How many instances are to created and what platform is used to deploy and monitor them (AWS, Azure)
    - What attributes these instances have (VM name, size [only one allowed on educational accounts but include anyways], permanent disk size [disk usage is optional], and any other items that you think are necessary (e.g. ssh keys)
    - What Docker images are to be deployed on each instances and where those images reside (i.e. Docker Hub public registry, your private registry on Docker Hub)
    - Start images if they are to be used as background processes

Pre-Condition 2: Load Docker images to your own private Docker Hub registry

- You will be given a set of images to put on your registry before grading sessions begin.
- The images can be added manually – you do not need to script this.

Possible Optional Pre-Conditions

Pre-Condition 3: Create ssh keys

Pre-Condition 4: Create information file for monitoring purposes

- This could include names for the deployed VMs, what users have access to each VM, error logs, etc. You can get creative here.

One-Click Deployment

- Write a script (Python or bash) that will use all the pre-condition information to start creating VMs and configuring them. You may use APIs or CLIs.
- This script should be able to report on the status of what is happening and report errors if they occur.
- It is possible to have two scripts – one for AWS and one for Azure.

- For authentication purposes, you can login before running the script or ask for authentication information interactively in the script.

Monitoring

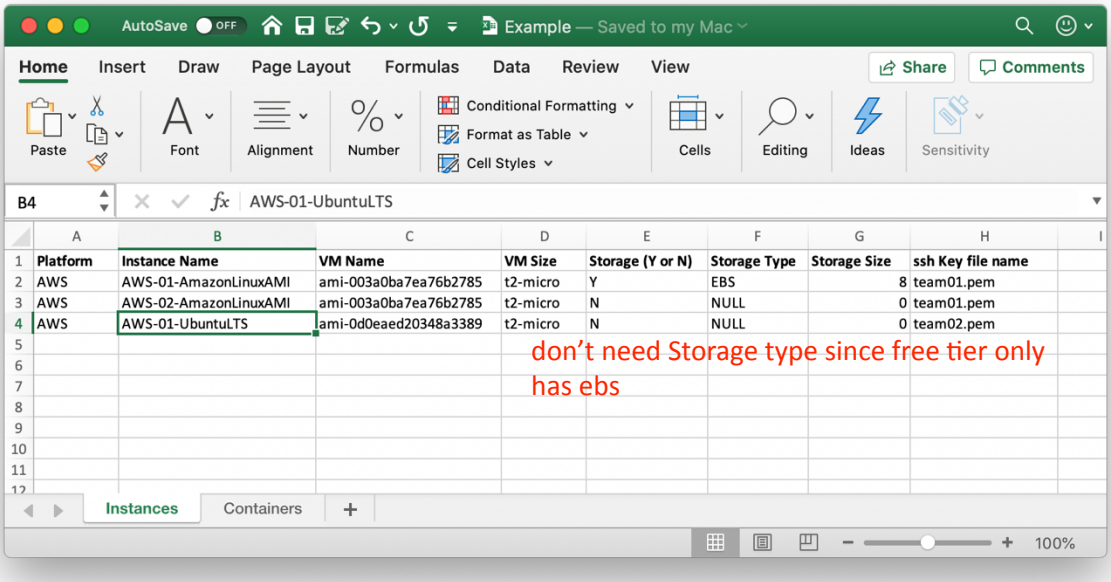
- After you have executed your “one-click” deployment, how will you demonstrate that everything has been done correctly?
- This can be done by either running a script or website that will show that the VMs are running and have the correct Docker images installed or it can be done manually through the dashboards for each Cloud platform (AWS and Azure)

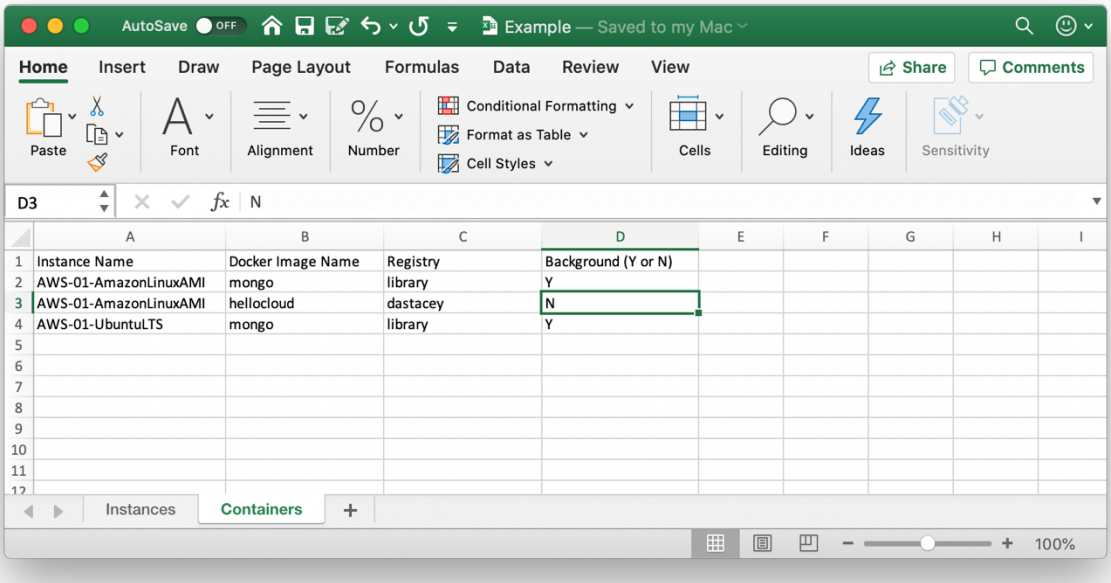
Example Scenario

I will use a spreadsheet to enter all the information. A template with appropriate column names will be given to the user and they will fill in all the information needed per VM. For example: Sheet 1 has the information about the VM configurations and Sheet 2 has the Docker information (see images below). I have only shown AWS instances in the example but could add either another spreadsheet format or use this one (maybe modified) for Azure.

I would also create the necessary ssh key files and load the necessary Docker images into my private repository on Docker Hub.

Then I would run deployment program: deploy.py or deploy.sh. This would create the instances and load the Docker images and start them in the background if required (reporting errors if they occur).





Then I would run my monitoring program: monitor.py or monitor.sh to show what is running. I might also have a webpage on each VM (if there is a web server running) that shows the state of the VM (see image of website – not exactly what I might show but similar).