Task 1:

Download and install a virtual machine on virtual box. The virtual machine should be a server. Enable password less SSH on the virtual machine to access it.

Read about SSH.

Task 2:

Install server image and give it IP address 192.168.80.15. NAT the VM to access internet and other VM’s.

Install the virtual machine on virtual box and enable it so that host can access it. To enable to VM to communicate with each other, enable NAT network on it.

Read about subnetting and NAT. Read network adapter types and where they are used.

Task 3:

Install nginx web server on a virtual machine. It will display the welcome page by using the local host. Configure it in a way that it displays the welcome page on the browser of your host machine.

Task 4:

Download two virtual machines and configure its network adapter. Using the virtual machine made in task 1, configure the virtual machines in a way that both the virtual machines use the one created in task 1 as a gateway to access the internet.

Also enable the virtual machines to communicate with each other.

Read about how to use a virtual machine as a router. Read about IP Masquerading.

Task 5:

After you have completed task 4, create another linux virtual machine and download NFS server on it. Download NFS clients on the 3 virtual machines created in task 4. Configure the NFS server in a way that any file placed in server can be shared over the clients.

Read NFS File system server and clients.

Task 6:

Read about AWS (mainly EC2 instances, availability zones and security groups). Learn about instances type and storage.

Imran bhai would have shared the credentials for AWS. You will have to create an EC2 instance. Change the availability zone to Ireland (staging area). Do not do this task in Oregan. It is the main production environment. Kindly do this task in the presence of Imran.

t2.nano using amazon Linux 2 AMI, 8Gb storage

SSH into the instance. Allow security groups to block all incoming traffic other than your IP.

After successfully completing this task. Delete the VM that you created on AWS.

Task 7:

Configure the installed nginx web server in a way that it displays the welcome page on your laptop but not on your mobile.

Task 8:

Read about shell scripting.

Create a virtual machine Ubuntu Server, not desktop. Write a shell script that takes user input. As a username and public key, and creates that user, and sets the permission, so the new user can SSH and also use the sudo command without a password.

Task 9:

The add user script that was created in Shell, enabled it to take array of values, and map array of SSH-public keys to it, and create multiple user. For Example: Users: Bill, Bob, Brian ….. SSH-Keys: Bill’s SSH-Key, Bob’s SSH-Key, Brain’s SSH-Key The Script should loop over the array, and create those users. Please make sure the script works for N number of users, where the value of N equals however many number of users entered.

Reading: Read about loops in SHELL Script. Ask Ashar/Mazhar for New Relic Credentials

Task 10:

Create a GITHUB account, and created a repo, upload your shell script in the repo, add a readme which will have set’s of instructions on how to run your script.

Reading: Read about version control, and branches in GIT, enable your branch to have multiple branches, i.e. Develop, Staging, Production.

Task 11:

GITHUB Continued: The Repo you created for your user script, organize into the following way, have a develop branch, staging branch and master. Lock all these branches, and make it compulsory, to create a Pull Request every time you modify your script. For Example: You need to update useradd.sh You will clone your branch, checkout develop, create a new branch from develop. Modify your script and then create a Pull request against develop. Once the pull request is approved merge it into develop, and then staging and then master.

Reading: Please read about how to put checks in GitHub, how you should manage the code, how to make sure all branches are in sync and are not drifted from each branch.

Catch up and on the week’s work. Make sure you didn’t just copy past, and have a good concept of your task. Share a summery in the email to Ismail, so that he can see your progress.

Task 12:

Update your Useradd script, and enable it to check the release for example you created it for Ubuntu, now download Centos, and create a VM, and make sure the script can detect which flavor of linux it is and create user’s accordingly. Once this is done, enable your script to delete users as well. For example once the scrip it run, it should provide an option 1. Add User 2. Delete existing user. Once it is successful push your changes to github

Task 11:

Start exploring new relic, have a session with the backend team, Please ask Asher to assign someone to give you a walk through on how new relic works, and what different kind of code errors are visible.

Reading:

Read and get a general understanding of Python 3

Task 12:

Create a new repo on GITHUB, and convert your shell script to run using python. This will take a few hit and trials, but please understand the language, again don’t just copy past.

By now Imran would have share Jenkins credentials. Log into staging Jenkins look around, please don’t modify any jobs. Read about Jenkins and declarative pipeline

Task 13:

Install Jenkins on a VM, access it from your windows laptop browser, set up a pipeline to run your shell / python script on a vm that you will use ssh to access via Jenkins.

Read about Docker, and docker-compose.

Task 14:

Download puppet learning vm and follow the steps to begin the quest challenge. It will give you some tasks that you have to complete to get the understanding of puppet.