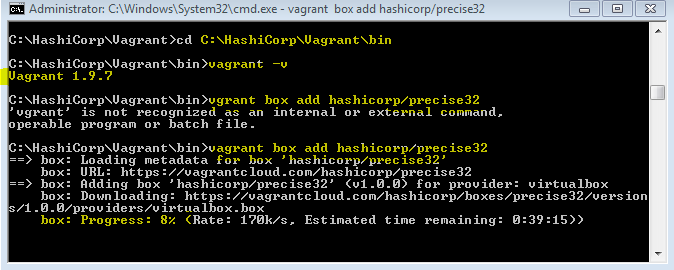
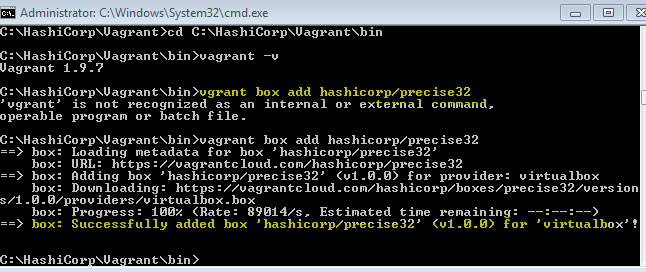
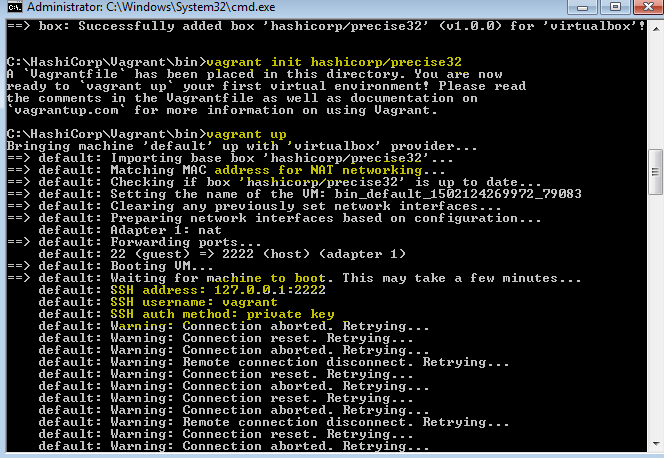
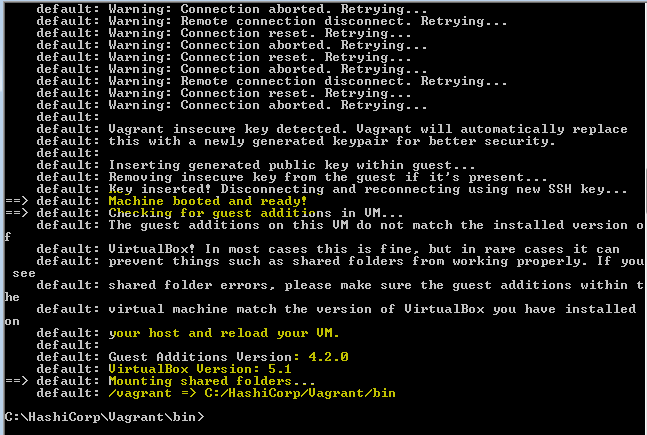
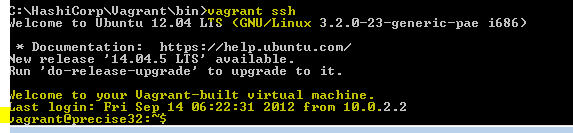
1. Setup
2. =====
3. 1. Download and install virtualbox (https://www.virtualbox.org/)
4. 2. Download and install vagrant (http://vagrantup.com)
5. Summary
6. =======
7. This test simulates real world problems a Dev/Ops Engineer may face and have to solve. You will have to learn how to use (potentially) new tools, diagnose system health issues, write some utilities, and automate the whole process to end up with a reproducable image.
8. At the end of this you should be able to send a new Vagrantfile and associated scripts/configs that will allow the interviewer to recreate your completed and fixed test environment from the same base image provided in the test Vagrantfile.
9. Instructions
10. ============
11. 1. Learn the basics of how to use Vagrant, launch the vagrant VM, and login to it via SSH (Note: this will take some time to download the VM image the first time)
12. 2. Analyze the health of the system. Note anything you find and fix if there are issues.
13. 3. Ensure apache is running. If is is not, diagnose and fix.
14. - You should be able to see a test page going here in your browser: http://localhost:8080/app
15. 4. Reconfigure apache to run using HTTPS rather than HTTP. Any HTTP links should redirect to HTTPS.
16. - You should be able to see the exact same test page as above by going here in your browser: https://localhost:8443/app
17. - You will get an exception about an insecure connection due to the self-signed cert. Feel free to ignore this.
18. 5. Install memcached
19. 6. Add a cronjob that runs /home/vagrant/exercise-memcached.sh once per minute # ROLE OF CRON JOB
20. 7. Write a python web application that outputs memcached stats
21. - It should additionally calculate the "get" hit rate and show it as a percentage ("X% of gets missed the cache")
22. - It should additionally show the percentage of memcached memory used
23. - Don't worry about the look and style of the page. Just make it functional.
24. - There is a basic python Flask app in /var/www/app you can use as a starting template
25. - If you don't know python or do not wish to use it, then you may use any other language you want. The catch is you will have to configure it yourself so it serves on port 80 and 443 (Must be HTTPS) inside the VM.
26. 8. Write your own vagrant file and provisioning scripts/configs that recreate everything you just did.
27. 9. Create a tar.gz of your new vagrant file with any associated provisioning scripts/configs and send back to the interviewer.
28. The interviewer should be able to simply un-tar your archive, run `vagrant up`, and have your completed and fixed environment running for review.



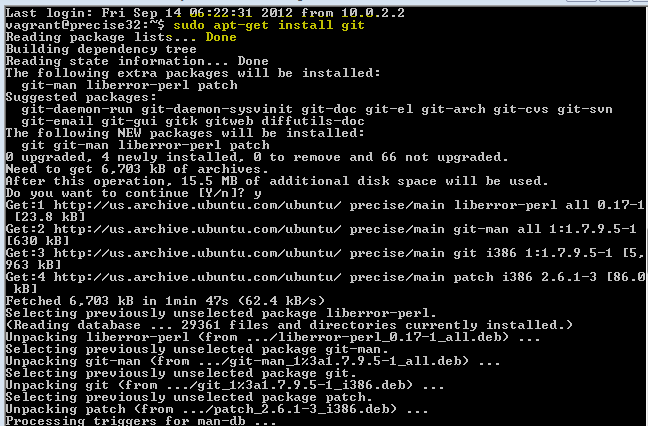


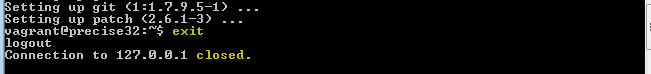


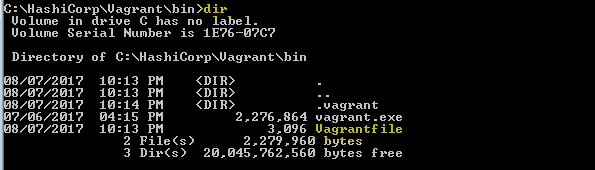


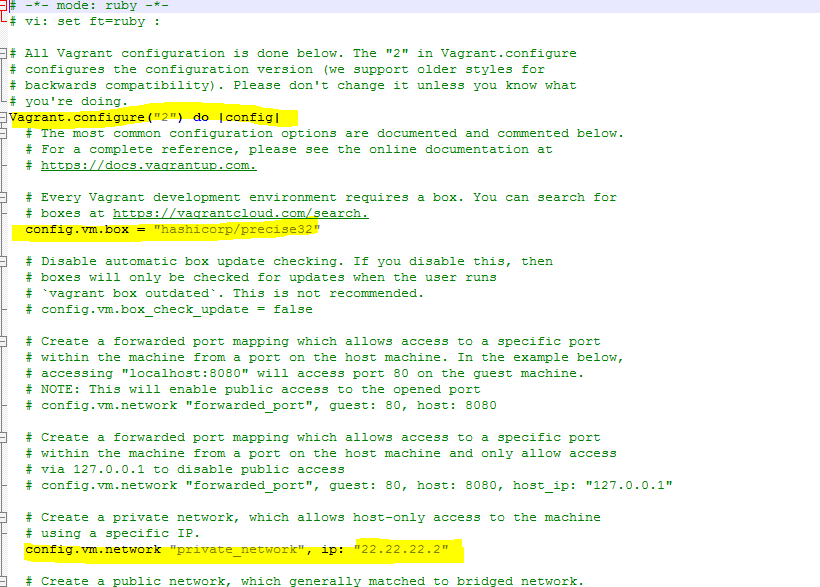


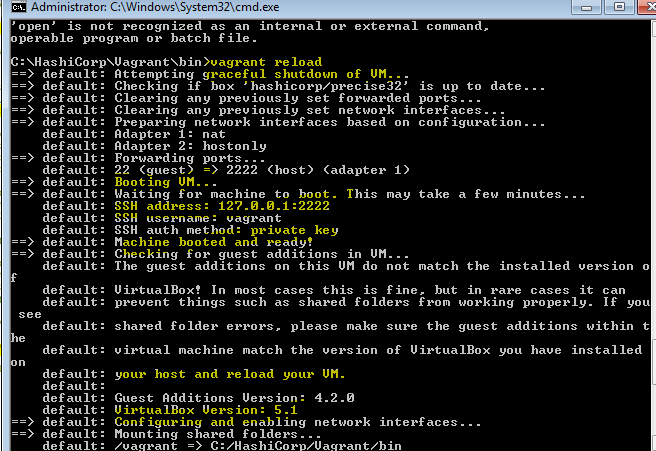
**Installing GIT**

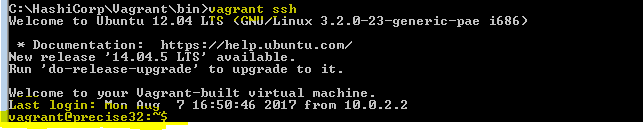


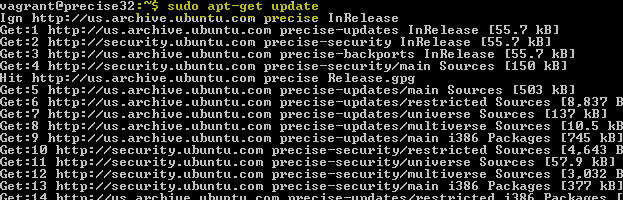


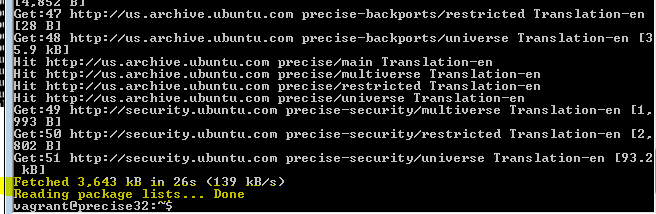




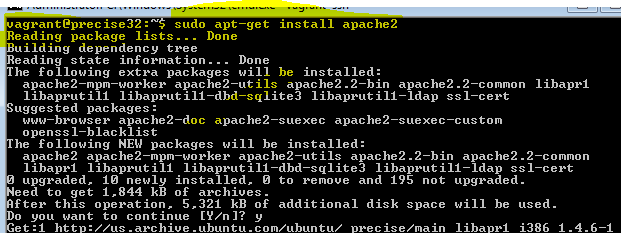
-

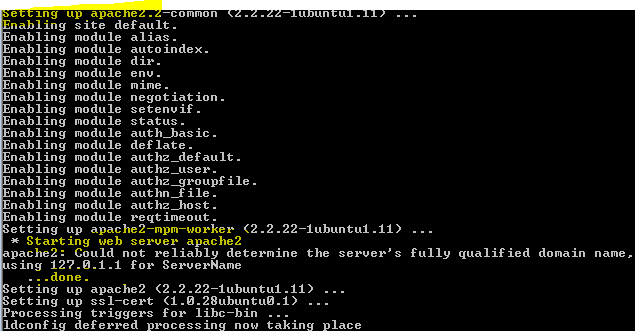


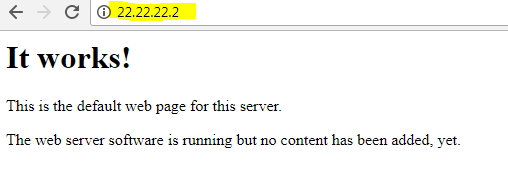




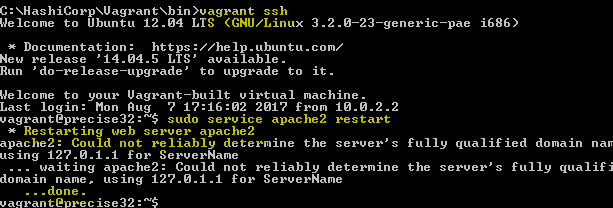
**Installing Apache**

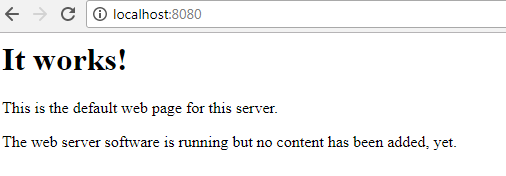






**Changed config file and reloaded and restarted apache result**





**APACHE PATH :**

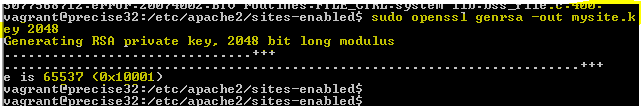


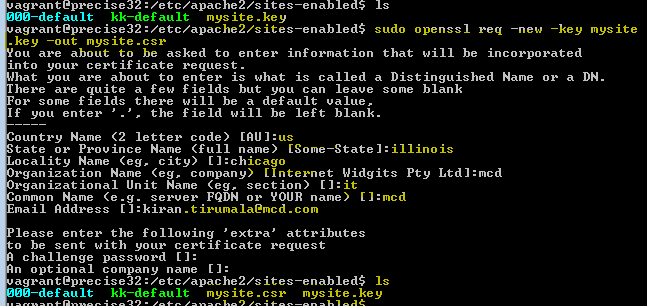
***HTTPS CONFIGURATION:***



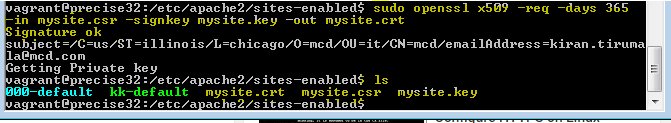


**Generating RSA key & CSR**



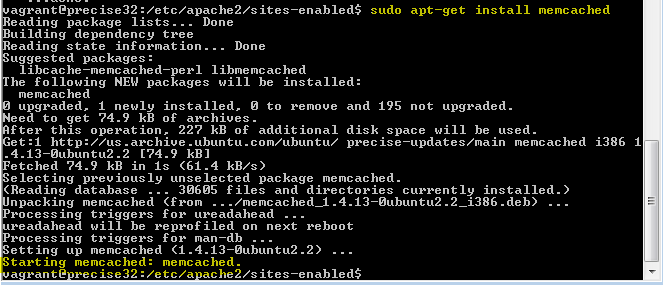


**Crt file creation**





**Installed mem cached**



**Crontjob creation**

