1. Create a new VM from the ubuntu image.
2. Name the VM as “yourname-Ansible”. Power on the VM

**A screenshot of a computer

Description automatically generated**

1. Open the terminal and log in to VM with User ID: “Yourname” and use the password “Devops”.

A screenshot of a computer program

Description automatically generated

1. Get the IP address of the VM and access the machine using Putty

A screenshot of a computer

Description automatically generated

A computer screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Add a new user named “ansible”, set the password as “Devops”, add it to root group, reboot machine, and log in with user credentials. Perform entire lab with newly created VM user.

A screenshot of a computer program

Description automatically generated

1. Change the Hostname of VM to “<yourname>-ansible”.

A black screen with white text

Description automatically generated

1. Upgrade and update your host.

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

1. Install curl on your machine

A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

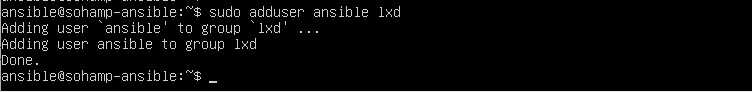
**PART 1 – Task: Linux Containers:**

1. To get started with Linux containers, we need to install the LXD package related to it first, download, and install a package using the following command.

A screen shot of a computer

Description automatically generated

1. Now we will add the user to the LXD group for management purposes.



1. Reboot the machine so that changes will get applied immediately. And verify membership details

A screenshot of a computer program

Description automatically generated

1. Before we can start creating containers, we need to tell LXD a little bit about our storage and network needs.

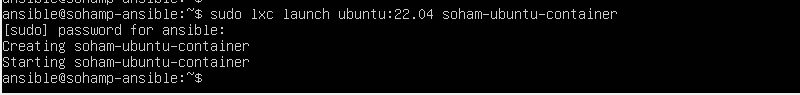
A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Let’s launch Ubuntu and centos containers





1. List the available images and list the currently running containers.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Access the shell of a container and Launch the shell of the ubuntu and centos containers you created to display Current working directory, IP address information, Current utilization.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Stop the containers you have created. Delete both the containers you created.

Before stopping and deleting the containers:

A screenshot of a computer

Description automatically generated

Stopping containers:





A screen shot of a computer

Description automatically generated

After deleting containers:



A black screen with white text

Description automatically generated

**Task: Launch Web Server inside a Container**

1. Launch a container using the ubuntu 18.04 image and name it as ‘<your-name>-web’

A screenshot of a computer

Description automatically generated

1. Login to the shell and then update and upgrade your container



A screenshot of a computer program

Description automatically generated

1. Install Nginx

A screenshot of a computer

Description automatically generated

1. Edit the default index file inside the container, located at /var/www/html/index.nginx-debian.html Make the following changes to the file, also mention your name in the script.



A computer screen shot of a black screen

Description automatically generated

1. Restart the Nginx server and find the container's IP address. Then verify the webpage you just created by accessing the URL from a web browser. (Use your container’s IP)

A screenshot of a computer program

Description automatically generated



A screenshot of a computer

Description automatically generated

1. Exit from the container’s shell and take a snapshot of your container and name it as ‘<your-name>-web-1.0’.



1. Verify whether the snapshot was created or not.

A screenshot of a computer

Description automatically generated

1. Add the following line at the end in the index file of your web container and restart Nginx. <p>Version 1.1</p>

A screenshot of a computer program

Description automatically generated

1. Verify the change by reloading the web page.

A screenshot of a computer

Description automatically generated

1. Create a new snapshot and name it ‘your-name’-web-1.1



1. List the 2 snapshots you created; snapshot names are expected to be seen.

A screenshot of a computer

Description automatically generated

1. Restore your container to ‘your-name’-web-1.0 and reload the page to see the webpage going back to the previous stage.



A screenshot of a computer

Description automatically generated

1. Delete the latest snapshot you created and verify the same.



A screenshot of a computer

Description automatically generated

14. Delete all containers you created till now.

A black screen with white text

Description automatically generated

**PART 2**

Ansible Installation:

1. Let’s update your system and install ansible.

A screenshot of a computer

Description automatically generated

A screen shot of a black screen

Description automatically generated

A screen shot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generated

1. Verify installation by running commands

A screenshot of a computer program

Description automatically generated

1. Now issue ‘which ansible’ and provide the output



1. Using Linux container, run 2 containers called 'ansible-target1' and 'ansible-target2' based on ubuntu:18.04 image. These containers will be our target machines which we will control using ansible. Access the shell of containers and note down their IP address.

A screen shot of a black screen

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. To manage target machines using ansible, target machines need to have ‘ssh’ and ‘python3’ installed. Install ‘ssh’ and ‘python3’ on both target containers and provide proof for the same.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

1. Issue the following commands to enable password key-based login : ssh-keygen -t rsa

A screenshot of a computer

Description automatically generated

1. Change the root user’s (inside the container) password to “Devops”. (Perform this for both containers)

A screen shot of a computer

Description automatically generated

A black screen with white text

Description automatically generated

1. Edit the ‘sshd\_config’ file from both containers

A black screen with white text

Description automatically generated

A screenshot of a computer program

Description automatically generated





A screenshot of a computer program

Description automatically generated



1. Copy the above key into your both target machine’s authorized\_keys file located at ‘.ssh/authorized\_keys’.

A screen shot of a computer

Description automatically generated

A computer screen with white text

Description automatically generated

1. Create a directory named ‘<yourname>’ and navigate to it.

A black background with white text

Description automatically generated

1. Create an inventory file named ‘inventory’ and edit it as shown in the following snapshot. Add the IP Addresses of both containers in ‘webservers’ group.

A screenshot of a computer

Description automatically generated

1. Ansible Ad-Hoc commands.

A computer screen shot of a computer program

Description automatically generated

A computer screen shot of a computer program

Description automatically generated

**PART 3 – Ansible Ad-Hoc Commands**

1. Create 3 more containers using the same ubuntu image and name them APP1, APP2, and DB.

A screenshot of a computer program

Description automatically generated

1. What are Ansible Modules? List at least 7 ansible modules.

Ans: Ansible modules referred to as “task plugins” or “library plugins” are independent pieces of code that can be accessed via the command line or as a component of a playbook task. Each module is executed, usually on the distant target node, and the results are gathered.

There are many ansible modules among them 7 ansible modules are:

* Ping: To check if the host is reachable.
* Command: a shell command is carried out on the targeted hosts.
* Shell: Conducts shell operation on the targeted hosts.
* Script: Performs local script execution on the distant hosts.
* Copy: Transfers files from the control computer to distant hosts.
* File: Controls the files and the directories on distant hosts.
* Templates: Generates files on remote hosts using Jinja2 templates yum (for Red hat/Cent OS) and apt (for Debian/ Ubuntu) on Linux-based systems, manages packages.

1. Install ‘python’ and ‘ssh’ in the newly created containers. Add your host’s ssh key into the target machines (Containers).

A screenshot of a computer program

Description automatically generated

A screen shot of a computer

Description automatically generated



A screenshot of a computer program

Description automatically generated



A black screen with white text

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer error

Description automatically generated



A screenshot of a computer program

Description automatically generated



A black screen with white text

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer error

Description automatically generated



A screenshot of a computer screen

Description automatically generated



A black background with white text

Description automatically generated

A screenshot of a computer screen

Description automatically generated

1. Edit the inventory file and add the APP1 and APP2 container’s IP in a new group called ‘application’ and the DB IP address in a group called ‘database’. Also, add their IPs in the default group ‘web servers’

A screen shot of a computer

Description automatically generated

1. Paste the output of the ansible ad-hoc ping command to ping web servers, applications, and databases separately. You should get success messages.

A screenshot of a computer

Description automatically generated

A computer screen shot of a computer code

Description automatically generated

A computer screen shot of a computer program

Description automatically generated

1. Paste the output of the ansible Ad-Hoc command to update the cache and upgrade the ‘database’ container.

A screenshot of a computer program

Description automatically generated

1. RUN Ansible ad-hoc command on the ‘application’ group which creates a user with ‘yourname’ having the primary group as root. Set password as ‘Devops.



A screenshot of a computer

Description automatically generated

1. RUN Ansible ad-hoc command on ‘application’ which creates a file ‘SMU’ on the location ‘/opt’ and the file should have read, write, execute permission to Owner (user) and Group, and only read permission to Others.

A screenshot of a computer

Description automatically generated

1. RUN Ansible ad-hoc command on ‘database’ which installs ‘apache2’ and starts its service.



A screenshot of a computer program

Description automatically generated



A screen shot of a computer

Description automatically generated

1. RUN ansible Ad-Hoc command on ‘database’ to gather facts.



A screenshot of a computer screen

Description automatically generated

1. Run Ansible ad-hoc command on all containers which will return the amount of free memory.

A screenshot of a computer

Description automatically generated

**PART 4 – Ansible Playbooks**

**NOTE: I faced an issue with my containers, so I had to create it again. I have not repeated PART 3 as it would a huge work to repeat. So, I repeated the steps which are needed to run the further steps.**

1. Add your inventory file to Ansible’s configuration file (ansible.cfg), so that you won’t need to provide it each time. To do this you need to copy the content of the ‘inventory’ file from ‘<yourname>’ directory and paste it to ‘/etc/ansible/hosts’.

A computer screen with a black screen

Description automatically generated

1. Create a Playbook named ‘firstplaybook.yml’ to install and start Nginx. It should also install init scripts, Ip route, MySQL on APP1 and APP2 containers, and MariaDB-server on DB container. Paste screenshots of the playbook, command, and output after running the playbook.



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated



A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Create a playbook named ‘secondplaybook.yml’ to add three users John, Michael, and Peter to only ‘APP1 container from application group’. Use Ansible Loops in this playbook. Paste screenshots of a playbook, the command used to run it, and its output after running the playbook.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Create a playbook named ‘thirdplaybook.yml’ to compress data from the containers ‘/var/log’ directory into the ‘ansible’ users (from Ansible host) home directory. Playbook must have Ansible variables for compression format, compression source path, and compression file

destination. Paste screenshots of a playbook, the command used to run it, and its output after running the playbook.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

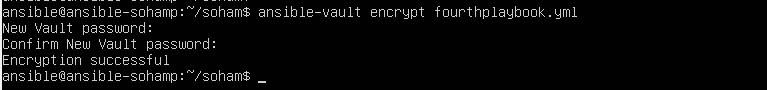


1. Create a playbook named ‘fourthplaybook.yml’ to install and start ‘FTP’ on the application group.

A screenshot of a computer

Description automatically generated

Encrypt this playbook using Ansible Vault.



Password provided: Devops

Check its content after encryption using the ‘cat fourthplaybook.yml’ command. Are you able to see exactly what was written in it? If not, then how will you check its content?

**Ans:** The content of the file is encrypted using AES256 encryption method, so I am not able to see the exact contents of the file. If I need to check the contents, I will have to decrypt the file using the password that I provided during encryption.

A screenshot of a computer screen

Description automatically generated

Once done, run this encrypted playbook on both containers from the ‘application’ group in such a way that Playbook will not get executed on both at the same time, it should run on 2 containers in one after another manner (Use Parallelism in Ansible). Paste screenshots of a playbook, various commands used here, and their outputs.

A screenshot of a computer

Description automatically generated

\_\_\_End of Lab\_\_\_