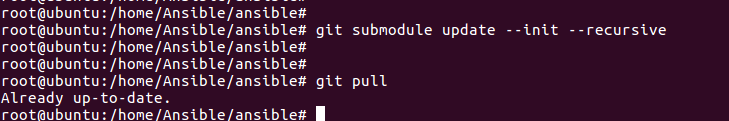
Automation using ansible.

Ansible installation:

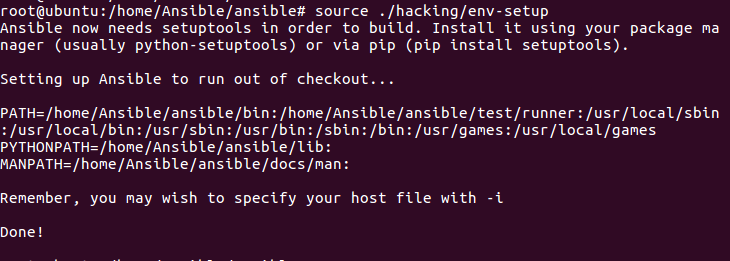
1. Install git-core and update installed packages.
2. In your home directory, create a folder ‘Ansible’ and get into it.
3. Clone the ansible repository available on git using following command
   1. git clone https://github.com/ansible/ansible
4. How would you check the commit logs of the repository?



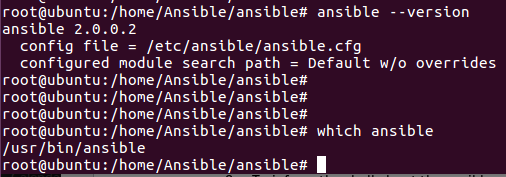
1. Update the ansible modules using following command;
   1. git submodule update --init --recursive
2. Make sure your git repository is updated using command;
   1. git pull



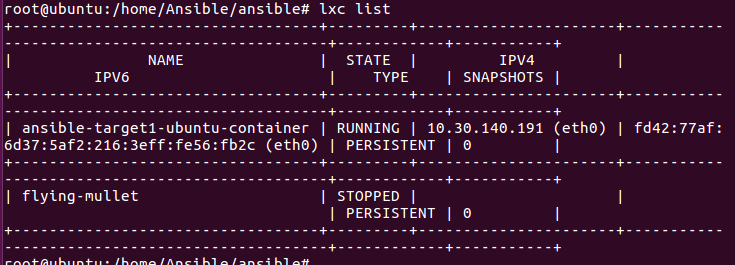
1. install the additional softwares needed by ansible to run on a host machine;
   1. apt-get install python-jinja2 python-paramiko python-yaml sshpass
2. To inform the shell about the ansible package, issue following command;
   1. ‘source ./hacking/env-setup’ inside your ansible git directory

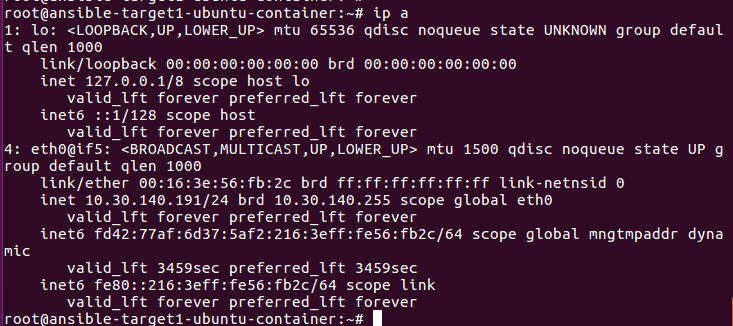


1. Now issue ‘which ansible’ and ‘ansible –version’ and provide the output

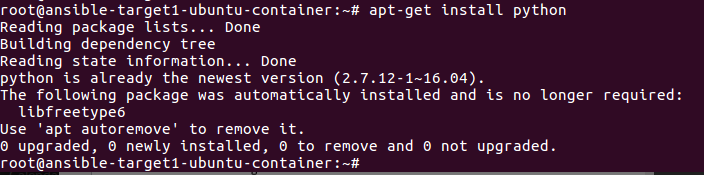


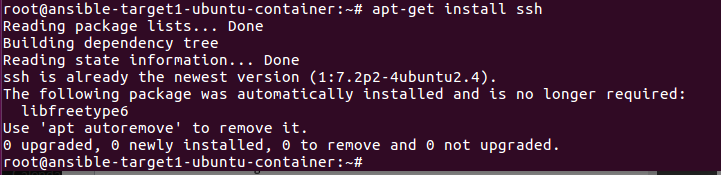
1. Using linux container lab document, install lxd and run a container called 'ansible-target1' based on ubuntu image. This container will be our target machine which we will control using ansible. Access the shell of container and note down the IP address.



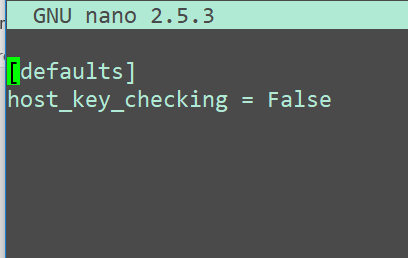


1. To manage target machines using ansible, it is important for target machines to have ssh and python. Install ssh and python on the target container and provide the proof for the same.

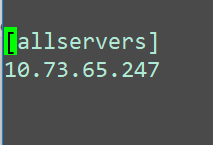




1. In your Ansible host machine, create a folder named ‘Ansible-Lab’ and create two files called ‘ansible.cfg’ and ‘inventory’.
2. Edit ansible.cfg as show below;

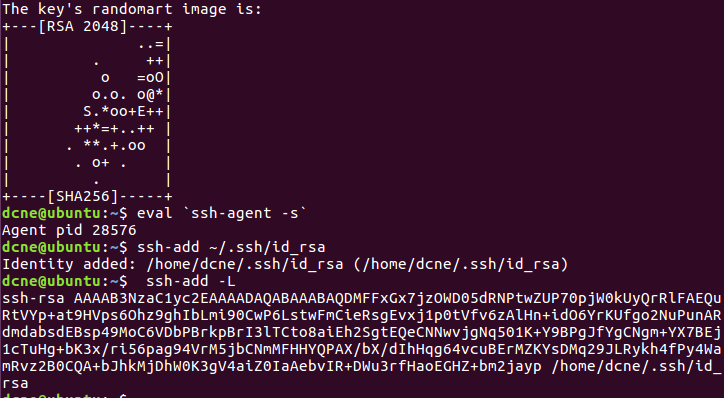


1. Edit the inventory file and create a section/group called ‘allservers’ and add your ansible-target machine (container’s IP) address below it

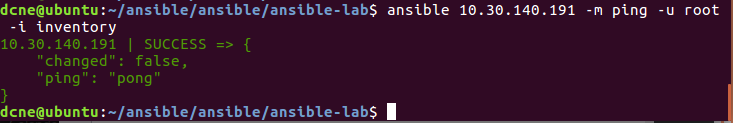


1. Issue the following commands to enable password key based login;
   1. ssh-keygen
   2. in your ansible-lab folder, issue ‘eval ‘ssh-agent -s’
   3. ssh-add ~/.ssh/id\_rsa
   4. issue ssh-add -L. You should see the output as given below;

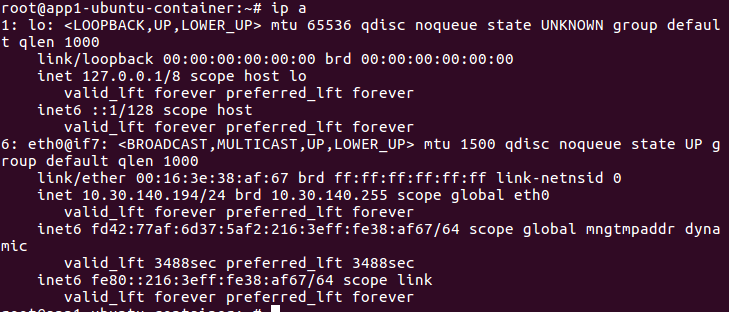
Copy the above key into your target machines authorized\_keys file located at /.ssh/authorized\_keys

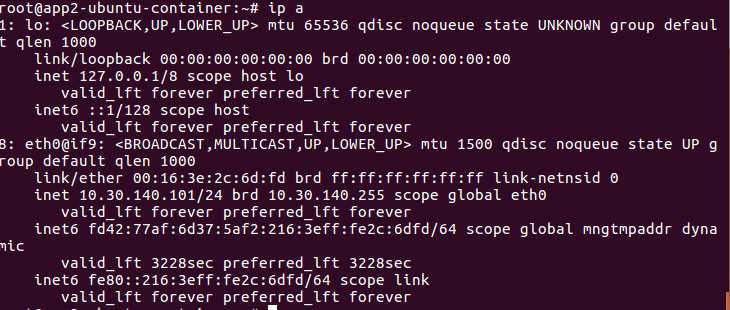


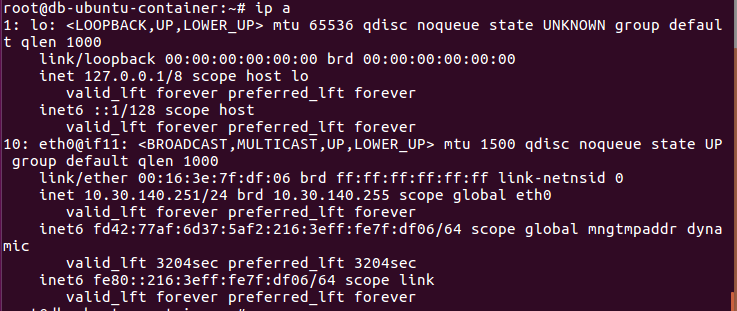
1. Ansible Ad-Hoc commands:
   1. ‘ansible ‘container-IP’ -m ping -u root’. You should see output as below;



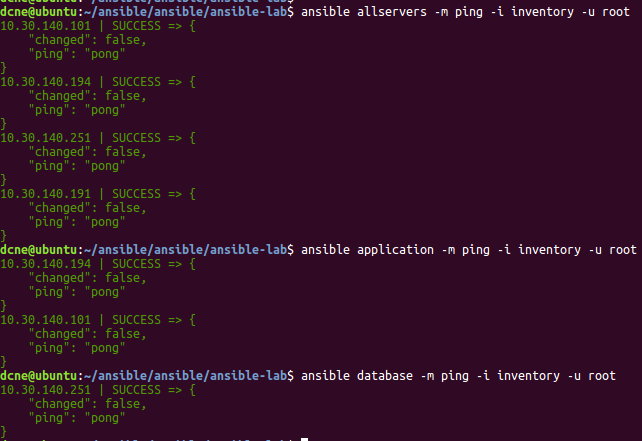
1. Task:
   1. Create 3 more containers and name them as APP1, APP2 and DB.
   2. Install python and ssh in the newly created containers. Add your host’s ssh key into the target machines. (Containers)
   3. Edit the inventory file and add APP1 and APP2 container’s IP in a new group called ‘application’ and DB IP address in a group called ‘database’. Also, add their IPs in the default group ‘allservers’



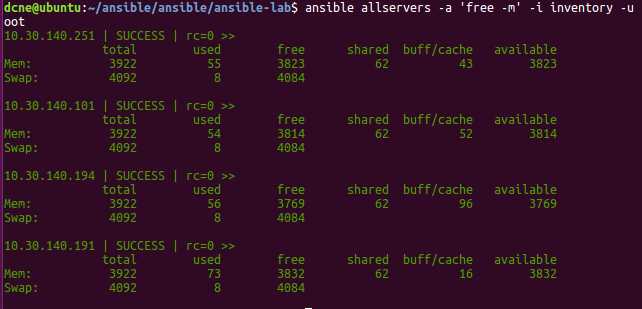




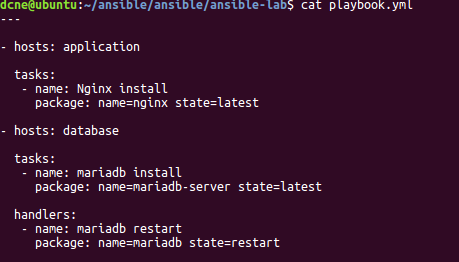
* 1. Paste the output of ansible ad-hoc ping command to allservers, application and database separately. You should get response as shown below;



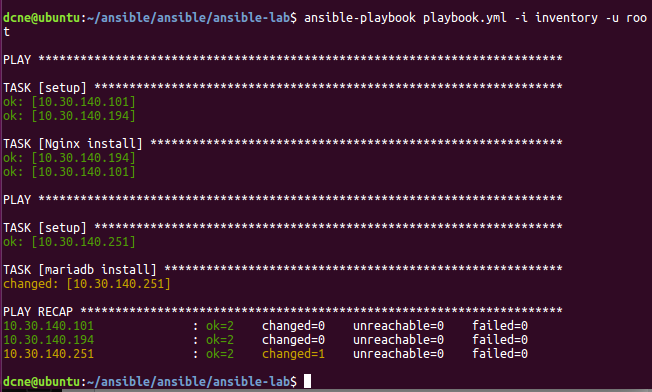
* 1. Run Ansible ad-hoc command on allservers which will return the amount of free memory on all the target machines. (command to see free memory is ‘free m’) Your output should look like as shown below



1. Simple playbook format to install a package:



1. Task:
   1. Create a Playbook which will install nginx on APP1 and APP2 containers and mariadb server on DB.
   2. Query all the servers to check whether the service is running or not.



That’s all folks!