Ansible is written in python.

The first thing ansible does is generate the inventory of hosts.

Install ansible on wsl:

sudo apt install openssh-client

sudo apt update

sudo apt upgrade -y

sudo apt install software-properties-common

sudo add-apt-repository --yes --update ppa:ansible/ansible

sudo apt install ansible -y

ansible --version

curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip" -o "awscliv2.zip"

unzip awscliv2.zip

sudo apt install unzip

unzip awscliv2.zip

sudo ./aws/install

aws --version

./run.sh test1 test2

aws configure

aws configure set region us-west-2

ls

./run.sh test1 test2

ls -l RoyG\_key.pem

chmod 400 filename

chmod 400 RoyG\_key.pem

ls

53 ls -l RoyG\_key.pem

54 ansible-playbook -i inventory.yaml playbook.yaml

55 ssh -i ubuntu@34.208.163.121

56 ssh ubuntu@34.208.163.121 -i /home/roy/ansible/RoyG\_key.pem

57 ls

58 ansible-playbook -i inventory\_aws\_ec2.yaml playbook.yaml

Don’t need virtual environemnet!

Not 100% sure it’s safe, but using –break-system-packages is the only way it works:

python3 -m pip install boto3 --break-system-packages

Which instances will it affect?

ansible-inventory -i inventory\_aws\_ec2.yaml –list

ansible-inventory -i inventory\_aws\_ec2.yaml --list > inventory\_output.json

72 ansible-playbook -i inventory\_aws\_ec2.yaml playbook.yaml

79 exit

80 ssh ubuntu@35.92.36.92 -i /home/roy/ansible/RoyG\_key.pem

83 hostname

To return to PowerShell from WSL: File > Close Remote Connection.

To log in to WSL as root: wsl -u root

Change studio code to run with WSL. button left hand side small icon.

before lauching machines via code (run.sh), we must install aws CLI, and then

configure it.

aws configure

Go to AWS IAM -> I am user33 -> Create access key -> CLI -> Enter access key and secret access key. VERY IMPORTANT: deactivate access key when not in use. Delete access key when finished. Also, the secret key only appears when you first create it and cannot be accessed later.

To show all instances:

aws ec2 describe-instances

What modules does ansible have? See ansible documentation. Click module to see explanation and examples.

Create files:

Note file types- very important!

* playbook.yaml
* run.sh
* inventory.yaml
* ansible.cfg

There are also other types of inventory files such as aws\_ec2 inventory. In that case, it’s important to include the “aws\_ec2” in the file name.

Using run.sh to create instances:

chmod 755 run.sh

./run.sh

In run.sh, let’s say we created 5 machines.

After creation, we can take the ip addresses of all machines and paste them into the inventory.ini under [all]

inventory\_aws\_ec2.yaml

connect to machine from WSL:

ssh ubuntu@34.219.161.197 -i ~/RoyG\_key.pem

it is also possible to write a dynamic inventory file, using python code (importing boto3, an aws python library), but we’ll probably never have to do that.

Roles: see playbook2

After playbook2 ran, we can check whether jenkins was installed correctly by ip in url+port.

And check whether nfs was installed by logging into machine and sudo -I -> cat /etc/export and we can that the file was updated, and ls systemctl status nfs-server

Creating roles:

In order to simplify the playbook, because it’s too long:

Create a roles folder on the same level as the other files (playbook etc).

Inside roles create a folder called nfs and a folder called jenkins.

Inside the nfs folder, create a folder called tasks. And the same in the jenkins folder.

Inside jenkins/tasks create a file called main.yaml

And the same in nfs/tasks.

Cut from the playbook from “tasks” onwards into the relevant main.yaml files.

And now in the playbook, instead of tasks, we’ll writer roles (i.e name of folder: either jenkins or nfs). See playbook3.

We have an option to debug ansible before running it.

ansible -I inventory\_aws\_ec2.yaml roy\_purp1 -m debug -a “var=jenkins\_ip\_global”

We want to let nfs know the jenkins ip, which is dynamic. Therefore we need to grab it on the go. Ansible first collects facts about the machine. So the ip is there. We can export it to a json and see which field we need.

Example of command from ansible documentation:

Ansible <hostname> -m ansible.builtin.setup -a “filter=ansible\_local”.

Then we use set\_fact (in the jenkins part). If we end the var name with “\_global”,

it knows it’s global.

In the jenkins machine:

df -h

here we should be able to see that /mnt/export is mapped to var/jenkins\_home

cat /etc/fstab and here as well.

docker ps

docker inspect \*enter docker id\*

and here we should be able to see the mount is ok under “Mounts” and “Volumes”

Now check permissions:

ls -lah /var/jenkins\_home/

owner should be ubuntu ubuntu and r/w/x