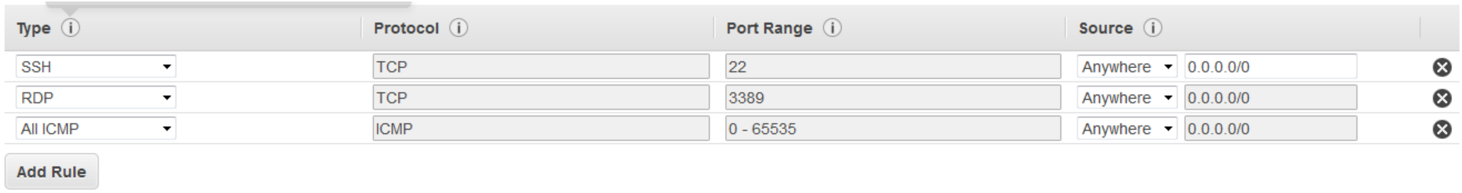
First create 2 (3?) EC2 Ubuntu 20.04 instances on AWS. (22.04 has some bugs with installing ansible easily)

Makes sure to set their security group to allow ssh and rdp and ICMP for pings.



Type	Protocol	Port Range	Source	
SSH	TCP	22	Anywhere	0.0.0.0/0
RDP	TCP	3389	Anywhere	0.0.0.0/0
All ICMP	ICMP	0 - 65535	Anywhere	0.0.0.0/0

SSH - TCP – 22

RDP – TCP – 3389

ALL ICMPv4 – ICMP – 0-65535

All Set Source to “Anywhere”

Give them names “Ansible Master” and “Ansible Host”

Usernames for EC2 instance types: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/connection-prereqs.html

SSH into both machines with PuTTY.

To SSH to Ubuntu vm with putty:

Copy the DNS hostname into the hostname field in PuTTY: e.g. ec2-34-245-156-35.eu-west-1.compute.amazonaws.com

In “Connection->SSH->Auth” we must place our key, however the .pem format is not accepted by putty.

Start “Puttygen”, select type of key as “ssh-2 RSA” or “RSA” if no option for ssh-2. Load the .pem key (need to set it to show all files). “Save Private Key” – ignore warning about no passcode

If you have the .ppk key, but no .pem version. Open “PuttyGen” and load the .ppk key. Select “Conversions->OpenSSH” and click yes on the pop-up, then save the .pem version somewhere you can find it.

Login as user “ubuntu”, the key handles the password.

##

On the MASTER:

Install Ansible (older version):

sudo apt update

sudo apt install ansible

(may give an error of files not existing, repeat it if so)

INSTALLING PIP:

```
sudo apt install python3-pip
```

INSTALL BOTO3:

```
pip install boto3
```

SETTING UP THE MASTER TO ALLOW IT TO CONNECT TO THE HOST:

Add the .pem key to the MASTER to allow it to SSH to the HOST(s).

```
vi ~/.ssh/id_rsa
```

Copy content of the .pem file (all of it) and use right-click to paste it into the id_rsa file.

Save with Esc->":wq"

In PuTTYGen, load the private key .ppk file. Copy the public key into the file id_rsa.pub on the MASTER:

```
vi ~/.ssh/id_rsa.pub
```

Add read and write privileges:

```
chmod 0600 ~/.ssh/id*
```

https://www.linode.com/docs/guides/modify-file-permissions-with-chmod/

Test sshing to the host with:

```
ssh ubuntu@<host public ip address>
```

After verifying you connected successfully, you can type "exit" to close the ssh connection.

##

Ansible Inventory Files:

http://docs.ansible.com/ansible/latest/user_guide/intro_inventory.html

Located at /etc/ansible/hosts

make a copy of it before editing : sudo cp hosts hosts_orig

```
sudo vi hosts
```

(if you forget sudo, do :q! to exit)

File contents below:

```
        [<group name>]

        <host public ip address>
```

^^^File contents above:

test it:

Collections listings:

https://docs.ansible.com/ansible/latest/collections/index.html

ping at:

https://docs.ansible.com/ansible/latest/collections/ansible/builtin/index.html#plugins-in-ansible-builtin

ansible –m ping <group name>

"all" will be all hosts from all groups. Localhost will run on the invoking machine!

Another example: aws_s3_bucket_info
https://docs.ansible.com/ansible/latest/modules/aws_s3_bucket_info_module.html#aws-s3-bucket-info-module

requires boto3 too! also install the module itself "`ansible-galaxy collection install community.aws`"

pip install boto3

ansible -m aws_s3_bucket_info localhost

- ➔ error no credentials!!

uh oh, what do we do to set them??

http://docs.ansible.com/ansible/latest/scenario_guides/guide_aws.html

export AWS_ACCESS_KEY_ID='AKIAV7Q4WIBC7BB2E457'   #manually type the single quotes..

export AWS_SECRET_ACCESS_KEY =' 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'

#ansible -m community.aws.aws_s3_bucket_info localhost

ansible -m aws_s3_bucket_info localhost

trying it on the group/all won't work since there's no environment variables set on the hosts!

##

Ansible Playbooks:

http://docs.ansible.com/ansible/latest/user_guide/playbooks.html

simple playbook example of using ping first!

ansible-playbook <playbook-name>

YAML syntax: https://docs.ansible.com/ansible/latest/reference_appendices/YAMLSyntax.html#yaml-syntax

##

Can check if apache2 is running or not on the target host pc using:

sudo systemctl status apache2   (from the host)

http://docs.ansible.com/ansible/latest/modules/apt_module.html

We want to use “apt” to install apache2!

http://docs.ansible.com/ansible/latest/modules/copy_module.html#copy-module

We want to copy over an index.html file to the apache2 server! (we must write it first too…)

http://docs.ansible.com/ansible/latest/modules/file_module.html

want to set the owner and group to www-data

(google apache2 www-data to explain this one…)

Use “service” module to start/stop a service (e.g. the apache2 server)

https://docs.ansible.com/ansible/latest/modules/service_module.html

##

http://docs.ansible.com/ansible/latest/dev_guide/developing_api.html

DO NOT USE, NOT ADEQUATELY SUPPORTED!

***** USING os.system in python TO RUN SHELL COMMANDS ******

import os

os.system(“ls”) #list directory contents

os.system(“ansible-playbook mybook.yaml”) #run a playbook called mybook.yaml