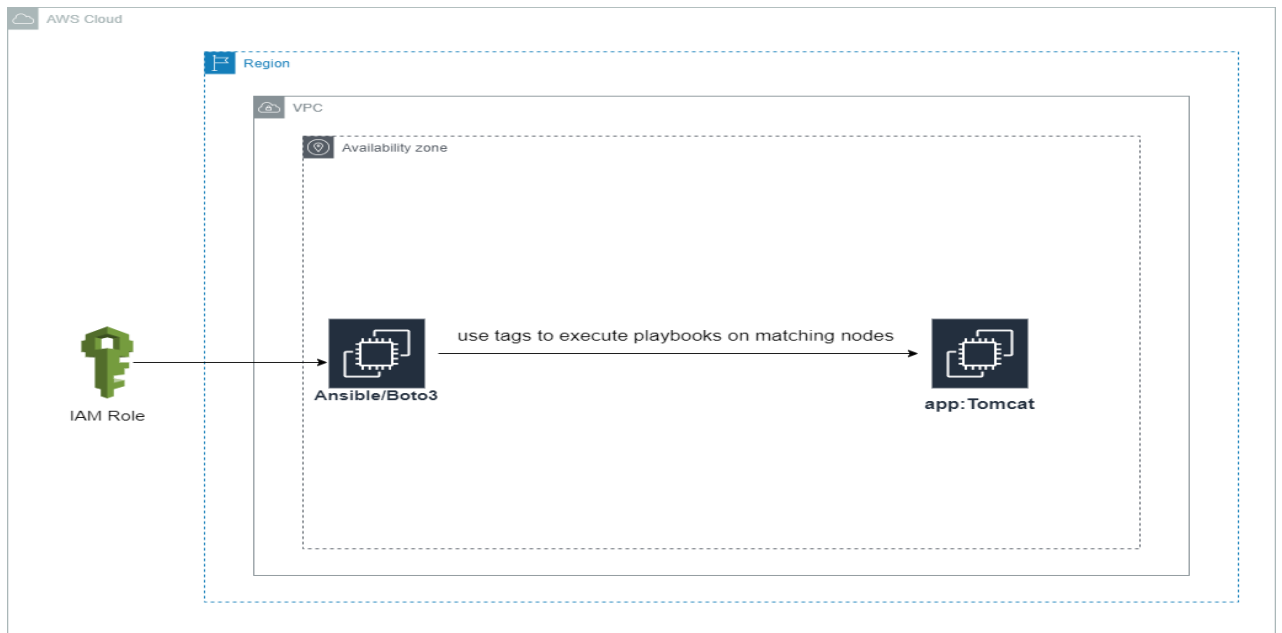


Ansible Dynamic Inventory AWS

<https://www.linkedin.com/in/aziz-afoni-811503227/>



Content

1. Setup Ansible Dynamic inventory
2. Grouping EC2 resources with dynamic inventory
3. Execute ansible commands with ec2 dynamic inventory
4. Using dynamic inventory inside a playbook

Prerequisites

1. AWS account
2. Knowledge in Ansible

Step 1: check if python3 & pip3 are installed in ansible server

- Python --version
- Used the following commands to install python and pip if you don't have it
- For Debian, Ubuntu-> `sudo apt-get install python3 -y`
- `sudo apt-get install python3-pip -y`
- For centos, Redhat, -> `sudo yum install python3 -y`
- `sudo yum -y install python3-pip`

step 2: installing boto3 library so ansible can be able to make API calls to AWS and retrieve EC2 instance details

```
- sudo pip3 install boto3
```

```
[ec2-user@ip-172-31-18-30 ~]$ sudo pip3 install boto3
WARNING: Running pip install with root privileges is generally not a good idea. Try 'pip3 install --user' instead.
Collecting boto3
  Downloading boto3-1.22.8-py3-none-any.whl (132 kB)
    | 132 kB 16.5 MB/s
Collecting s3transfer<0.6.0,>=0.5.0
  Downloading s3transfer-0.5.2-py3-none-any.whl (79 kB)
    | 79 kB 14.1 MB/s
Collecting jmespath<2.0.0,>=0.7.1
  Downloading jmespath-1.0.0-py3-none-any.whl (23 kB)
Collecting botocore<1.26.0,>=1.25.8
  Downloading botocore-1.25.8-py3-none-any.whl (8.7 MB)
    | 8.7 MB 38.5 MB/s
Collecting python-dateutil<3.0.0,>=2.1
  Downloading python_dateutil-2.8.2-py2.py3-none-any.whl (247 kB)
    | 247 kB 58.7 MB/s
Collecting urllib3<1.27,>=1.25.4
  Downloading urllib3-1.26.9-py2.py3-none-any.whl (138 kB)
    | 138 kB 55.0 MB/s
Collecting six>=1.5
  Downloading six-1.16.0-py2.py3-none-any.whl (11 kB)
Installing collected packages: six, python-dateutil, jmespath, urllib3, botocore, s3transfer, boto3
Successfully installed boto3-1.22.8 botocore-1.25.8 jmespath-1.0.0 python-dateutil-2.8.2 s3transfer-0.5.2 six-1.16.0 urllib3-1.26.9
[ec2-user@ip-172-31-18-30 ~]$
```

step 3: create an inventory directory

- sudo mkdir -p /opt/ansible/inventory
- cd /opt/ansible/inventory
- create a file names aws_ec2.yaml in the inventory directory
- sudo touch aws_ec2.yaml
- open the file using any editor of your choice
- sudo vi aws_ec2.yaml

option1: if your ansible server is not running in aws use access key and secret access key. paste the following config in aws-ec2.yaml

N:B make sure your yaml script follows the right format can use a yaml validator to check

```
---
plugin: aws_ec2
aws_access_key:
aws_secret_key:
keyed_groups:
  - key: tags
    prefix: tag
```

option2: if your ansible is running on aws use an ec2 role for this demo I use an ec2 role with ec2fullaccess

- navigate to IAM -> Roles -> Create role -> select ec2 under use case -> amazonec2fullaccess
 - click next -> give a name to the role -> the create role
 - navigate back to ec2 -> select ansible server-> actions -> Security -> modify IAM role-> select the role and save
 - paste the following in /opt/ansible/inventory/aws_ec2.yaml
- ```
plugin: aws_ec2
keyed_groups:
 - key: tags
 prefix: tag
```

step 4 : open /etc/ansible/ansible.cfg

- sudo vi /etc/ansible/ansible.cfg
- Find the [inventory] section and add the following line to enable the ec2 plugin
- enable\_plugins = aws\_ec2

```
variable
module_compression = 'ZIP_DEFLATED'

This controls the cutoff point (in bytes) on --diff for files
set to 0 for unlimited (RAM may suffer).
max_diff_size = 1048576

This controls how ansible handles multiple --tags and --skip-tags arguments
on the CLI. If this is True then multiple arguments are merged together. If
it is False, then the last specified argument is used and the others are ignored.
This option will be removed in 2.8.
merge_multiple_cli_flags = True

Controls showing custom stats at the end, off by default
show_custom_stats = True

Controls which files to ignore when using a directory as inventory with
possibly multiple sources (both static and dynamic)
inventory_ignore_extensions = '.orig', '.bak', '.ini', '.cfg', '.retry', '.pyc', '.pyo'

This family of modules use an alternative execution path optimized for network appliances
only update this setting if you know how this works, otherwise it can break module execution
network_group_modules=ios, nxos, ios, iosxr, junos, vyos

When enabled, this option allows lookups (via variables like {{lookup('foo')}} or when used as
a loop with 'with_foo') to return data that is not marked 'unsafe'. This means the data may contain
Jinja2 templating language which will be run through the templating engine.
Enabling this could be a security issue
allow_unsafe_lookups = False

set default errors for all plays
any_errors_fatal = False

[inventory]
enable inventory plugins, default: 'host_list', 'script', 'auto', 'yaml', 'ini', 'toml'
enable_plugins = host_list, virtualbox, yaml, constructed
enable_plugins += aws_ec2
ignore these extensions when parsing a directory as inventory source
ignore_extensions = .pyc, .pyo, .swp, .bak, ~, .rpm, .md, .txt, ~, .orig, .ini, .cfg, .retry
ignore files matching these patterns when parsing a directory as inventory source
ignore_patterns=

If 'true' unparseable inventory sources become fatal errors, they are warnings otherwise.
unparsable_is_fatal=False

[privilege_escalation]
become_method
```

Step 5: test dynamic inventory configuration by listing ec2 instances

- ansible-inventory -i /opt/ansible/inventory/aws\_ec2.yaml --list
- if you encounter an error. ERROR! The ec2 dynamic inventory plugin requires boto3 and botocore.
- Use sudo yum install python-boto3 and then retry the command
- The above command returns the list of ec2 instances with all its parameters in JSON format.
- If you want to use the dynamic inventory as a default Ansible inventory, edit the /etc/ansible/ansible.cfg file and search for inventory parameters under defaults. Change the inventory parameter value as shown below.
- inventory = /opt/ansible/inventory/aws\_ec2.yaml

```
config file for ansible - https://ansible.com/
#
nearly all parameters can be overridden in ansible.cfg
or with command line flags; ansible will read ANSIBLE_CONFIG,
ansible.cfg in the current working directory, ansible.cfg in
the home directory or /etc/ansible/ansible.cfg, whichever it
finds first

[defaults]
some basic default values...
inventory = /opt/ansible/inventory/aws_ec2.yaml
library = /usr/share/ansible/library/
module_utils = /usr/share/ansible/module_utils/
plugin_utils = /usr/share/ansible/plugin_utils/
plugin_filters_cfg = /etc/ansible/plugin_filters.yml
remote_user = root
poll_interval = 15
forks = 5
hash_behaviour = smart
ssh_host_check = true
allow_ssh_host_keys = true
gathering = explicit
gather_timeout = 10
fact_path = /etc/ansible/facts.d
module_no_log = False
module_set_locale = False

plays will gather facts by default, which contain information about
the remote system.
#
smart - gather by default, but don't regather if already gathered
explicit - do not gather by default, must use gather_facts: True
implicit - do not gather by default, must use gather_facts: True
#
this only affects the gathering done by a play's gather_facts directive,
by default gathering retrieves all facts subsets
all - gather all subsets
network - gather min and network facts
hardware - gather hardware facts (longest facts to retrieve)
virtual - gather virtual facts
other - gather other facts from facter
you can combine them using comma (ex: network,virtual)
you can disable the using (ex: !hardware,!facter,!other)
a minimal set of facts is always gathered.
gather_subset = all

some hardware related facts are collected
at a maximum timeout of 10 seconds. This
action tells you increase or decrease that
timeout.
-- timeout --
```

- Now run inventory list command without passing the inventory file
- `Ansible-inventory --list`

```

"ip-172-31-24-45.ec2.internal",
"ip-172-31-30-227.us-west-2.compute.internal",
"ip-172-31-30-251.ec2.internal",
"ip-172-31-81-247.ec2.internal",
"ip-172-31-86-37.ec2.internal"
]
},
"tag_Name_Ansible": {
 "hosts": [
 "ec2-54-242-122-76.compute-1.amazonaws.com"
]
},
"tag_Name_linuxadmin_centos": {
 "hosts": [
 "ip-172-31-81-247.ec2.internal"
]
},
"tag_Name_jenkins_master": {
 "hosts": [
 "ip-172-31-86-37.ec2.internal"
]
},
"tag_Name_nexus": {
 "hosts": [
 "ip-172-31-24-45.ec2.internal"
]
},
"tag_Name_sonar": {
 "hosts": [
 "ip-172-31-30-251.ec2.internal"
]
},
"tag_Name_web1": {
 "hosts": [
 "ec2-3-94-82-223.compute-1.amazonaws.com"
]
},
"tag_Name_web2": {
 "hosts": [
 "ec2-3-87-86-287.compute-1.amazonaws.com"
]
},
"tag_department_accountant": {
 "hosts": [
 "ip-172-31-30-227.us-west-2.compute.internal"
]
},
"tag_name_my_first_instance": {
 "hosts": [
 "ip-172-31-30-227.us-west-2.compute.internal"
]
}
}
[ec2-user@ip-172-31-18-30 ~]$

```

## Step6: grouping ec2 resources based on tags

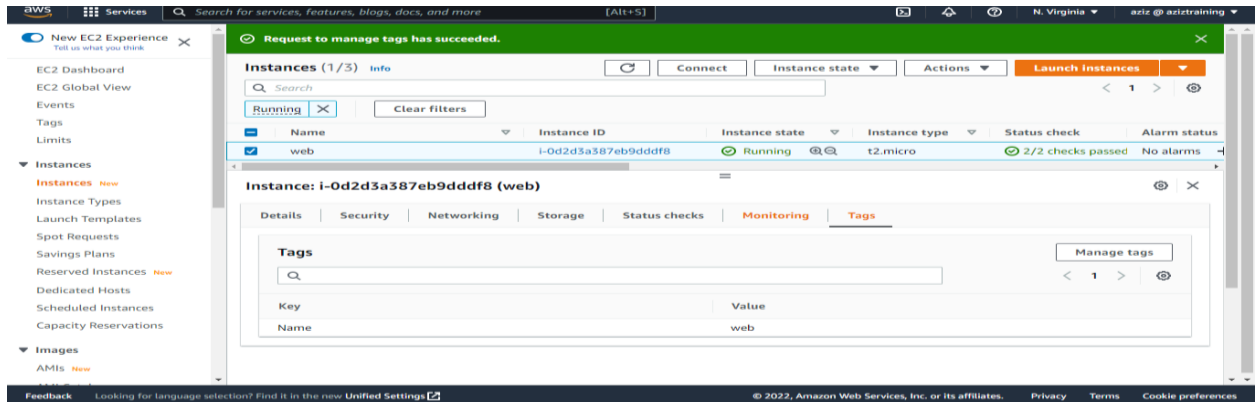
- The primary use case of AWS Ansible dynamic inventory is to execute Ansible playbooks or ad-hoc commands against a single or group of categorized or grouped instances based on tags, regions, or other ec2 parameters.
- you can group instances using tags, instances type, instance names, custom filters, and more. Take a look at all supported filters and keyed groups [from here](#).
- I will edit my ansible inventory file to group based on a specific tag

```

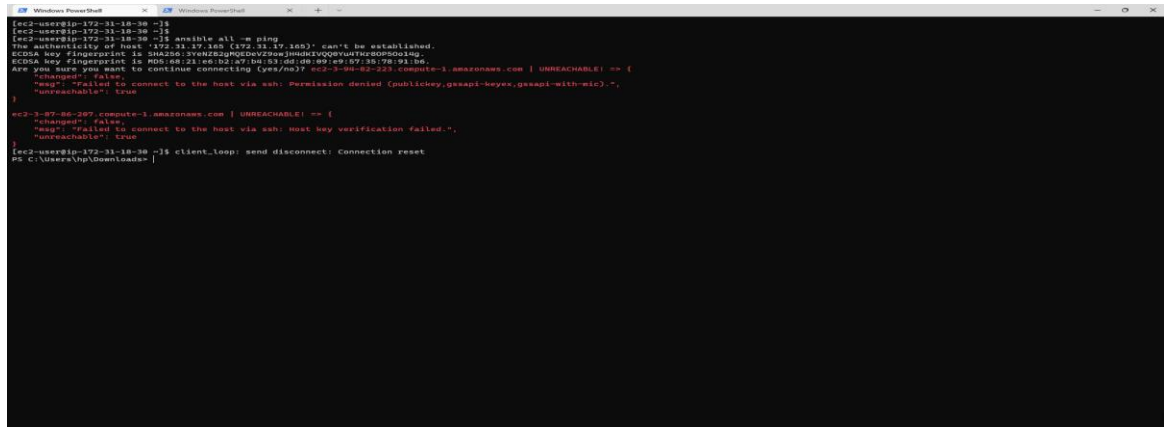
- ---
- plugin: aws_ec2
- regions :
- - us-east-1
- filters:
- tag:Name: web

```

- in this new inventory ansible get our ec2 instances that have a tag Name: web and in the region us-east-1



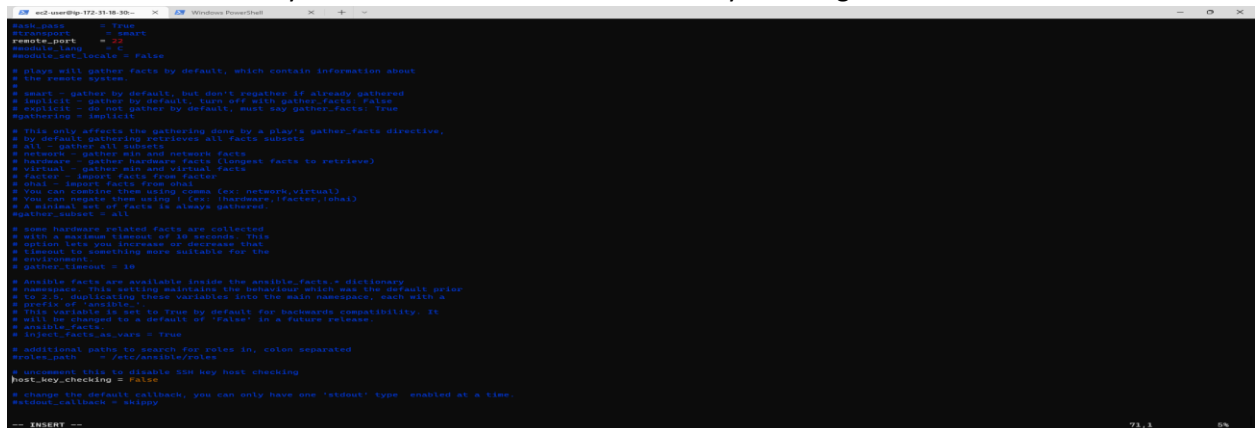
- let's ping them real quick



- NOTE: we are not able to ping the instances because we haven't configured an ssh agent yet
- Using SSH agent is the best way to authenticate to your end nodes as this elevates the need to copy your .pem files around

## Step 7: configure SSH

1. In sudo vi /etc/ansible/ansible.cfg enable the following by uncommenting
  - Remote\_port=22
  - Host-key\_checking= false
  - since ssh will be done by ansible there is no need for host key checking



- Uncomment remote user and change to your default user on your ec2 instance I am user linux so mine is ec2-user

```
ansible facts
inject facts as vars = True
additional paths to search for roles in, colon separated
#roles_path = /etc/ansible/roles
uncomment this to disable SSH key host checking
host_key_checking = false
change the default callback, you can only have one 'stdout' type enabled at a time.
stdout_callback = skippy

Ansible ships with some plugins that require whitelisting,
this is done to avoid running all of a type by default.
These setting lists those that you must enable for your system.
Custom plugins should not need this unless plugin author specifies it.

enable callback plugins, they can output to stdout but cannot be 'stdout' type.
#callback_whitelist = timer, mail

Determine whether includes in tasks and handlers are 'static' by
default. As of 2.0, includes are dynamic by default. Setting these
values to True will make includes behave more like they did in the
1.x versions.
#task_includes_static = False
#handler_includes_static = False

Controls if a missing handler for a notification event is an error or a warning
error_on_missing_handler = True

change this for alternative sudo implementations
sudo_exe = sudo

What flags to pass to sudo
WARNING: leaving out the defaults might create unexpected behaviours
sudo_flags = -H -s -n

SSH timeout
#timeout = 10

default user to use for playbooks if user is not specified
(user for ansible will use current user as default)
remote_user = ec2-user

logging is off by default unless this path is defined
if so defined, consider logrotate
log_path = /var/log/ansible.log

default module name for /usr/bin/ansible
module_name = command

use this shell for commands executed under sudo
-- INSERT --
```

- Create file for your private key in /etc/ansible and give it permission chmod 400 example.pem -> paste your private key inside and save -> navigate back to vi /etc/ansible/ansible.cfg and do the following
- Uncomment private\_key\_file and provide the path to the key file

```
if sudo is constrained
#timeout = /bin/su

if inventory variables overlap, does the higher precedence one win
or are hash values merged together? The default is 'replace' but
this can also be set to 'merge'
hash_behaviour = replace

by default, variables from roles will be visible in the global variable
scope. To prevent this, the following option can be enabled, and only
tasks and handlers within the role will see the variables there
#private_role_vars = yes

list any Jinja2 extensions to enable here:
#jinja2_extensions = jinja2.ext.do,jinja2.ext.i18n

if set, always use this private key file for authentication, same as
using ansible-private-key in ansible or ansible-playbook
private_key_file = /etc/ansible/ansible.pem

if set, configures the path to the Vault password file as an alternative to
specifying --vault-password-file on the command line
#vault_password_file = /path/to/vault_password_file

Format of string (if ansible_managed is) available within Jinja2
templates indicates to users editing templates files will be replaced.
replacing {file}, {host} and {uid} and strftime codes with proper values.
#ansible_managed = Ansible managed: {file} modified on %Y-%m-%d %H:%M:%S by {uid} on {host}
{file}, {host}, {uid}, and the timestamp can all interface with idioms
in some situations as the default is a static string:
#ansible_managed = Ansible managed

by default, ansible-playbook will display "Skipping [host]" if it determines a task
should not be run on a host. Set this to "False" if you don't want to see these "Skipping"
messages. NOTE: the task header will still be shown regardless of whether or not the
task is skipped.
#display_skipped_hosts = True

by default, if a task in a playbook does not include a name: field then
ansible-playbook will construct a header that includes the task's action but
not the task's args. This is a security feature because ansible cannot know
```

- Scroll down and uncomment everything under escalate privilege\_escalation

```
[inventory]
enable inventory plugins, default: 'host_list', 'script', 'auto', 'yaml', 'ini', 'toml'
#enable_plugins = host_list, virtualbox, yaml, constructed
enable_plugins = aws_ec2
ignore these extensions when parsing a directory as inventory source
#ignore_extensions = .pyc, .pyo, .swp, .bak, ~, .rpm, .md, .txt, ~, .orig, .ini, .cfg, .retry

ignore files matching these patterns when parsing a directory as inventory source
#ignore_patterns=

if 'true' unparsed inventory sources become fatal errors, they are warnings otherwise.
#unparsed_is_failed=False

[privilege_escalation]
become=True
become_method=sudo
become_user=root
become_ask_pass=False

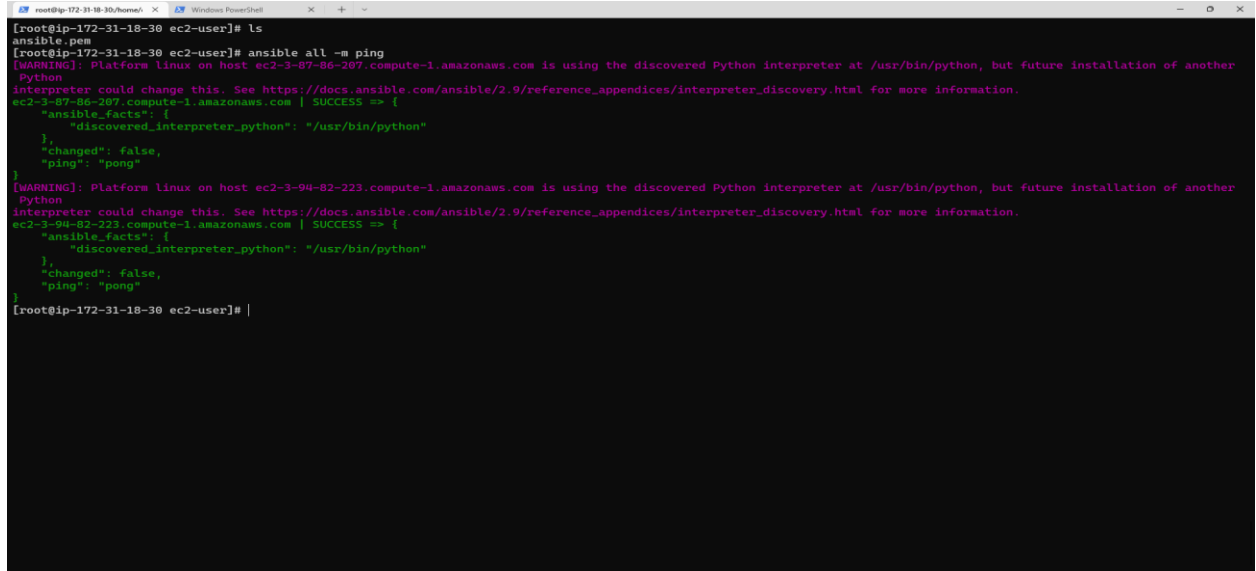
[paramiko_connection]

uncomment this line to cause the paramiko connection plugin to not record new host
keys encountered. Increases performance on new host additions. Setting works independently of the
host key checking setting above.
#record_host_keys=False

by default, Ansible requests a pseudo-terminal for commands executed under sudo. Uncomment this
line to disable this behaviour
#pty=False
```

## step 8: testing connectivity

- Let's run
- Ansible all -m ping



```
[root@ip-172-31-18-30 ~]# ls
ansible.pem
[root@ip-172-31-18-30 ec2-user]# ansible all -m ping
[WARNING]: Platform linux on host ec2-3-87-86-207.compute-1.amazonaws.com is using the discovered Python interpreter at /usr/bin/python, but future installation of another
Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
ec2-3-87-86-207.compute-1.amazonaws.com | SUCCESS => {
 "ansible_facts": {
 "discovered_interpreter_python": "/usr/bin/python"
 },
 "changed": false,
 "ping": "pong"
}
[WARNING]: Platform linux on host ec2-3-94-82-223.compute-1.amazonaws.com is using the discovered Python interpreter at /usr/bin/python, but future installation of another
Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
ec2-3-94-82-223.compute-1.amazonaws.com | SUCCESS => {
 "ansible_facts": {
 "discovered_interpreter_python": "/usr/bin/python"
 },
 "changed": false,
 "ping": "pong"
}
[root@ip-172-31-18-30 ec2-user]#
```

- 
- We can now ssh into our servers

**N:B**

As a best practice always secure your private key with ansible vault