Ansible components

Inventory

The “inventory” is a configuration file where you define the host information. In the above /etc/ansible/hosts example, we declared two servers under test-hosts.

Playbooks

In most cases – especially in enterprise environments – you should use Ansible playbooks. A playbook is where you define how to apply policies, declare configurations, orchestrate steps and launch tasks either synchronously or asynchronously on your servers. Each playbook is composed of one or more “plays”. Playbooks are normally maintained and managed in a version control system like Git. They are expressed in YAML (Yet Another Markup Language).

Plays

Playbooks contain plays. Plays are essentially groups of tasks that are performed on defined hosts to enforce your defined functions. Each play must specify a host or group of hosts. For example, using:

– hosts: all

…we specify all hosts. Note that YML files are very sensitive to white spaces, so be careful!

Tasks

Tasks are actions carried out by playbooks. One example of a task in an Apache playbook is:

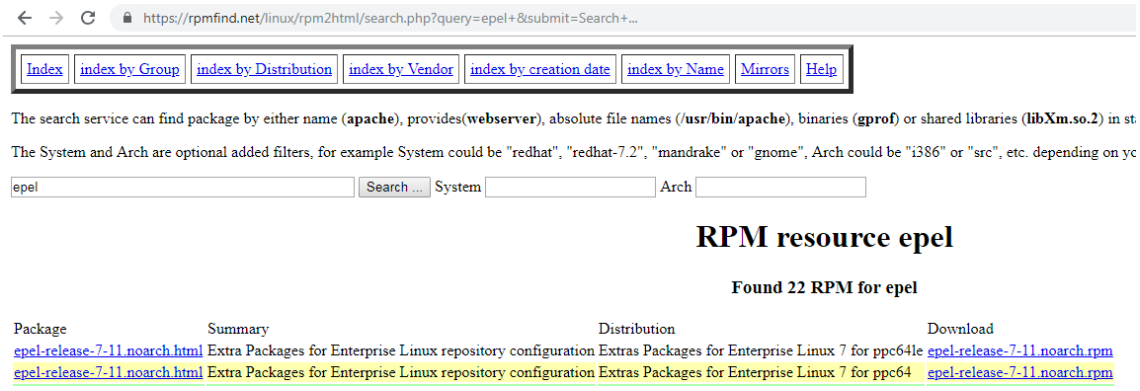
- name: Install Apache httpd

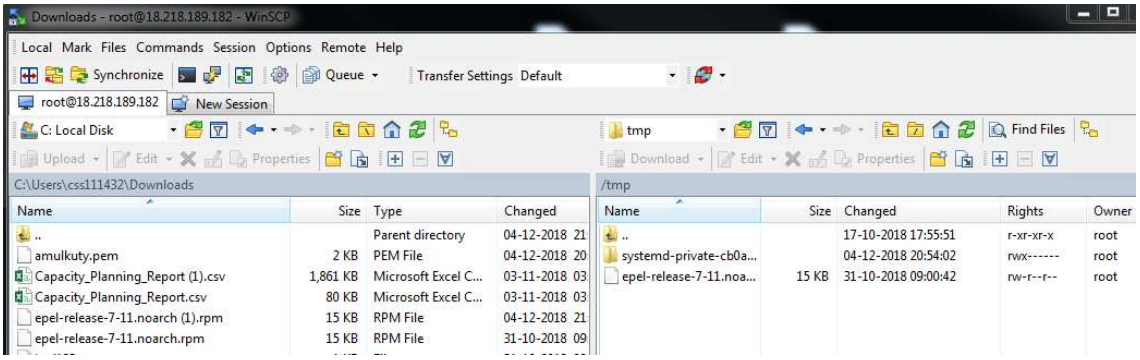
A task definition can contain modules such as yum, git, service, and copy.

Roles

A role is the Ansible way of bundling automation content and making it reusable. Roles are organizational components that can be assigned to a set of hosts to organize tasks. Therefore, instead of creating a monolithic playbook, we can create multiple roles, with each role assigned to complete a unit of work. For example: a webserver role can be defined to install Apache and Varnish on a specified group of servers.

```
[root@ip-172-31-44-249 ~]# yum install ansible
Loaded plugins: amazon-id, rhui-lb, search-disabled-repos
rhui-REGION-client-config-server-7
rhui-REGION-rhel-server-releases
rhui-REGION-rhel-server-rh-common
(1/7): rhui-REGION-rhel-server-releases/7Server/x86_64/group
(2/7): rhui-REGION-client-config-server-7/x86_64/primary_db
(3/7): rhui-REGION-rhel-server-rh-common/7Server/x86_64/updateinfo
(4/7): rhui-REGION-rhel-server-rh-common/7Server/x86_64/group
(5/7): rhui-REGION-rhel-server-releases/7Server/x86_64/updateinfo
(6/7): rhui-REGION-rhel-server-rh-common/7Server/x86_64/primary_db
(7/7): rhui-REGION-rhel-server-releases/7Server/x86_64/primary_db
No package ansible available.
Error: Nothing to do
```






```
[root@ip-172-31-44-249 ~]# cd /tmp
[root@ip-172-31-44-249 tmp]# ls
epel-release-7-11.noarch.rpm  systemd-private-cb0a6820658440f2adb56f997e35c706-chronyd.service-VOiDMN
[root@ip-172-31-44-249 tmp]# rpm -ivh epel-release-7-11.noarch.rpm
warning: epel-release-7-11.noarch.rpm: Header V3 RSA/SHA256 Signature, key ID 352c64e5: NOKEY
Preparing...                          ################################# [100%]
Updating / installing...
   1:epel-release-7-11                ################################# [100%]
[root@ip-172-31-44-249 tmp]#
```

```
[root@ip-172-31-44-249 tmp]# yum install ansible
Loaded plugins: amazon-id, rhui-lb, search-disabled-repos
epel/x86_64/metalink
epel
epel/x86_64/updateinfo          FAILED
http://csc.mcs.sdsmt.edu/epel/7/x86_64/repodata/26bade002130
Trying other mirror.
To address this issue please refer to the below knowledge ba
```

```
[root@ip-172-31-44-249 tmp]# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:gGrYmm+vfUBRxcYoLYF/oPEJcEnhpwvTfyygQ0ButZc root@ip-172-31-44-249.us-east-2.compute.interna
The key's randomart image is:
+---[RSA 2048]----+
|.++=.+.=.        |
|oo* *.+ +        |
|.o.BoE..         |
|o+.+* ..         |
|+.B. .  S        |
|.B +..           |
|= . o.o          |
| o.. o.          |
| .ooo.           |
+----[SHA256]-----+
[root@ip-172-31-44-249 tmp]# cd /root/.ssh/
[root@ip-172-31-44-249 .ssh]# ssh-copy-id root@172.31.43.71
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host '172.31.43.71 (172.31.43.71)' can't be established.
ECDSA key fingerprint is SHA256:dvTRhyY9xw6OOUw/f1m3YD/50THjY0DP0ajUEyF5drk.
ECDSA key fingerprint is MD5:bf:3c:69:4a:aa:a3:e8:37:79:18:c9:ac:86:ad:c1:a7.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are a
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to in
root@172.31.43.71's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'root@172.31.43.71'"
and check to make sure that only the key(s) you wanted were added.

[root@ip-172-31-44-249 .ssh]#
```

Client IP

```
[root@ip-172-31-44-249 .ssh]# vi /etc/ansible/hosts
```

```
# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
#   - Comments begin with the '#' character
#   - Blank lines are ignored
#   - Groups of hosts are delimited by [header] elements
#   - You can enter hostnames or ip addresses
#   - A hostname/ip can be a member of multiple groups

# Ex 1: Ungrouped hosts, specify before any group headers.

## green.example.com
## blue.example.com
## 192.168.100.1
## 192.168.100.10
172.31.43.71          Client
```

```
[root@ip-172-31-44-249 .ssh]# ansible all -m ping
172.31.43.71 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
[root@ip-172-31-44-249 .ssh]#
```

```
[root@ip-172-31-23-192 .ssh]# ansible appservers -m ping
172.31.21.215 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
172.31.21.241 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
[root@ip-172-31-23-192 .ssh]# cat /etc/ansible/hosts
[appservers]
172.31.21.241
172.31.21.215
[root@ip-172-31-23-192 .ssh]#
```

For getting the Modules details:

```
[root@ip-172-31-23-192 .ssh]# ansible-doc -l
a10_server                                          Manage A10 Networks AX...
a10_server_axapi3                                   Manage A10 Networks AX...
a10_service_group                                   Manage A10 Networks AX...
a10_virtual_server                                  Manage A10 Networks AX...
aci_aaa_user                                        Manage AAA users (aaa:...
aci_aaa_user_certificate                            Manage AAA user certif...
aci_access_port_to_interface_policy_leaf_profile    Manage Fabric interfac...
aci_aep                                             Manage attachable Acce...
aci_aep_to_domain                                   Bind AEPs to Physical ...
aci_ap                                              Manage top level Appli...
aci_bd                                              Manage Bridge Domains ...
```

Working with modules:

1) Shell Module

```
[root@ip-172-31-23-192 .ssh]# ansible appservers -m shell -a "cd /; mkdir test"
172.31.21.215 | CHANGED | rc=0 >>


172.31.21.241 | CHANGED | rc=0 >>


[root@ip-172-31-23-192 .ssh]#
```

```
[root@ip-172-31-23-192 .ssh]# ansible appservers -m shell -a "cd /; ls"
```

2) YUM Module

```
[root@ip-172-31-23-192 .ssh]# ansible appservers -m yum -a "name=httpd state=present" -s
[DEPRECATION WARNING]: The sudo command line option has been deprecated in favor of the "bec
setting deprecation_warnings=False in ansible.cfg.
172.31.21.241 | CHANGED => {
    "ansible_facts": {
        "pkg_mgr": "yum"
    },
    "changed": true,
```

3) Service Module

```
[root@ip-172-31-23-192 .ssh]# ansible appservers -m service -a "name=httpd state=started" -s
[DEPRECATION WARNING]: The sudo command line option has been deprecated in favor of the "become" co
setting deprecation_warnings=False in ansible.cfg.
172.31.21.241 | CHANGED => {
    "changed": true,
    "name": "httpd",
    "state": "started",
```

```
[root@ip-172-31-23-192 .ssh]# ansible appservers -m service -a "name=httpd state=stopped" -s
[DEPRECATION WARNING]: The sudo command line option has been deprecated in favor of the "become"
setting deprecation_warnings=False in ansible.cfg.
172.31.21.241 | CHANGED => {
    "changed": true,
    "name": "httpd",
    "state": "stopped",
```

```
[root@ip-172-31-23-192 .ssh]# ansible appservers -m service -a "name=httpd state=restarted" -s
[DEPRECATION WARNING]: The sudo command line option has been deprecated in favor of the "become"
setting deprecation_warnings=False in ansible.cfg.
172.31.21.241 | CHANGED => {
    "changed": true,
    "name": "httpd",
    "state": "started",
    "status": {
```

4) Copy Module:

```
[root@ip-172-31-23-192 /]# cd ansible/
[root@ip-172-31-23-192 ansible]# ls
epel-release-7-11.noarch (2).rpm  epel-release-7-11.noarch.rpm
[root@ip-172-31-23-192 ansible]# ansible all -m copy -a "src=/ansible/epel-release-7-11.noarch.rpm dest=/tmp/" -s
[DEPRECATION WARNING]: The sudo command line option has been deprecated in favor of the "become" command line argum
setting deprecation_warnings=False in ansible.cfg.
172.31.21.241 | CHANGED => {
    "changed": true,
```

Client:

```
[root@ip-172-31-21-215 ec2-user]# cd /tmp
[root@ip-172-31-21-215 tmp]# ll
total 16
-rw-r--r--. 1 root root 15080 Dec 17 15:49 epel-release-7-11.noarch.rpm
```

Working with Playbook:

Playbook in YAML:

Eg:1:-

```
[root@ip-172-31-23-192 ansible]# cat 1.yml
---
- hosts: appservers
  remote_user: root
  tasks:
  - name: install firefox
    yum: name=firefox state=present
  - name: start firefox
    service: name=firefox state=started
```

```
---
- hosts: appservers
  remote_user: root
  tasks:
  - name: install firefox
    yum: name=firefox state=present
  - name: start firefox
    service: name=firefox state=started
```

```
[root@ip-172-31-23-192 ansible]# ansible-playbook 1.yml

PLAY [appservers] ***************************************************************************************************

TASK [Gathering Facts] **********************************************************************************************
ok: [172.31.21.215]
ok: [172.31.21.241]

TASK [install firefox] **********************************************************************************************
changed: [172.31.21.215]
changed: [172.31.21.241]

TASK [start firefox] ************************************************************************************************
fatal: [172.31.21.215]: FAILED! => {"changed": false, "msg": "Could not find the requested service firefox: host"}
fatal: [172.31.21.241]: FAILED! => {"changed": false, "msg": "Could not find the requested service firefox: host"}
       to retry, use: --limit @/ansible/1.retry

PLAY RECAP **********************************************************************************************************
172.31.21.215              : ok=2    changed=1    unreachable=0    failed=1
172.31.21.241              : ok=2    changed=1    unreachable=0    failed=1
```

Eg:2:-

```
[root@ip-172-31-23-192 ansible]# cat 2.yml
---
- hosts: appservers
  remote_user: root
  tasks:
  - name: install java
    yum: name=java-1.8.0-openjdk-devel state=present
  - name: start Java
    service: name=java-1.8.0-openjdk-devel state=started
```

```
---
- hosts: appservers
  remote_user: root
  tasks:
  - name: install java
    yum: name=java-1.8.0-openjdk-devel state=present
  - name: start Java
    service: name=java-1.8.0-openjdk-devel state=started
```

```
[root@ip-172-31-23-192 ansible]# ansible-playbook 2.yml

PLAY [appservers] ***************************************************

TASK [Gathering Facts] **********************************************
ok: [172.31.21.241]
ok: [172.31.21.215]

TASK [install java] *************************************************
changed: [172.31.21.215]
changed: [172.31.21.241]
```

Interacting with AWS:

Installing boto3:

yum install -y python python-dev python-pip

```
[root@ip-172-31-23-192 ansible]# yum install -y python python-dev python-pip
Loaded plugins: amazon-id, rhui-lb, search-disabled-repos
No package python-dev available.
Resolving Dependencies
```

pip install boto3

```
[root@ip-172-31-23-192 ansible]# pip install boto3
Collecting boto3
  Downloading https://files.pythonhosted.org/package
    100% |                                        | 133kB 6.
```

Playbook for Security group creation:

```
[root@ip-172-31-23-192 ansible]# cat secgrp.yml
- hosts: localhost
  connection: local
  gather_facts: false
  tasks:
      - name: create a security group in us-west-1
        ec2_group:
          name: HariTest
          description: an example ec2 group
          region: us-west-1
          aws_access_key: "AKIAIQBVEC3W6NBU5WTQ"
          aws_secret_key: "RylwtDjOIgHWhjnti0H40Ff6AzA9QeqE/3k6O08Q"
          rules:
            - proto: tcp
              from_port: 80
              to_port: 80
              cidr_ip: 0.0.0.0/0
        register: security_group
```

```yaml
- hosts: localhost
  connection: local
  gather_facts: false
  tasks:
      - name: create a security group in us-west-1
        ec2_group:
          name: HariTest
          description: an example ec2 group
          region: us-west-1
          aws_access_key: "AKIAIQBVEC3W6NBU5WTQ"
          aws_secret_key: "RylwtDjOIgHWhjnti0H40Ff6AzA9QeqE/3k6O08Q"
          rules:
            - proto: tcp
              from_port: 80
              to_port: 80
              cidr_ip: 0.0.0.0/0
        register: security_group
```

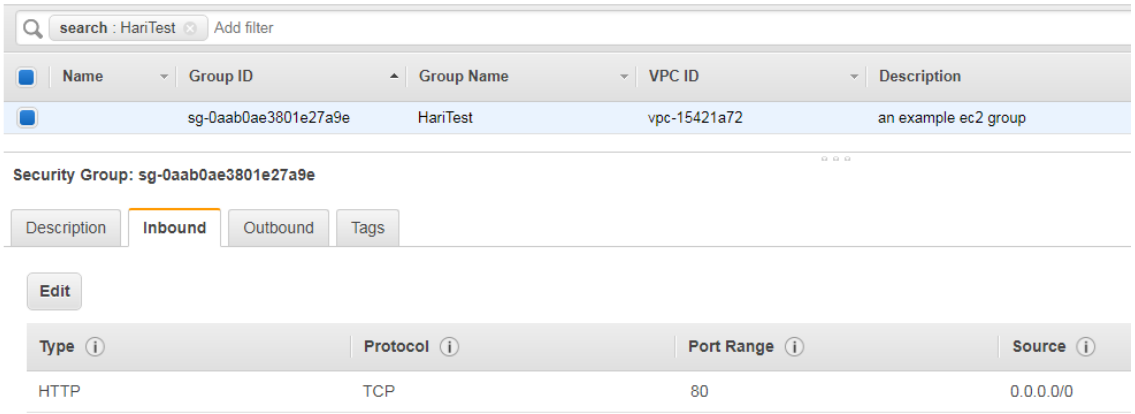
```
[root@ip-172-31-23-192 ansible]# ansible-playbook secgrp.yml

PLAY [localhost] ***************************************************************

TASK [create a security group in us-west-1] ************************************
changed: [localhost]

PLAY RECAP *********************************************************************
localhost                  : ok=1    changed=1    unreachable=0    failed=0
```

Output:



Playbook for Instance creation:

```
[root@ip-172-31-23-192 ansible]# cat 33.yml
- hosts: localhost
  connection: local
  gather_facts: false
  tasks:
      - name: create ec2 instance
        ec2:
          aws_access_key: "AKIAIQBVEC3W6NBU5WTQ"
          aws_secret_key: "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"
          image: ami-18726478
          wait: yes
          instance_type: t2.micro
          group_id: sg-0aab0ae3801e27a9e
          region: us-west-1
          count_tag:
            Name: apacheserver
          exact_count: 1
        register: ec2
```

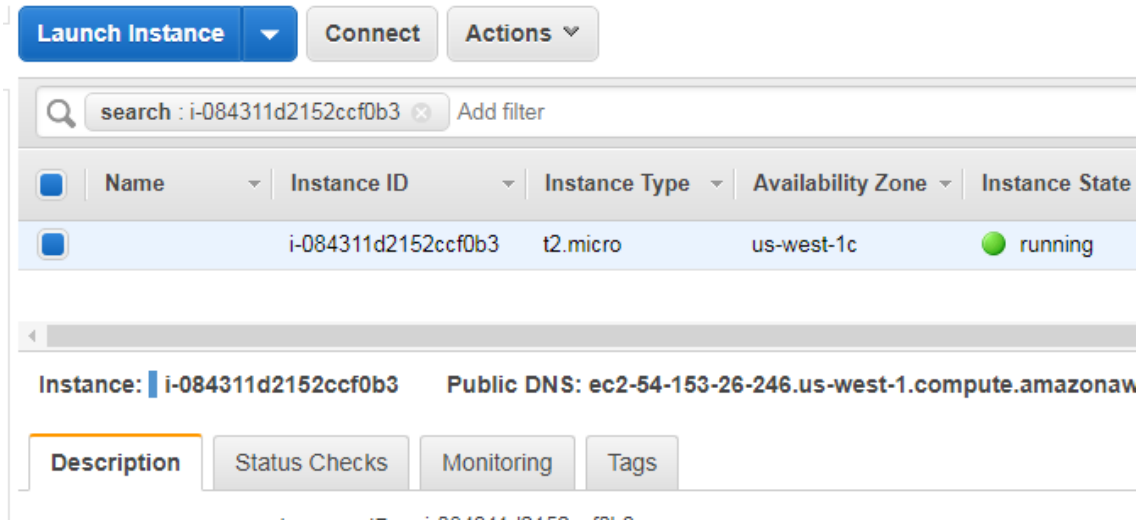
```yaml
- hosts: localhost
  connection: local
  gather_facts: false
  tasks:
    - name: create ec2 instance
      ec2:
        aws_access_key: "AKIAIQBVEC3W6NBU5WTQ"
        aws_secret_key: "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"
        image: ami-18726478
        wait: yes
        instance_type: t2.micro
        group_id: sg-0aab0ae3801e27a9e
        region: us-west-1
        count_tag:
          Name: apacheserver
        exact_count: 1
      register: ec2
```

```
[root@ip-172-31-23-192 ansible]# ansible-playbook 33.yml

PLAY [localhost] ***************************************************************

TASK [create ec2 instance] *****************************************************
changed: [localhost]

PLAY RECAP *********************************************************************
localhost                  : ok=1    changed=1    unreachable=0    failed=0
```



Reference Link : https://www.infinitypp.com/ansible/create-aws-resources-using-ansible/