Fetch Module (opposite of copy module)

Go to managed node

```
$ ssh 172.31.9.252
$ cd /etc/tomcat8
$ ls
```

There is server.xml file
I want to get the file (server.xml) from nodes to controller

```
$ exit  ( come back to controller )
$ ansible all  -m fetch  -a 'src=/etc/tomcat8/server.xml dest=/tmp'  -b
```

Now to got tmp folder

```
$  cd /tmp
$ ls
```

You will find three folders. The names of the folers are IP address of managed nodes

```
$ cd 172.31.35.102
$ ls
$ cd etc
$ ls
$ cd tomcat8
$ ls
```

Notes:
Fetch module is used to copy files from managed nodes to controller.
Command to copy tomcat-server.xml file from all managed nodes into /tmp folder on the controller.

```
$ ansible all  -m fetch  -a
```

```
'src=/etc/tomcat8/server.xml dest=/tmp'  -b
```

Git Modules

This is used to perform git version controlling on the managed nodes.

```
ansible all -m  git  -a 'repo=https://github.com/sunildevops77/repo1.git  dest=/tmp/mygit'  -b
```

The above command will download the files in all managed nodes.

Go to managed node and check

```
$  ssh 172.31.35.79
$ cd /tmp
$  ls
$ cd mygit
$  ls
$ exit
```

Notes:
Ansible command to clone remote git repository into all managed nodes

```
ansible all -m  git  -a 'repo=https://github.com/sunildevops77/rep1.git  dest=/tmp/mygit' -b
```

++++++++++++++++++

Service Module

This is used for starting/ stoping / restarting the services.

Ansible command to restart tomcat8 on all managed nodes

```
$  ansible all   -m service    -a 'name=tomcat8  state=restarted'  -b
```

state=restarted is for restarting a service
state=stopped is for stopping a running service
state=started is for starting a stopped service

Replace module

Go to managed node

```
$ ssh 172.31.36.52
$ cd /etc/tomcat8/
$ ls
$ sudo  vim  server.xml
```

Look for connector port , to see the port number in which it is running. (line 74)

Now, we want to change the port number on all managed nodes, in this scenario
we use replace module.

Quit the server.xml file

```
$ exit   ( to come back to controller )
$  ansible all  -m replace  -a
```

```
'regexp=8080 replace=9090
path=/etc/tomcat8/server.xml'  -b
```

Lets check tomcat is respoding on 9090 port in managed node

Get public DNS from aws
ec2-13-251-114-207.ap-southeast-1.compute.amazonaws.com
ec2-13-234-48-168.ap-south-1.compute.amazonaws.com

Open Browser
URL ---
ec2-13-251-114-207.ap-southeast-1.compute.amazonaws.com:9090

We will not get the page, becuase we need to restart the service

```
$  ansible all   -m service    -a
'name=tomcat8  state=restarted'  -b
```

Now, try the above URL --- it

Works!!

replace module

This is used for replacing a specific string with other string.
Ex:
Ansible command to change the port number of tomcat from 8080 to 9090

```
$ ansible all  -m replace  -a
'regexp=8080 replace=9090
path=/etc/tomcat8/server.xml'  -b
```

uri module

I want to check facebook is reachable for not in all managed nodes.

```
$ ansible all -m   uri  -a
'url=http://facebook.com'
```

In the output (green color)

status - 200

Give a invalid url , we get status as -1

Ex:

```
$ ansible all -m   uri  -a
'url=http://hgyi9cb.com'
```

Now, I want to stop tomcat in all managed nodes (Just repeat)

```
$  ansible all   -m service    -a
'name=tomcat8  state=stopped'  -b
```

Notes:

urI module is used to check if the url is reachable or not.

Command to check if facebook.com is reachable on all managed nodes.

```
$ ansible all -m   uri  -a
'url=http://facebook.com
status=200'
```

++++++++++++++++++++++++++++++++++++

Lets have an example of all modules
Requirement: I want to install tomcat all manages nodes , then i want to copy users.xml in all managed nodes,
I want to change port number of tomcat , then i want to restart the service, finally i want to check url is reachable or not.

1st we need to unintall tomcat in all managed nodes.

```
$ ansible  all  -m  apt -a
'name=tomcat8 state=absent
purge=yes'  -b
```

```
--------------
$  ansible all -m  apt -a
'name=tomcat8 state=present' -b
```

```
$  ansible all -m  copy  -a
'src=tomcat-users.xml
dest=/etc/tomcat8'  -b
```

```
$ ansible all  -m replace  -a
'regexp=8080 replace=9090
path=/etc/tomcat8/server.xml'  -b
$  ansible all   -m service    -a
'name=tomcat8  state=restarted'  -b
```

To check tomcat is running individually on all servers,
take the private ip of all nodes
172.31.11.96
172.31.6.207
172.31.12.138

```
$ ansible all -m   uri  -a
'url=http://172.31.11.96:9090'
```

It returns status as 200

Similarly check the other two nodes

```
$ ansible all -m   uri  -a
'url=http://172.31.6.207:9090'
$ ansible all -m   uri  -a
'url=http://172.31.12.138:9090'
```

++++++++++++++++++++

Notes:

Requirement.

I want to install tomcat all modules.Copy tomcat-users.xml in all managed nodes.
Change port number of tomcat from 8080 to 9090. Restart the tomcat8 service.
Finally i want to check url is reachable or not.

```
$  ansible all -m  apt -a
'name=tomcat8 state=present' -b

$  ansible all -m  copy  -a
'src=tomcat-users.xml
dest=/etc/tomcat8'  -b
$ ansible all  -m replace  -a
'regexp=8080 replace=9090
path=/etc/tomcat8/server.xml'  -b
$  ansible all   -m service    -a
'name=tomcat8  state=restarted'  -b
```

To check tomcat is running individually on all servers,
take the private ip of all nodes
172.31.11.96
172.31.6.207
172.31.12.138

```
$ ansible all -m   uri  -a
'url=http://172.31.11.96:9090
status=200'
```

It returns status as 200

Similarly check the other two nodes

```
$ ansible all -m   uri  -a
'url=http://172.31.6.207:9090
status=200'
$ ansible all -m   uri  -a
'url=http://172.31.12.138:9090
status=200'
```

Play books

Notes:

Adhoc commands are capable of working only on one module and one set of arguments.

When we want to perform complex configuration management activities,

adhoc commands will be difficult to manage.

In such scenarios, we use play books.

Play book is combination of plays.

Each play is designed to do some activity on the managed nodes.

These plays are created to work on single host or a group of hosts or all the hosts.

The main advantage of play books is reusability.

Play books are created using yaml files.

```
$ mkdir  playbooks
$ cd playbooks
$ vim playbook1.yml
INSERT   mode

---
- name: Install git and clone a
remote repository
  hosts: all
  tasks:
    - name: Install git
      apt:
       name: git
       state: present
       update_cache: yes
    - name: clone remote git
repository
      git:
        repo:
https://github.com/sunilkumark11/git
-9am-batch.git
        dest: /home/ubuntu/newgit
```

```
...
```

To check the syntax:

```
$ ansible-playbook  playbook1.yml
--syntax-check
```

(Do not use tab when creating yml file)

To run the playbook

```
$ ansible-playbook  playbook1.yml
-b
```

+++++++++++++++++++++++++++++