[demouser@controller ~]$

[demouser@controller ~]$ ssh-copy-id demouser@172.31.16.187

/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/demouser/.ssh/id\_rsa.

The authenticity of host '172.31.16.187 (172.31.16.187)' can't be established.

ECDSA key fingerprint is SHA256:eYswYlId7zTZuEpAoNrMp48HWu4D5qAo5+Rw7HCkK90.

ECDSA key fingerprint is MD5:a4:05:a4:09:4d:e3:fe:4e:d8:22:2f:30:1e:2f:0b:eb.

Are you sure you want to continue connecting (yes/no)? yes

/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any th

/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it

demouser@172.31.16.187's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'demouser@172.31.16.187'"

and check to make sure that only the key(s) you wanted were added.

[demouser@controller ~]$ ssh 'demouser@172.31.16.187'

Last login: Thu Dec 27 03:52:53 2018

[demouser@remote\_machine1 ~]$

[demouser@remote\_machine1 ~]$

[demouser@remote\_machine1 ~]$

[demouser@remote\_machine1 ~]$ Connection reset by 3.16.167.196 port 22

pb186007@WINPB186007-S9F MINGW64 ~/Desktop

.comh -i "redhat1\_key.pem" ec2-user@ec2-3-16-167-196.us-east-2.compute.amazonaws.

Last login: Thu Dec 27 02:43:02 2018 from nat10.teradata.com

[ec2-user@controller ~]$ sudo -i

[root@controller ~]# cat /etc/redhat-release

Red Hat Enterprise Linux Server release 7.6 (Maipo)

[root@controller ~]# Connection reset by 3.16.167.196 port 22

pb186007@WINPB186007-S9F MINGW64 ~/Desktop

.comh -i "redhat1\_key.pem" ec2-user@ec2-3-16-167-196.us-east-2.compute.amazonaws.

Last login: Thu Dec 27 13:16:16 2018 from nat10.teradata.com

[ec2-user@controller ~]$ sudo -i

[root@controller ~]# clear

[root@controller ~]# ssh 172.31.16.187

root@172.31.16.187's password:

Permission denied, please try again.

root@172.31.16.187's password:

[root@controller ~]# sudo - demouser

sudo: -: command not found

[root@controller ~]# su - demouser

Last login: Thu Dec 27 03:40:35 UTC 2018 on pts/0

[demouser@controller ~]$ ssh 172.31.16.187

Last login: Thu Dec 27 03:57:34 2018 from ip-172-31-24-218.us-east-2.compute.internal

[demouser@remote\_machine1 ~]$

[demouser@remote\_machine1 ~]$

[demouser@remote\_machine1 ~]$ exit

logout

Connection to 172.31.16.187 closed.

[demouser@controller ~]$ cd /etc/ansible

[demouser@controller ansible]$

[demouser@controller ansible]$ pwd

/etc/ansible

[demouser@controller ansible]$

[demouser@controller ansible]$ ls -lrt

total 24

drwxr-xr-x. 2 root root 6 Dec 14 05:57 roles

-rw-r--r--. 1 root root 1016 Dec 14 05:57 hosts

-rw-r--r--. 1 root root 20277 Dec 14 05:57 ansible.cfg

[demouser@controller ansible]$

[demouser@controller ansible]$

[demouser@controller ansible]$ vi hosts

[demouser@controller ansible]$ cp hosts hosts.org

cp: cannot create regular file ‘hosts.org’: Permission denied

[demouser@controller ansible]$ sudo cp hosts hosts.org

[demouser@controller ansible]$

[demouser@controller ansible]$

[demouser@controller ansible]$ ls -lrt

total 28

drwxr-xr-x. 2 root root 6 Dec 14 05:57 roles

-rw-r--r--. 1 root root 1016 Dec 14 05:57 hosts

-rw-r--r--. 1 root root 20277 Dec 14 05:57 ansible.cfg

-rw-r--r--. 1 root root 1016 Dec 28 01:58 hosts.org

[demouser@controller ansible]$ rm hosts

rm: remove write-protected regular file ‘hosts’? ^C

[demouser@controller ansible]$ echo "" > hosts

-bash: hosts: Permission denied

[demouser@controller ansible]$ sudo echo "" > hosts

-bash: hosts: Permission denied

[demouser@controller ansible]$ sudo rm hosts

[demouser@controller ansible]$ sudo vi hosts

[demouser@controller ansible]$

[demouser@controller ansible]$

[demouser@controller ansible]$ sudo cat hosts

[demogroup]

172.31.16.187

[demouser@controller ansible]$

[demouser@controller ansible]$

[demouser@controller ansible]$ ansible demogroup -m ping

172.31.16.187 | SUCCESS => {

"changed": false,

"ping": "pong"

}

[demouser@controller ansible]$

[demouser@controller ansible]$ ansible demogroup -m yum -a "name=httpd action=present

172.31.16.187 | FAILED! => {

"ansible\_facts": {

"pkg\_mgr": "yum"

},

"changed": false,

"msg": "Unsupported parameters for (yum) module: action Supported parameters incluwngrade, autoremove, bugfix, conf\_file, disable\_excludes, disable\_gpg\_check, disable\_plerepo, download\_only, enable\_plugin, enablerepo, exclude, install\_repoquery, installrame, releasever, security, skip\_broken, state, update\_cache, update\_only, use\_backend,rts"

}

[demouser@controller ansible]$ ansible demogroup -m yum -a "name=httpd, action=presen

172.31.16.187 | FAILED! => {

"ansible\_facts": {

"pkg\_mgr": "yum"

},

"changed": false,

"msg": "Unsupported parameters for (yum) module: action Supported parameters incluwngrade, autoremove, bugfix, conf\_file, disable\_excludes, disable\_gpg\_check, disable\_plerepo, download\_only, enable\_plugin, enablerepo, exclude, install\_repoquery, installrame, releasever, security, skip\_broken, state, update\_cache, update\_only, use\_backend,rts"

}

[demouser@controller ansible]$ ansible demogroup -m yum -a "name=httpd state=present"

172.31.16.187 | FAILED! => {

"ansible\_facts": {

"pkg\_mgr": "yum"

},

"changed": false,

"msg": "Failure talking to yum: Could not contact any CDS load balancers: rhui2-cd2.aws.ce.redhat.com, rhui2-cds02.us-east-2.aws.ce.redhat.com."

}

[demouser@controller ansible]$ ansible demogroup -m -b yum -a "name=httpd state=prese

Usage: ansible <host-pattern> [options]

Define and run a single task 'playbook' against a set of hosts

Options:

-a MODULE\_ARGS, --args=MODULE\_ARGS

module arguments

--ask-vault-pass ask for vault password

-B SECONDS, --background=SECONDS

run asynchronously, failing after X seconds

(default=N/A)

-C, --check don't make any changes; instead, try to predict some

of the changes that may occur

-D, --diff when changing (small) files and templates, show the

differences in those files; works great with --check

-e EXTRA\_VARS, --extra-vars=EXTRA\_VARS

set additional variables as key=value or YAML/JSON, if

filename prepend with @

-f FORKS, --forks=FORKS

specify number of parallel processes to use

(default=5)

-h, --help show this help message and exit

-i INVENTORY, --inventory=INVENTORY, --inventory-file=INVENTORY

specify inventory host path or comma separated host

list. --inventory-file is deprecated

-l SUBSET, --limit=SUBSET

further limit selected hosts to an additional pattern

--list-hosts outputs a list of matching hosts; does not execute

anything else

-m MODULE\_NAME, --module-name=MODULE\_NAME

module name to execute (default=command)

-M MODULE\_PATH, --module-path=MODULE\_PATH

prepend colon-separated path(s) to module library

(default=[u'/home/demouser/.ansible/plugins/modules',

u'/usr/share/ansible/plugins/modules'])

-o, --one-line condense output

--playbook-dir=BASEDIR

Since this tool does not use playbooks, use this as a

subsitute playbook directory.This sets the relative

path for many features including roles/ group\_vars/

etc.

-P POLL\_INTERVAL, --poll=POLL\_INTERVAL

set the poll interval if using -B (default=15)

--syntax-check perform a syntax check on the playbook, but do not

execute it

-t TREE, --tree=TREE log output to this directory

--vault-id=VAULT\_IDS the vault identity to use

--vault-password-file=VAULT\_PASSWORD\_FILES

vault password file

-v, --verbose verbose mode (-vvv for more, -vvvv to enable

connection debugging)

--version show program's version number and exit

Connection Options:

control as whom and how to connect to hosts

-k, --ask-pass ask for connection password

--private-key=PRIVATE\_KEY\_FILE, --key-file=PRIVATE\_KEY\_FILE

use this file to authenticate the connection

-u REMOTE\_USER, --user=REMOTE\_USER

connect as this user (default=None)

-c CONNECTION, --connection=CONNECTION

connection type to use (default=smart)

-T TIMEOUT, --timeout=TIMEOUT

override the connection timeout in seconds

(default=10)

--ssh-common-args=SSH\_COMMON\_ARGS

specify common arguments to pass to sftp/scp/ssh (e.g.

ProxyCommand)

--sftp-extra-args=SFTP\_EXTRA\_ARGS

specify extra arguments to pass to sftp only (e.g. -f,

-l)

--scp-extra-args=SCP\_EXTRA\_ARGS

specify extra arguments to pass to scp only (e.g. -l)

--ssh-extra-args=SSH\_EXTRA\_ARGS

specify extra arguments to pass to ssh only (e.g. -R)

Privilege Escalation Options:

control how and which user you become as on target hosts

-s, --sudo run operations with sudo (nopasswd) (deprecated, use

become)

-U SUDO\_USER, --sudo-user=SUDO\_USER

desired sudo user (default=root) (deprecated, use

become)

-S, --su run operations with su (deprecated, use become)

-R SU\_USER, --su-user=SU\_USER

run operations with su as this user (default=None)

(deprecated, use become)

-b, --become run operations with become (does not imply password

prompting)

--become-method=BECOME\_METHOD

privilege escalation method to use (default=sudo),

valid choices: [ sudo | su | pbrun | pfexec | doas |

dzdo | ksu | runas | pmrun | enable | machinectl ]

--become-user=BECOME\_USER

run operations as this user (default=root)

--ask-sudo-pass ask for sudo password (deprecated, use become)

--ask-su-pass ask for su password (deprecated, use become)

-K, --ask-become-pass

ask for privilege escalation password

Some modules do not make sense in Ad-Hoc (include, meta, etc)

ERROR! Extraneous options or arguments

[demouser@controller ansible]$ ansible demogroup -m -b yum -a "name=httpd state=lates

Usage: ansible <host-pattern> [options]

Define and run a single task 'playbook' against a set of hosts

Options:

-a MODULE\_ARGS, --args=MODULE\_ARGS

module arguments

--ask-vault-pass ask for vault password

-B SECONDS, --background=SECONDS

run asynchronously, failing after X seconds

(default=N/A)

-C, --check don't make any changes; instead, try to predict some

of the changes that may occur

-D, --diff when changing (small) files and templates, show the

differences in those files; works great with --check

-e EXTRA\_VARS, --extra-vars=EXTRA\_VARS

set additional variables as key=value or YAML/JSON, if

filename prepend with @

-f FORKS, --forks=FORKS

specify number of parallel processes to use

(default=5)

-h, --help show this help message and exit

-i INVENTORY, --inventory=INVENTORY, --inventory-file=INVENTORY

specify inventory host path or comma separated host

list. --inventory-file is deprecated

-l SUBSET, --limit=SUBSET

further limit selected hosts to an additional pattern

--list-hosts outputs a list of matching hosts; does not execute

anything else

-m MODULE\_NAME, --module-name=MODULE\_NAME

module name to execute (default=command)

-M MODULE\_PATH, --module-path=MODULE\_PATH

prepend colon-separated path(s) to module library

(default=[u'/home/demouser/.ansible/plugins/modules',

u'/usr/share/ansible/plugins/modules'])

-o, --one-line condense output

--playbook-dir=BASEDIR

Since this tool does not use playbooks, use this as a

subsitute playbook directory.This sets the relative

path for many features including roles/ group\_vars/

etc.

-P POLL\_INTERVAL, --poll=POLL\_INTERVAL

set the poll interval if using -B (default=15)

--syntax-check perform a syntax check on the playbook, but do not

execute it

-t TREE, --tree=TREE log output to this directory

--vault-id=VAULT\_IDS the vault identity to use

--vault-password-file=VAULT\_PASSWORD\_FILES

vault password file

-v, --verbose verbose mode (-vvv for more, -vvvv to enable

connection debugging)

--version show program's version number and exit

Connection Options:

control as whom and how to connect to hosts

-k, --ask-pass ask for connection password

--private-key=PRIVATE\_KEY\_FILE, --key-file=PRIVATE\_KEY\_FILE

use this file to authenticate the connection

-u REMOTE\_USER, --user=REMOTE\_USER

connect as this user (default=None)

-c CONNECTION, --connection=CONNECTION

connection type to use (default=smart)

-T TIMEOUT, --timeout=TIMEOUT

override the connection timeout in seconds

(default=10)

--ssh-common-args=SSH\_COMMON\_ARGS

specify common arguments to pass to sftp/scp/ssh (e.g.

ProxyCommand)

--sftp-extra-args=SFTP\_EXTRA\_ARGS

specify extra arguments to pass to sftp only (e.g. -f,

-l)

--scp-extra-args=SCP\_EXTRA\_ARGS

specify extra arguments to pass to scp only (e.g. -l)

--ssh-extra-args=SSH\_EXTRA\_ARGS

specify extra arguments to pass to ssh only (e.g. -R)

Privilege Escalation Options:

control how and which user you become as on target hosts

-s, --sudo run operations with sudo (nopasswd) (deprecated, use

become)

-U SUDO\_USER, --sudo-user=SUDO\_USER

desired sudo user (default=root) (deprecated, use

become)

-S, --su run operations with su (deprecated, use become)

-R SU\_USER, --su-user=SU\_USER

run operations with su as this user (default=None)

(deprecated, use become)

-b, --become run operations with become (does not imply password

prompting)

--become-method=BECOME\_METHOD

privilege escalation method to use (default=sudo),

valid choices: [ sudo | su | pbrun | pfexec | doas |

dzdo | ksu | runas | pmrun | enable | machinectl ]

--become-user=BECOME\_USER

run operations as this user (default=root)

--ask-sudo-pass ask for sudo password (deprecated, use become)

--ask-su-pass ask for su password (deprecated, use become)

-K, --ask-become-pass

ask for privilege escalation password

Some modules do not make sense in Ad-Hoc (include, meta, etc)

ERROR! Extraneous options or arguments

[demouser@controller ansible]$ ansible demogroup -m -b yum -a "name=httpd state=insta

Usage: ansible <host-pattern> [options]

Define and run a single task 'playbook' against a set of hosts

Options:

-a MODULE\_ARGS, --args=MODULE\_ARGS

module arguments

--ask-vault-pass ask for vault password

-B SECONDS, --background=SECONDS

run asynchronously, failing after X seconds

(default=N/A)

-C, --check don't make any changes; instead, try to predict some

of the changes that may occur

-D, --diff when changing (small) files and templates, show the

differences in those files; works great with --check

-e EXTRA\_VARS, --extra-vars=EXTRA\_VARS

set additional variables as key=value or YAML/JSON, if

filename prepend with @

-f FORKS, --forks=FORKS

specify number of parallel processes to use

(default=5)

-h, --help show this help message and exit

-i INVENTORY, --inventory=INVENTORY, --inventory-file=INVENTORY

specify inventory host path or comma separated host

list. --inventory-file is deprecated

-l SUBSET, --limit=SUBSET

further limit selected hosts to an additional pattern

--list-hosts outputs a list of matching hosts; does not execute

anything else

-m MODULE\_NAME, --module-name=MODULE\_NAME

module name to execute (default=command)

-M MODULE\_PATH, --module-path=MODULE\_PATH

prepend colon-separated path(s) to module library

(default=[u'/home/demouser/.ansible/plugins/modules',

u'/usr/share/ansible/plugins/modules'])

-o, --one-line condense output

--playbook-dir=BASEDIR

Since this tool does not use playbooks, use this as a

subsitute playbook directory.This sets the relative

path for many features including roles/ group\_vars/

etc.

-P POLL\_INTERVAL, --poll=POLL\_INTERVAL

set the poll interval if using -B (default=15)

--syntax-check perform a syntax check on the playbook, but do not

execute it

-t TREE, --tree=TREE log output to this directory

--vault-id=VAULT\_IDS the vault identity to use

--vault-password-file=VAULT\_PASSWORD\_FILES

vault password file

-v, --verbose verbose mode (-vvv for more, -vvvv to enable

connection debugging)

--version show program's version number and exit

Connection Options:

control as whom and how to connect to hosts

-k, --ask-pass ask for connection password

--private-key=PRIVATE\_KEY\_FILE, --key-file=PRIVATE\_KEY\_FILE

use this file to authenticate the connection

-u REMOTE\_USER, --user=REMOTE\_USER

connect as this user (default=None)

-c CONNECTION, --connection=CONNECTION

connection type to use (default=smart)

-T TIMEOUT, --timeout=TIMEOUT

override the connection timeout in seconds

(default=10)

--ssh-common-args=SSH\_COMMON\_ARGS

specify common arguments to pass to sftp/scp/ssh (e.g.

ProxyCommand)

--sftp-extra-args=SFTP\_EXTRA\_ARGS

specify extra arguments to pass to sftp only (e.g. -f,

-l)

--scp-extra-args=SCP\_EXTRA\_ARGS

specify extra arguments to pass to scp only (e.g. -l)

--ssh-extra-args=SSH\_EXTRA\_ARGS

specify extra arguments to pass to ssh only (e.g. -R)

Privilege Escalation Options:

control how and which user you become as on target hosts

-s, --sudo run operations with sudo (nopasswd) (deprecated, use

become)

-U SUDO\_USER, --sudo-user=SUDO\_USER

desired sudo user (default=root) (deprecated, use

become)

-S, --su run operations with su (deprecated, use become)

-R SU\_USER, --su-user=SU\_USER

run operations with su as this user (default=None)

(deprecated, use become)

-b, --become run operations with become (does not imply password

prompting)

--become-method=BECOME\_METHOD

privilege escalation method to use (default=sudo),

valid choices: [ sudo | su | pbrun | pfexec | doas |

dzdo | ksu | runas | pmrun | enable | machinectl ]

--become-user=BECOME\_USER

run operations as this user (default=root)

--ask-sudo-pass ask for sudo password (deprecated, use become)

--ask-su-pass ask for su password (deprecated, use become)

-K, --ask-become-pass

ask for privilege escalation password

Some modules do not make sense in Ad-Hoc (include, meta, etc)

ERROR! Extraneous options or arguments

[demouser@controller ansible]$ ansible demogroup -m -b yum -a "name=httpd state=instal

Usage: ansible <host-pattern> [options]

Define and run a single task 'playbook' against a set of hosts

Options:

-a MODULE\_ARGS, --args=MODULE\_ARGS

module arguments

--ask-vault-pass ask for vault password

-B SECONDS, --background=SECONDS

run asynchronously, failing after X seconds

(default=N/A)

-C, --check don't make any changes; instead, try to predict some

of the changes that may occur

-D, --diff when changing (small) files and templates, show the

differences in those files; works great with --check

-e EXTRA\_VARS, --extra-vars=EXTRA\_VARS

set additional variables as key=value or YAML/JSON, if

filename prepend with @

-f FORKS, --forks=FORKS

specify number of parallel processes to use

(default=5)

-h, --help show this help message and exit

-i INVENTORY, --inventory=INVENTORY, --inventory-file=INVENTORY

specify inventory host path or comma separated host

list. --inventory-file is deprecated

-l SUBSET, --limit=SUBSET

further limit selected hosts to an additional pattern

--list-hosts outputs a list of matching hosts; does not execute

anything else

-m MODULE\_NAME, --module-name=MODULE\_NAME

module name to execute (default=command)

-M MODULE\_PATH, --module-path=MODULE\_PATH

prepend colon-separated path(s) to module library

(default=[u'/home/demouser/.ansible/plugins/modules',

u'/usr/share/ansible/plugins/modules'])

-o, --one-line condense output

--playbook-dir=BASEDIR

Since this tool does not use playbooks, use this as a

subsitute playbook directory.This sets the relative

path for many features including roles/ group\_vars/

etc.

-P POLL\_INTERVAL, --poll=POLL\_INTERVAL

set the poll interval if using -B (default=15)

--syntax-check perform a syntax check on the playbook, but do not

execute it

-t TREE, --tree=TREE log output to this directory

--vault-id=VAULT\_IDS the vault identity to use

--vault-password-file=VAULT\_PASSWORD\_FILES

vault password file

-v, --verbose verbose mode (-vvv for more, -vvvv to enable

connection debugging)

--version show program's version number and exit

Connection Options:

control as whom and how to connect to hosts

-k, --ask-pass ask for connection password

--private-key=PRIVATE\_KEY\_FILE, --key-file=PRIVATE\_KEY\_FILE

use this file to authenticate the connection

-u REMOTE\_USER, --user=REMOTE\_USER

connect as this user (default=None)

-c CONNECTION, --connection=CONNECTION

connection type to use (default=smart)

-T TIMEOUT, --timeout=TIMEOUT

override the connection timeout in seconds

(default=10)

--ssh-common-args=SSH\_COMMON\_ARGS

specify common arguments to pass to sftp/scp/ssh (e.g.

ProxyCommand)

--sftp-extra-args=SFTP\_EXTRA\_ARGS

specify extra arguments to pass to sftp only (e.g. -f,

-l)

--scp-extra-args=SCP\_EXTRA\_ARGS

specify extra arguments to pass to scp only (e.g. -l)

--ssh-extra-args=SSH\_EXTRA\_ARGS

specify extra arguments to pass to ssh only (e.g. -R)

Privilege Escalation Options:

control how and which user you become as on target hosts

-s, --sudo run operations with sudo (nopasswd) (deprecated, use

become)

-U SUDO\_USER, --sudo-user=SUDO\_USER

desired sudo user (default=root) (deprecated, use

become)

-S, --su run operations with su (deprecated, use become)

-R SU\_USER, --su-user=SU\_USER

run operations with su as this user (default=None)

(deprecated, use become)

-b, --become run operations with become (does not imply password

prompting)

--become-method=BECOME\_METHOD

privilege escalation method to use (default=sudo),

valid choices: [ sudo | su | pbrun | pfexec | doas |

dzdo | ksu | runas | pmrun | enable | machinectl ]

--become-user=BECOME\_USER

run operations as this user (default=root)

--ask-sudo-pass ask for sudo password (deprecated, use become)

--ask-su-pass ask for su password (deprecated, use become)

-K, --ask-become-pass

ask for privilege escalation password

Some modules do not make sense in Ad-Hoc (include, meta, etc)

ERROR! Extraneous options or arguments

[demouser@controller ansible]$ ansible demogroup -m -b yum -a "name=httpd state=instal

Usage: ansible <host-pattern> [options]

Define and run a single task 'playbook' against a set of hosts

Options:

-a MODULE\_ARGS, --args=MODULE\_ARGS

module arguments

--ask-vault-pass ask for vault password

-B SECONDS, --background=SECONDS

run asynchronously, failing after X seconds

(default=N/A)

-C, --check don't make any changes; instead, try to predict some

of the changes that may occur

-D, --diff when changing (small) files and templates, show the

differences in those files; works great with --check

-e EXTRA\_VARS, --extra-vars=EXTRA\_VARS

set additional variables as key=value or YAML/JSON, if

filename prepend with @

-f FORKS, --forks=FORKS

specify number of parallel processes to use

(default=5)

-h, --help show this help message and exit

-i INVENTORY, --inventory=INVENTORY, --inventory-file=INVENTORY

specify inventory host path or comma separated host

list. --inventory-file is deprecated

-l SUBSET, --limit=SUBSET

further limit selected hosts to an additional pattern

--list-hosts outputs a list of matching hosts; does not execute

anything else

-m MODULE\_NAME, --module-name=MODULE\_NAME

module name to execute (default=command)

-M MODULE\_PATH, --module-path=MODULE\_PATH

prepend colon-separated path(s) to module library

(default=[u'/home/demouser/.ansible/plugins/modules',

u'/usr/share/ansible/plugins/modules'])

-o, --one-line condense output

--playbook-dir=BASEDIR

Since this tool does not use playbooks, use this as a

subsitute playbook directory.This sets the relative

path for many features including roles/ group\_vars/

etc.

-P POLL\_INTERVAL, --poll=POLL\_INTERVAL

set the poll interval if using -B (default=15)

--syntax-check perform a syntax check on the playbook, but do not

execute it

-t TREE, --tree=TREE log output to this directory

--vault-id=VAULT\_IDS the vault identity to use

--vault-password-file=VAULT\_PASSWORD\_FILES

vault password file

-v, --verbose verbose mode (-vvv for more, -vvvv to enable

connection debugging)

--version show program's version number and exit

Connection Options:

control as whom and how to connect to hosts

-k, --ask-pass ask for connection password

--private-key=PRIVATE\_KEY\_FILE, --key-file=PRIVATE\_KEY\_FILE

use this file to authenticate the connection

-u REMOTE\_USER, --user=REMOTE\_USER

connect as this user (default=None)

-c CONNECTION, --connection=CONNECTION

connection type to use (default=smart)

-T TIMEOUT, --timeout=TIMEOUT

override the connection timeout in seconds

(default=10)

--ssh-common-args=SSH\_COMMON\_ARGS

specify common arguments to pass to sftp/scp/ssh (e.g.

ProxyCommand)

--sftp-extra-args=SFTP\_EXTRA\_ARGS

specify extra arguments to pass to sftp only (e.g. -f,

-l)

--scp-extra-args=SCP\_EXTRA\_ARGS

specify extra arguments to pass to scp only (e.g. -l)

--ssh-extra-args=SSH\_EXTRA\_ARGS

specify extra arguments to pass to ssh only (e.g. -R)

Privilege Escalation Options:

control how and which user you become as on target hosts

-s, --sudo run operations with sudo (nopasswd) (deprecated, use

become)

-U SUDO\_USER, --sudo-user=SUDO\_USER

desired sudo user (default=root) (deprecated, use

become)

-S, --su run operations with su (deprecated, use become)

-R SU\_USER, --su-user=SU\_USER

run operations with su as this user (default=None)

(deprecated, use become)

-b, --become run operations with become (does not imply password

prompting)

--become-method=BECOME\_METHOD

privilege escalation method to use (default=sudo),

valid choices: [ sudo | su | pbrun | pfexec | doas |

dzdo | ksu | runas | pmrun | enable | machinectl ]

--become-user=BECOME\_USER

run operations as this user (default=root)

--ask-sudo-pass ask for sudo password (deprecated, use become)

--ask-su-pass ask for su password (deprecated, use become)

-K, --ask-become-pass

ask for privilege escalation password

Some modules do not make sense in Ad-Hoc (include, meta, etc)

ERROR! Extraneous options or arguments

[demouser@controller ansible]$ ansible demogroup -m yum -a "name=httpd state=installed

172.31.16.187 | FAILED! => {

"ansible\_facts": {

"pkg\_mgr": "yum"

},

"changed": false,

"msg": "Failure talking to yum: Could not contact any CDS load balancers: rhui2-cd2.aws.ce.redhat.com, rhui2-cds02.us-east-2.aws.ce.redhat.com."

}

[demouser@controller ansible]$ ansible demogroup -m yum -a "name=httpd state=installed

[demouser@controller ansible]$ pwd

/etc/ansible

[demouser@controller ansible]$

[demouser@controller ansible]$ sudo vi playbook1.yaml

[demouser@controller ansible]$ sudo cat playbook1.yaml

---

- hosts: demogroup

tasks:

- name: Installing httpd

yum:

name: httpd

state: present

[demouser@controller ansible]$

[demouser@controller ansible]$

[demouser@controller ansible]$ ansible-playbook playbook1.yaml

PLAY [demogroup] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [172.31.16.187]

TASK [Installing httpd] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

fatal: [172.31.16.187]: FAILED! => {"changed": false, "msg": "Failure talking to yum: ntact any CDS load balancers: rhui2-cds01.us-east-2.aws.ce.redhat.com, rhui2-cds02.us-e.redhat.com."}

[WARNING]: Could not create retry file '/etc/ansible/playbook1.retry'. [Errno

Permission denied: u'/etc/ansible/playbook1.retry'

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

172.31.16.187 : ok=1 changed=0 unreachable=0 failed=1

[demouser@controller ansible]$ ansible-playbook -b playbook1.yaml

PLAY [demogroup] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [172.31.16.187]

TASK [Installing httpd] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [172.31.16.187]

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

172.31.16.187 : ok=2 changed=1 unreachable=0 failed=0

[demouser@controller ansible]$

[demouser@controller ansible]$

[demouser@controller ansible]$ pwd

/etc/ansible

[demouser@controller ansible]$ ls -lrt

total 32

drwxr-xr-x. 2 root root 6 Dec 14 05:57 roles

-rw-r--r--. 1 root root 20277 Dec 14 05:57 ansible.cfg

-rw-r--r--. 1 root root 1016 Dec 28 01:58 hosts.org

-rw-r--r--. 1 root root 26 Dec 28 02:02 hosts

-rw-r--r--. 1 root root 108 Dec 28 02:32 playbook1.yaml

[demouser@controller ansible]$

[demouser@controller ansible]$ sudo vi playbook1.yaml

[demouser@controller ansible]$ sudo cat playbook1.yaml

---

- hosts: demogroup

tasks:

- name: Installing httpd

yum:

name: httpd

state: present

- name: starting httpd

service:

name: httpd

state: started

[demouser@controller ansible]$

[demouser@controller ansible]$

[demouser@controller ansible]$

[demouser@controller ansible]$ ansible-playbook -b playbook1.yaml

PLAY [demogroup] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [172.31.16.187]

TASK [Installing httpd] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [172.31.16.187]

TASK [starting httpd] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [172.31.16.187]

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

172.31.16.187 : ok=3 changed=1 unreachable=0 failed=0

[demouser@controller ansible]$

[demouser@controller ansible]$

[demouser@controller ansible]$ ansible-playbook -b playbook1.yaml

PLAY [demogroup] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [172.31.16.187]

TASK [Installing httpd] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [172.31.16.187]

TASK [starting httpd] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [172.31.16.187]

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

172.31.16.187 : ok=3 changed=0 unreachable=0 failed=0

[demouser@controller ansible]$ ls -lrt

total 32

drwxr-xr-x. 2 root root 6 Dec 14 05:57 roles

-rw-r--r--. 1 root root 20277 Dec 14 05:57 ansible.cfg

-rw-r--r--. 1 root root 1016 Dec 28 01:58 hosts.org

-rw-r--r--. 1 root root 26 Dec 28 02:02 hosts

-rw-r--r--. 1 root root 184 Dec 28 02:41 playbook1.yaml

[demouser@controller ansible]$ vi playbook1.yaml

[demouser@controller ansible]$ vi hosts

[demouser@controller ansible]$ sudo vi hosts

[demouser@controller ansible]$ ansible-playbook -b playbook1.yaml

PLAY [demogroup] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [172.31.16.187]

fatal: [172.31.0.0]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host 172.31.0.0 port 22: Connection timed out\r\n", "unreachable": true}

TASK [Installing httpd] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [172.31.16.187]

TASK [starting httpd] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [172.31.16.187]

[WARNING]: Could not create retry file '/etc/ansible/playbook1.retry'.

[Errno 13] Permission denied: u'/etc/ansible/playbook1.retry'

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

172.31.0.0 : ok=0 changed=0 unreachable=1 failed=0

172.31.16.187 : ok=3 changed=0 unreachable=0 failed=0

[demouser@controller ansible]$

[demouser@controller ansible]$

[demouser@controller ansible]$ pwd

/etc/ansible

[demouser@controller ansible]$

[demouser@controller ansible]$

[demouser@controller ansible]$ ls -lrt

total 32

drwxr-xr-x. 2 root root 6 Dec 14 05:57 roles

-rw-r--r--. 1 root root 20277 Dec 14 05:57 ansible.cfg

-rw-r--r--. 1 root root 1016 Dec 28 01:58 hosts.org

-rw-r--r--. 1 root root 184 Dec 28 02:41 playbook1.yaml

-rw-r--r--. 1 root root 37 Dec 28 02:55 hosts

[demouser@controller ansible]$ vi index.html

[demouser@controller ansible]$ sudo vi index.html

[demouser@controller ansible]$ pwd

/etc/ansible

[demouser@controller ansible]$ vi playbook1.yaml

[demouser@controller ansible]$ sudo vi playbook1.yaml

[demouser@controller ansible]$ ansible-playbook -b playbook1.yaml

PLAY [demogroup] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [172.31.16.187]

fatal: [172.31.0.0]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host 172.31.0.0 port 22: Connection timed out\r\n", "unreachable": true}

TASK [Installing httpd] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [172.31.16.187]

TASK [starting httpd] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [172.31.16.187]

TASK [example copying file with owner and permissions] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [172.31.16.187]

[WARNING]: Could not create retry file '/etc/ansible/playbook1.retry'.

[Errno 13] Permission denied: u'/etc/ansible/playbook1.retry'

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

172.31.0.0 : ok=0 changed=0 unreachable=1 failed=0

172.31.16.187 : ok=4 changed=1 unreachable=0 failed=0

[demouser@controller ansible]$

[demouser@controller ansible]$

[demouser@controller ansible]$