Ansible  – “**setup”** module is responsible to gather facts of the remote hosts. The system facts are nothing but the system configuration which includes the hostname, IP address, filesystems, OS releases, Users, Network parameters, CPU, memory and many more. This module is automatically included in all the playbooks to gather useful variables which can be used to create the dynamic inventory or perform the specific tasks. There is a way to write the custom facts about the hosts to filter further. Let’s start exploring it.

**Assumptions**:

* Ansible Server : ansible-server
* Remote host: gpfslinapp1

**Ansible – SETUP module:**

1. Here is the temporary inventory file.

[linadm@ansible-server automation]$ cat lin-servers.1

gpfslinapp1

[linadm@ansible-server automation]$

2. To get the facts of the remote hosts, use the following command.

[linadm@ansible-server automation]$ ansible -m setup -i lin-servers.1 all

gpfslinapp1 | SUCCESS => {

"ansible\_facts": {

"ansible\_all\_ipv4\_addresses": [

"192.168.3.151"

],

"ansible\_all\_ipv6\_addresses": [

"fe80::5af3:5374:a618:9c07"

],

"ansible\_apparmor": {

"status": "disabled"

<<>>

3. If you would like to run the remote facts on the specific hosts from the inventory, use the following command.

[linadm@ansible-server automation]$ ansible -m setup -i lin-servers.1 gpfslinapp1

gpfslinapp1 | SUCCESS => {

"ansible\_facts": {

"ansible\_all\_ipv4\_addresses": [

"192.168.3.151"

],

"ansible\_all\_ipv6\_addresses": [

"fe80::5af3:5374:a618:9c07"

],

"ansible\_apparmor": {

"status": "disabled"

},

"ansible\_architecture": "x86\_64",

"ansible\_bios\_date": "05/20/2014",

"ansible\_bios\_version": "6.00",

"ansible\_cmdline": {

"BOOT\_IMAGE": "/vmlinuz-3.10.0-862.3.3.el7.x86\_64",

"LANG": "en\_US.UTF-8",

**“setup” Module – Use Filter in Adhoc Mode:**

1. Limit the setup module facts, use “filter” option. The following command filters the currently mounted filesystems on the remote host.

[linadm@ansible-server automation]$ ansible -m setup -i lin-servers.1 gpfslinapp1 -a 'filter=ansible\_mounts'

gpfslinapp1 | SUCCESS => {

"ansible\_facts": {

"ansible\_mounts": [

{

"block\_available": 4250037,

"block\_size": 4096,

"block\_total": 4638976,

"block\_used": 388939,

"device": "/dev/sda3",

"fstype": "xfs",

"inode\_available": 9240033,

"inode\_total": 9283072,

"inode\_used": 43039,

"mount": "/",

"options": "rw,relatime,attr2,inode64,noquota",

"size\_available": 17408151552,

"size\_total": 19001245696,

"uuid": "1abdbd4f-d020-4b23-a68d-5518b98e7ec0"

},

{

"block\_available": 36413,

"block\_size": 4096,

"block\_total": 75945,

"block\_used": 39532,

"device": "/dev/sda1",

"fstype": "xfs",

"inode\_available": 153266,

"inode\_total": 153600,

"inode\_used": 334,

"mount": "/boot",

"options": "rw,relatime,attr2,inode64,noquota",

"size\_available": 149147648,

"size\_total": 311070720,

"uuid": "ebf6278a-2098-493e-8e3a-7f7d8d4841d5"

}

]

},

"changed": false

}

[linadm@ansible-server automation]$

2. To filter the remote host’s OS distribution, use the following ansible variable.

[linadm@ansible-server automation]$ ansible -m setup -i lin-servers.1 gpfslinapp1 -a 'filter=ansible\_distribution'

gpfslinapp1 | SUCCESS => {

"ansible\_facts": {

"ansible\_distribution": "CentOS"

},

"changed": false

}

[linadm@ansible-server automation]$

### ****Classifying the hosts using the “setup” Module:****

1.  To find out the running kernel on all the hosts, use the following command.

[linadm@ansible-server automation]$ ansible -m setup -i lin-servers.1 all -a 'filter="ansible\_kernel"' -o

gpfslinapp1 | SUCCESS => {"ansible\_facts": {"ansible\_kernel": "3.10.0-862.3.3.el7.x86\_64"}, "changed": false}

uaans69 | SUCCESS => {"ansible\_facts": {"ansible\_kernel": "2.6.32-696.20.1.el6.x86\_64"},"changed": false}

[linadm@ansible-server automation]$

**Note**: “uaans69” – host has been added into the inventory.