1. How can I set the PATH variable for a task or any other environment variable?

Setting environment variables can be done with the *environment* keyword. It can be used at the task or the play level:

environment:

PATH: "**{{** ansible\_env.PATH **}}**:/thingy/bin"

SOME: value

1. How can encrypted passwords for the user module be generated?

Ansible has a very simple ad-hoc command for this

ansible all -i localhost, -m debug -a "msg={{ 'mypassword' | password\_hash('sha512', 'mysecretsalt') }}"

We can also use the Passlib library of Python, e.g

python -c "from passlib.hash import sha512\_crypt; import getpass; print(sha512\_crypt.using(rounds=5000).hash(getpass.getpass()))"

On top of this, we should also avoid storing raw passwords in playbook or host\_vars, instead, we should use integrated methods to generate a hash version of a password.

1. How can a list of hosts in a group be looped over within a template?

This can be done by accessing the “$groups” dictionary in the template, like so:

{% for host in groups['db\_servers'] %}

{{ host }}

{% endfor %}

If we need to access facts also we need to make sure that the facts have been populated. For instance, a play that talks to db\_servers:

- hosts: db\_servers

tasks:

- debug: msg="Something to debug"

Now, this can be used within a template, like so:

{% for host in groups['db\_servers'] %}

{{ hostvars[host]['ansible\_eth0']['ipv4']['address'] }}

{% endfor %}.

1. What is the best way to see all of my host's inventory variables?

By running the following command, you can see vars resulting from what you’ve defined in the inventory:

ansible -m debug -a "var=hostvars['hostname']" localhost

1. How should I set up a jump host to connect to servers that I don't have direct access to?

First, we need to set a ProxyCommand in ansible\_ssh\_common\_args inventory variable, since any arguments specified in this variable are added to the sftp/scp/ssh command line when connecting to the relevant host(s). For example

[gatewayed]

staging1 ansible\_host=10.0.2.1

staging2 ansible\_host=10.0.2.2 To create a jump host for these we need to add a command in ansible\_ssh\_common\_args

ansible\_ssh\_common\_args: '-o ProxyCommand="ssh -W %h:%p -q user@devopsschool.com"'

In this way whenever we will try to connect to any host in the gatewayed group ansible will append these arguments to the command line.

1. How do you deal with different machines that require different user accounts or ports to log in?

Setting inventory variables in the inventory file is the easiest way.

**Note**

Ansible 2.0 has deprecated the “ssh” from ansible\_ssh\_user, ansible\_ssh\_host, and ansible\_ssh\_port to become ansible\_user, ansible\_host, and ansible\_port. If you are using a version of Ansible prior to 2.0, you should continue using the older style variables (ansible\_ssh\_\*). These shorter variables are ignored, without warning, in older versions of Ansible.

For instance, suppose these hosts have different usernames and ports:

**[webservers]**

asdf.example.com ansible\_port**=**5000 ansible\_user=alice

jkl.example.com ansible\_port**=**5001 ansible\_user=bob

You can also dictate the connection type to be used, if you want:

**[testcluster]**

localhost ansible\_connection**=**local

/path/to/chroot1 ansible\_connection**=**chroot

foo.example.com ansible\_connection**=**paramiko

You may also wish to keep these in group variables instead, or file them in a group\_vars/<groupname> file. See the rest of the documentation for more information about how to organize variables.

1. Assume you're configuring the production environment with Ansible, and your playbook uses an encrypted file. Passwords must be entered while using encrypted files. Can this process be automated, though, given Ansible is utilized for automation?

Yes this process can be automated.

To automate password input we can have a password file for all the passwords of encrypted files will be saved and ansible can make a call to fetch those when required.

ansible\_ssh\_common\_args: '-o ProxyCommand="ssh -W %h:%p -q user@gateway.example.com"'

This can also be achieved by having a separate script that specifies the passwords. But in this case, we need to print a password to stdout to work without annoying errors.

ansible-playbook launch.yml –vault-password-file ~/ .vault\_pass.py

8. Is Ansible a free and open-source software?

Yes, Ansible is an open-source tool that is a powerful automation software tool that one can use.