Three steps to Ansible enlightenment

Contents

1 Use ansible to run shell commands **2**

2 Using ansible modules **2**

2.1 Install a package 2

2.2 Copy a file 2

2.3 Create a directory 3

3 Writing playbooks **3**

4 Optional Guru level: Roles **3**

Can you use ansible to apply updates to all your VMs? Yes you can! You can start simple and grow as your experience develops.

Edit the file `/etc/ansible/hosts` as root (remember `sudo`) and add all your VMs under a group heading, like this:

```
[vms]
host1.ws.nsrc.org
host2.ws.nsrc.org
host3.ws.nsrc.org
```

or the short form:

```
[vms]
host[1:3].ws.nsrc.org
```

1 Use ansible to run shell commands

```
# ansible vms [-u sysadm] -skK -m shell -a 'apt-get install -y bash'
```

Here is an explanation of the flags

`-u sysadm` login as user `sysadm` (only required if the login username is different from your local username)

`-s` use sudo to get root on the remote machine

`-k` prompt for login password

`-K` prompt for sudo password (just hit Enter to use the same password)

`-m shell` invoke the shell module

`-a '...'` arguments to the shell module (i.e. which shell command to run)

This is very easy, but it will not work if the remote command prompts for input. (The flag `-y` to apt-get assumes "yes" to confirmation prompts)

To specify a different inventory file use the `-i` option, e.g. `-i myhosts`

2 Using ansible modules

Your next step is to start learning the specific ansible modules for each system administration task. Here are some common ones.

2.1 Install a package

Use the apt module to install packages on Ubuntu and Debian hosts.

```
# ansible vms [-u sysadm] -skK -m apt -a 'pkg=bash state=present'
```

2.2 Copy a file

Use the copy module. Example: to copy a file `foo` from the current directory to `/etc/foo` on the target hosts, and set the mode to make it world-readable:

```
# ansible vms [-u sysadm] -skK -m copy -a 'src=foo dest=/etc/foo mode=644'
```

2.3 Create a directory

The file module can set the ownership and mode of files, and also create directories.

```
# ansible vms [-u sysadm] -skK -m file -a 'path=/etc/bar state=directory'
```

Use `state=absent` to delete a file or a directory and its contents.

3 Writing playbooks

Once you know the ansible module and arguments, you can put them into a playbook, e.g. `foo.yml`

```
- hosts:
    - vms
  tasks:
    - apt: pkg=bash state=installed
```

and run it:

```
# ansible-playbook [-u sysadm] -skK foo.yml
```

Now you can make a single playbook which performs multiple tasks.

4 Optional Guru level: Roles

Roles make your tasks easier to re-use in multiple playbooks. Create a directory `roles`, inside that a subdirectory for the role name (say "bar"), and inside that a subdirectory `tasks`

Now create file `roles/bar/tasks/main.yml`

```
- apt: pkg-bash state=installed
```

Then foo.yml becomes:

```
- hosts:
    - vms
  roles:
    - bar
```

Any files you want to copy can be stored in `roles/bar/files/...`