Ansible Cheat Sheet

By Seth Kenlon

Ansible provides simple but powerful automation for cross-platform computer support. Ansible playbooks are written in YAML, and executed either locally or remotely.

Command	
`ansible-playbook file.yaml`	Run an Ansible playbook called **file.yaml**
Authentication options	
`--user, -u <username>`	Log in as username
`--private-key, --key-file <key>`	Log in using SSH key (usually in ~/.ssh)
`--ssh-extra-args`	Pass extra command options to SSH
`--vault-id <id>`	Use vault identity **ID**
`--vault-password-file <key>`	Use vault password file **key**
`--ask-vault-pass`	Prompt for a vault password
`--become`	Escalate privileges
`--ask-become-pass`	Prompt for a password for **become**
`--become-method`	Escalate privilege using a specific method
`ansible-doc --type foo --list`	List choices for **become**, **connection**, and other Ansible options
Control options	
`--syntax-check`	Verify syntax of a playbook, but do not run it
`--list-hosts`	Show hosts listed in a playbook
`--list-tasks`	Show tasks defined in a playbook
`--start-at-task <task_name>`	Run playbook starting at **task name**
`--check`	Run the playbook but don't make changes
`--diff`	Show diffs for what changes are made
`--module-path`	Prepend colon-separated path to default path
`--connection <method>`	Connect over **method**

Playbook and YAML

```
Parameter: value
```
A YAML **mapping** entry is a key and a value

```
- foo
- bar
- baz
```
A YAML **sequence** entry is an itemized list

```
Distro:
   - Fedora
   - RHEL
   - Debian
   - Slackware
```
A **mapping** entry can contain a **sequence**

```
OS:
   - Linux: Fedora
   - BSD: NetBSD
```
Sequence items can contain **mappings**

Playbook structure

```
---
```
YAML files start with three dashes

```
- name: "My play"
```
Use the **name** mapping to name your play

```
   hosts: all
```
Indent, and define which hosts the play runs on. List target hosts in **etc/ansible/hosts**

```
   tasks:
```
Open a **tasks** mapping, which will contain a **sequence**

```
  - name: "My task"
```
Give the task a name with the **name** mapping

```
    some_module:
```
Import a module as a new **mapping** containing a **sequence** of parameters. Find required and optional parameters in the module's documentation.

```
      path: '/example/'
```
Parameters are usually **mappings** using the command option as the **key** and an argument as the **value**.

```
  - name: "My other task"
```
A play may contain more than one task

```
    other_module:
      foo: 'bar'
```
A task usually imports a module