



Introduction to Ansible

Jan 12th 2017

www.quoininc.com

Boston

Charlotte

New York

Washington DC

Managua

What is Ansible

- Automation engine
 - Cloud provisioning
 - Configuration management
 - Application development
 - And more
- Infrastructure as code

Competition

- Chef
- Puppet
- CFEngine

Architecture

- One **control** node
 - e.g. your laptop
- Many **managed** nodes
 - e.g. your web and database servers
- Ansible must be installed on the control node
- Managed nodes only require Python and SSH
- The control node automatically deploys any required Python modules to the managed nodes

Installation

- A virtualenv is recommended
- Install by pip

```
$ virtualenv virtualenv
```

```
$ . virtualenv/bin/activate
```

```
(virtualenv)$ pip install ansible
```

- Better to use a requirements.txt

```
(virtualenv)$ pip install -r requirements.txt
```

Inventory

- List of the managed nodes, their groups, and properties
- Simple `.ini` text file
- Dynamic inventory
 - e.g. automatically collect the list of nodes via the AWS management API

Inventory

- Hosts
- Groups
- Variables
 - Host variables
 - Group variables
 - Can be inline in the inventory or placed in external file YAML files
 - Available for use in playbooks, roles, templates, etc.

Ad-hoc Commands

- Using our new inventory file we can ping our managed nodes

```
(virtualenv)$ ansible all \  
    -i inventory/brooklyn.ini \  
    -m ping
```

- **Collecting facts**

```
(virtualenv)$ ansible all \  
    -i inventory/brooklyn.ini \  
    -m setup
```


Playbooks

- A playbook is the main configuration file in Ansible
- YAML
- Lists the operations that will be performed against all or part of the inventory
 - Roles
 - Tasks
 - Handlers

Roles

- You could have one gigantic playbook
- Better to split tasks and handlers out into roles
- Reusable
 - Ansible Galaxy
- Roles can also express dependencies

Tasks

- A task executes a module with some arguments
- A module performs some operation on a managed node based on the task's arguments
- Idempotent

Tasks

- Ansible comes with a large library of modules
 - Copying, editing, or templating a file
 - Configuring users
 - Shell commands
 - Many more
- You can extend that library
 - Python
 - Other languages

Tasks

- Jinja2 templates are available for task **values**
 - Control structures work, but only within a value
- Ansible has its own control structures
 - when
 - with_items, with_files, etc.
 - Quirky
 - You can't use a with_items loop over a list of tasks
 - Unless you split that list into a separate file

Handlers

- Modules should be idempotent
- Actions need to be performed after configuration changes
 - e.g. restart httpd
- Handlers execute a module after a task changes some configuration
- Tasks 'notify' handlers
- A handler runs once even if it has been notified by multiple tasks

Dependencies

- Roles can express dependencies
- Additional roles can be included in a play even though they aren't explicitly listed in the main playbook
- Can be included conditionally or always

Running a Playbook

```
(virtualenv)$ ansible-playbook\  
-i inventory/brooklyn.ini \  
site.yml
```


Limiting the Inventory

```
(virtualenv)$ ansible-playbook \  
-i inventory/brooklyn.ini \  
-l brooklyn-fw \  
site.yml
```

```
(virtualenv)$ ansible-playbook \  
-i inventory/brooklyn.ini \  
-l @site.retry \  
site.yml
```

Limiting the Tasks

```
(virtualenv)$ ansible-playbook \  
-i inventory/brooklyn.ini \  
-t pf \  
site.yml
```

```
(virtualenv)$ ansible-playbook \  
-i inventory/brooklyn.ini \  
-skip-tags pf \  
site.yml
```