Introduction to Ansible

By @colinfdrake

What is it?

```
$ man ansible

ansible – run a command somewhere else
```

But Really

> App deployment, configuration management and orchestration - all from one system. Ansible is powerful IT automation that you can learn quickly.

Why?

- Provision a *consistent* build environment
  - Build nodes/servers you can depend on
  - Or even new developer machines!
- Or just automate any task across servers
  - Discover servers of type X, run command Y on each
- Create/provision a new VM, make a production deployment
- Make a production deployment on an existing server (idempotency)

Basic Concepts

- Inventory
- Facts
- Tasks
- Playbooks
- Roles

Inventory

- The servers to operate on, may be grouped
- Operated on via local login or SSH, no "Ansible server" needed
- Can also by dynamically discovered (built-in AWS discovery)

```
# Typical inventory file, predefined list

[local]
127.0.0.1

[remote]
buildserver1.ourcompany.com
buildserver2.ourcompany.com
```

Facts

- Describe the state of the system
- Collected for each node in the inventory
- Ansible provides many built-ins
  - `ansible_os_family`, `ansible_env`, ...
- Define custom facts as the output of commands
- Used as variables in tasks, file templates, etc.

Tasks

- Many built-in ways to run tasks:
  - `shell`, `apt`, `service`, `copy`, ...
- Declarative: usually describes the desired state of the system
  - Ex: `apt: package="vim" state="installed"`
- Should be idempotent
  - i.e. won't run `service start` if service is already running
  - Facts may be used to determine if a task needs to be run

Playbooks

- Defines which tasks to run on which nodes in the inventory
- May configure static variables

```
# Playbook example (YAML)

- hosts: all
  vars:
    text: "Hello World"
  tasks:
    - shell: echo "{{ text }}"
    - gem: name="rake" state="installed"

- hosts: local
  tasks:
    - shell: echo "This is only locally run"
```

Roles

- Help abstract and modularize Playbooks
- Break up steps, variable definitions, etc. into a role that can be executed by name
- Ansible Galaxy is the repository of community maintained roles
- Leverage these to make life less bad

```
# To execute roles in a Playbook...

roles:
  - TupleAustin.xcode
  - TupleAustin.xcode-cli
  - TupleAustin.homebrew-ruby  # open sourced!
```

Example Project Structure

```
inventory           # Inventory file
requirements.yml    # List of Ansible Galaxy packages
main.yml            # Main playbook
vars/
  main.yml          # Variables associated with playbook
roles/
  some-role/        # A custom role, similar structure
    defaults/
      main.yml
    tasks/
      main.yml
  some-other-role/
    ...
```

Live Demo