Ansible Bootcamp
Learning Goals

• Explain what Ansible is (What)
• Describe Ansible use cases (Why)
• Identify use cases and describe the solutions Ansible provide (When)
• Know the components of Ansible (How)
Ansible

• Ansible is for:
  • Application Deployment
  • Multi-Tier Orchestration
  • Configuration Management

• Why Ansible? Ansible is:
  • Simple
  • Easy to write, read, maintain and evolve - without writing scripts or custom code
  • Fast to learn and set up
Ansible : Use Cases

- Remote execution
- Configuration Management
- Deployment and Orchestration
Ansible: Remote Execution

- Replacement for traditional systems administration tasks
- Checking system responsiveness and uptime
- Gathering information about a collection of systems
- Replace one off rsync scripts, fabric, or terminal multiplexing
Ansible: Configuration Management

• Lengthy, fine-grained system configuration,
  • e.g. adding users, ssh keys, installing and configuring services, and bootstrapping systems to a given state.

• "Configuration remediation" – ensure consistent server configuration.
  • removes manual configuration errors
  • Checks for "Configuration Drift"

• Version-controlled – service configurations can be reliably reproduced.
Ansible: Deployment and Orchestration

• Separate physical infrastructure from tasks
• Allow model-based description of services
• Good for complex, interdependent service deployment
Aspects of Ansible

• Simple connection model:
  • No master-slave relationships
  • Everything can configure everything (even self-configuration)
  • No custom agent to set up
  • Security provided by SSH
  • Simplest mode is push from control node to controlled nodes
Vocabulary

- **Inventory:**
  - Hosts
  - Groups

- **Execution:**
  - Tasks
  - Plays
  - Playbooks

- **Batteries:**
  - Modules
  - Library
Vocabulary: Inventory

• Hosts
  • A remote machine that Ansible manages
  • Can have individual variables (host name, connection port number, etc)

• Groups
  • Easy way to combine hosts with similar aspects
  • All hosts in a group share group variables

• Inventory
  • Fill description of hosts which you own and control
  • Can nest groups and move hosts in and out of groups
Vocabulary: Execution

• Playbook
  • How Ansible orchestrates, configures, administers or deploys systems
  • A combined, coherent description of a complex state of applications, hosts, services.

• Plays
  • A mapping between a set of hosts and the tasks which run on those hosts
  • defines the role that those systems will perform.
  • There can be one or many plays in a playbook

• Tasks: Application of a single module on the specified host(s)
Vocabulary: Batteries

• Modules:
  • units of work that Ansible ships out to remote machines.
  • can be implemented in any language.
  • Return JSON or simple `key=value` pairs.
  • Once modules are executed on remote machines, they are removed, → no long running daemons used

• Library:
  • Collection of available modules
Architecture

http://slides.com/racku/ansible-fundamentals#/4/17
Not Covered Yet

- Variables
- Filters
- Conditionals
- Loops
Ansible: Footprint

- Control machine:
  * *NIX (no Windows yet)
  * (Optional) SSH Client
  * Python 2.6
  * Since 2.0 dependencies are maintained with pip

- Managed node:
  * OpenSSH server
  * > Python-2.4
Hands-on session
Install Ansible
Ansible Ad-Hoc

1. Create an inventory of localhost
2. Collect facts
3. Ensure that there is an Ansible user on the host
4. Ensure that ntpd is installed