

CS 447/647

Configuration Management

Overview

What is configuration management?

What programming paradigm does configuration management use?

What are the elements of configuration management?

What are some common configuration management tasks?

Configuration Management

- Changes should be
 - Structured
 - Automated
 - Consistent
- Difficult when dealing with heterogeneity
 - Debian, Ubuntu, CentOS
 - Windows: 10, Server 2012, Server 2019

Configuration Management (CM)

- Automates the management of Operating Systems
 - Eliminate toil
- Network-based
 - SSH - **ansible**
 - Client - Salt, Puppet, Chef
- Text File Configurations
 - Ansible, Salt - Yet Another Markup Language (YAML)
 - Puppet - Ruby syntax
 - Chef - IDK
- Developer Operations
 - Agile-like development process
 - Less hardware
 - More software engineering
 - Managing cloud resources
 - CLAMS - Culture, Lean, Automation, Measurement, Sharing

Configuration Management in a Nutshell

- Tradition Method
 - Shell scripts
 - Chaos - (Less time and micromanagey)
 - Procedural - violates a core Unix programming principle
- Configuration Management Method
 - Capture state in code
 - Track changes in revision control (git)
 - Declarative - Describe the state
 - Task - create user, install package, copy files, modify configuration.

Dangers of Configuration Management

- No standardization
 - Lexicons differ
 - Puppet - Agent node
 - Ansible - Host
 - Knowledge doesn't transfer
- Don't mix CM with ad-hoc
 - Snowflake systems
- Steep learning curves
 - Especially with large "code" bases
- Test in production
- Slows down development initially

Elements of Configuration Management

- Operations and Parameters
 - Small-scale tasks and checks
 - Out-of-the-box tasks
 - Create or remove users
 - Copy files
 - Render templates
 - Add lines to config files
 - Restart services
 - Run shell commands
 - Applied repeatedly without causing problems
 - Detect state
 - Cross-platform

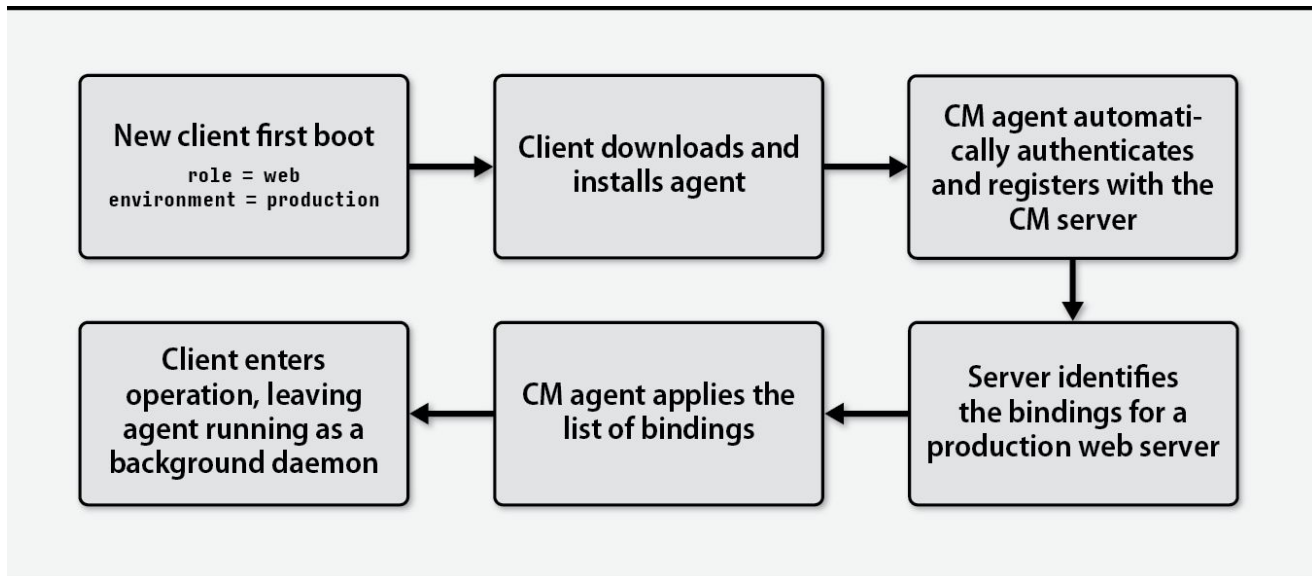
Elements of Configuration Management

- Variables
 - Used in configuration templates
 - Different scopes
 - Types
 - Scalars
 - Arrays
 - Dictionaries
- Facts
 - Discovers descriptive facts of client
 - IP
 - OS
 - Extensible

Elements of Configuration Management

- Change Handlers
 - Operations to perform after a change
 - Restart web server after config change
- Bindings
 - Associate tasks to specific Hosts
- Bundles
 - Collection of operations
 - Setting up WWW server
- Client Inventory and registration
 - Client daemon - Pull
 - Server - Push
 - Authentication

Setup



CM Systems

System	Web site	Languages and formats			Daemons	
		Impl	Config	Template	Server	Client
Ansible	ansible.com	Python	YAML	Jinja	No	No
Salt	saltstack.com	Python	YAML	Jinja	Optional	Optional
Puppet	puppet.com	Ruby	custom	ERB ^a	Optional	Optional
Chef	chef.io	Ruby	Ruby	ERB	Optional	Yes

a. ERB (embedded Ruby) is a basic syntax for embedding Ruby code in templates.

Microsoft Configuration Management (CM) - Windows GUI, Point and click.

Why YAML?

- Simple Serialization Format
 - Encode an object as string
- Not executable
- Human readable
 - Whitespace delimited
- Broad support for languages
- Types
 - Integers, Floats, Timestamps, Collections, Strings, Dictionary (key:value)
- Defines two functions
 - Dump (Present) and Load (Construct)

<https://yaml.org/>

Example 2.27. Invoice

```
--- !<tag:clarkevans.com,2002:invoice>
invoice: 34843
date   : 2001-01-23
bill-to: &id001
  given : Chris
  family: Dumars
  address:
    lines: |
      458 Walkman Dr.
      Suite #292
    city   : Royal Oak
    state  : MI
    postal : 48046
ship-to: *id001
product:
  - sku      : BL394D
    quantity : 4
    description: Basketball
    price    : 450.00
  - sku      : BL4438H
    quantity : 1
    description: Super Hoop
    price    : 2392.00
tax   : 251.42
total: 4443.52
comments:
  Late afternoon is best.
  Backup contact is Nancy
  Billsmer @ 338-4338.
```

Example 2.28. Log File

```
---
Time: 2001-11-23 15:01:42 -5
User: ed
Warning:
  This is an error message
  for the log file
---
Time: 2001-11-23 15:02:31 -5
User: ed
Warning:
  A slightly different error
  message.
---
Date: 2001-11-23 15:03:17 -5
User: ed
Fatal:
  Unknown variable "bar"
Stack:
  - file: TopClass.py
    line: 23
    code: |
      x = MoreObject("345\n")
  - file: MoreClass.py
    line: 58
    code: |-
      foo = bar
```

Our term	Ansible	Salt	Puppet	Chef
operation op type	task module	state function	resource resource type, provider	resource provider
op list parameter binding	tasks parameter play(book)	states parameter top file	class, manifest property, attribute classification, declaration	recipe attribute run list
master host client host client group	control host group	master minion nodegroup	master agent, node node group	server node role
variable fact	variable fact	variable grain	parameter, variable fact	attribute automatic attribute
notification handler	notification handler	requisite state	notify subscribe	notifies subscribes
bundle bundle repo	role galaxy	formula GitHub	module forge	cookbook supermarket

Introduction to Ansible

- Configuring Ansible
 - Default configuration - `/etc/ansible/ansible.cfg`
 - User configuration - `~/.ansible.cfg`

Example

1. Install sudo
2. Copy a sudoers file to server
3. Correct permissions
4. Add group named sudo
5. Add users to sudo group

pip and virtualenv

- pip is a tool for installing and managing Python packages
 - ansible
 - psycopg2 - PostgreSQL database driver
 - Bundled with Python ≥ 3.4
- PyPi (Python Package Index) is a software repository
 - 221,549 projects
- virtualenv is a tool to isolate your Python environment
 - Separates libraries from the system

Virtual Environments

- What if different applications require conflicting versions of a package?
- Can't install everything to `/usr/lib/python3.6/site-packages`
- Virtual environments have their own installation directories
- Work on different projects or use different applications within their own virtual environments

virtualenv

- `apt install virtualenv python3-virtualenv`
- `virtualenv -p python3 path/env_name` — create environment
- `source path/env_name/bin/activate` — activate environment (Unix)
- pip is installed into the environment

Setup

```
ssh $NETID@banyan.engr.unr.edu
```

```
mkdir confman && cd confman/  
virtualenv -p python3 ansible_env  
source ansible_env/bin/activate  
pip install ansible
```

\$HOME/.ansible.cfg

```
[defaults]
```

```
private_key_file = /home/$NETID/.ssh/server
```

```
remote_user = root
```

\$HOME/hosts.yml

```
[vm]
```

```
cs447-newellz2.ncr
```

```
[vm:vars]
```

```
ansible_python_interpreter=/usr/bin/python3
```

```
host_key_checking = False
```

```
ansible_ssh_common_args='-o StrictHostKeyChecking=no'
```

#Test

ansible -i hosts.yml -m ping vm

#Let's do something more advanced now

Windows Ansible

- OpenSSH Server or WinRM
- Very new
 - alpha
- Quite exciting, it replaces WinRM
 - WinRM was HTTP based
- Install Document
 - https://docs.microsoft.com/en-us/windows-server/administration/openssh/openssh_install_first_use

Steps

Enable and Install SSH

Add-WindowsCapability OpenSSH.Server~~~0.0.1.0

Add-WindowsCapability OpenSSH.Client~~~0.0.1.0

#Generate Keys

#Modify C:\ProgramData\ssh\administrator_authorized_keys

Windows Tasks

- Installing Software
- Installing Updates
- Users and Groups
- Registry Modification
- “Shell” commands
 - cmd.exe or powershell
 - cmd.exe is the default
 - HKLM/Software/OpenSSH DefaultShell:string
- Scheduled Tasks

Notes

<https://galaxy.ansible.com/community/general>