ANSIBLE

Introduction

$whoami

- Lead Database Admin @FBSI
- A Decade experience with multiple technologies
- RHCE/RHCoE-Ansible/VCP-DCV/ITIL - Certified
- FOSS Lover

linkedin.com/in/raghuram-sethuraman-71318982

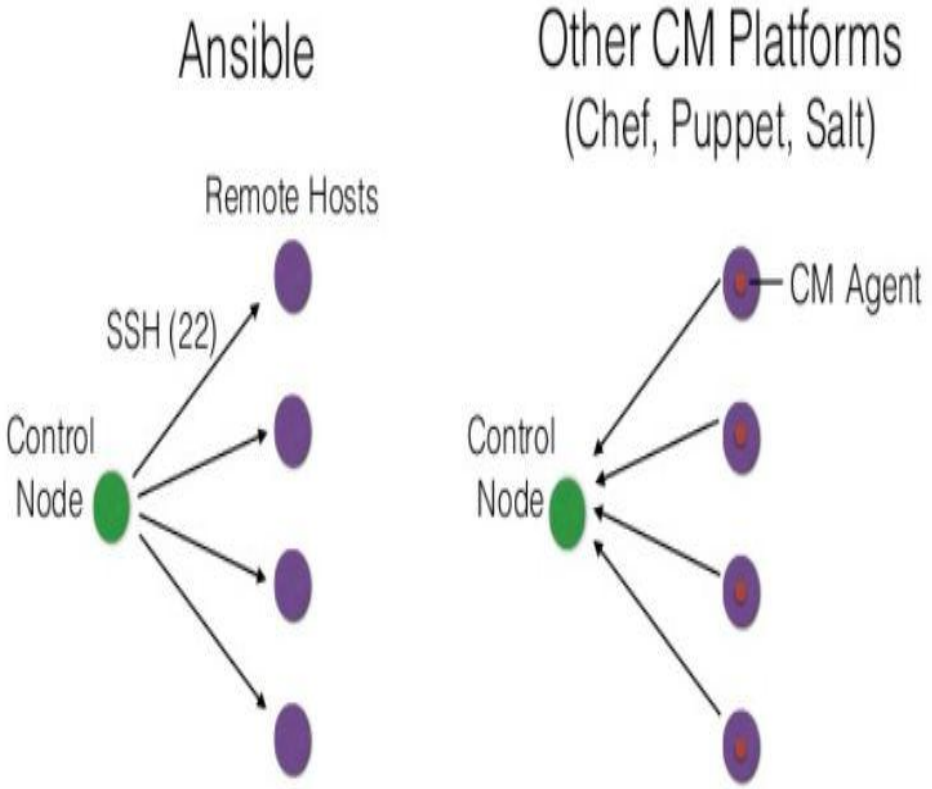
Agenda

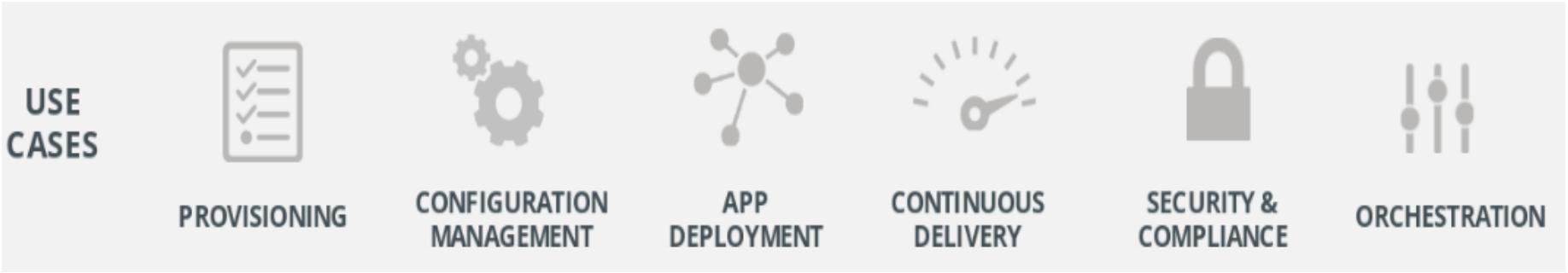
- Introduction & Use-Cases
- Architecture
- Installation & directory structure
- Modules
- Ansible Ad-Hoc Commands
- Ansible Facts

- Simple Playbook
- Conditions & Template
- Loops & Simple Variable
- Tags Example
- debug & Ignore_Errors
- Ansible Roles
- Q&A

Introduction & Use-Cases

- Efficient → Agentless, minimal setup, desired state (no unnecessary change), push-Based architecture, Easy Targeting based on Facts
- Fast →Easy to learn/to remember, simple declarative language
- Scalable →Can manage thousands of nodes
- Secure →SSH transport
- Large community → 4K of roles on Ansible

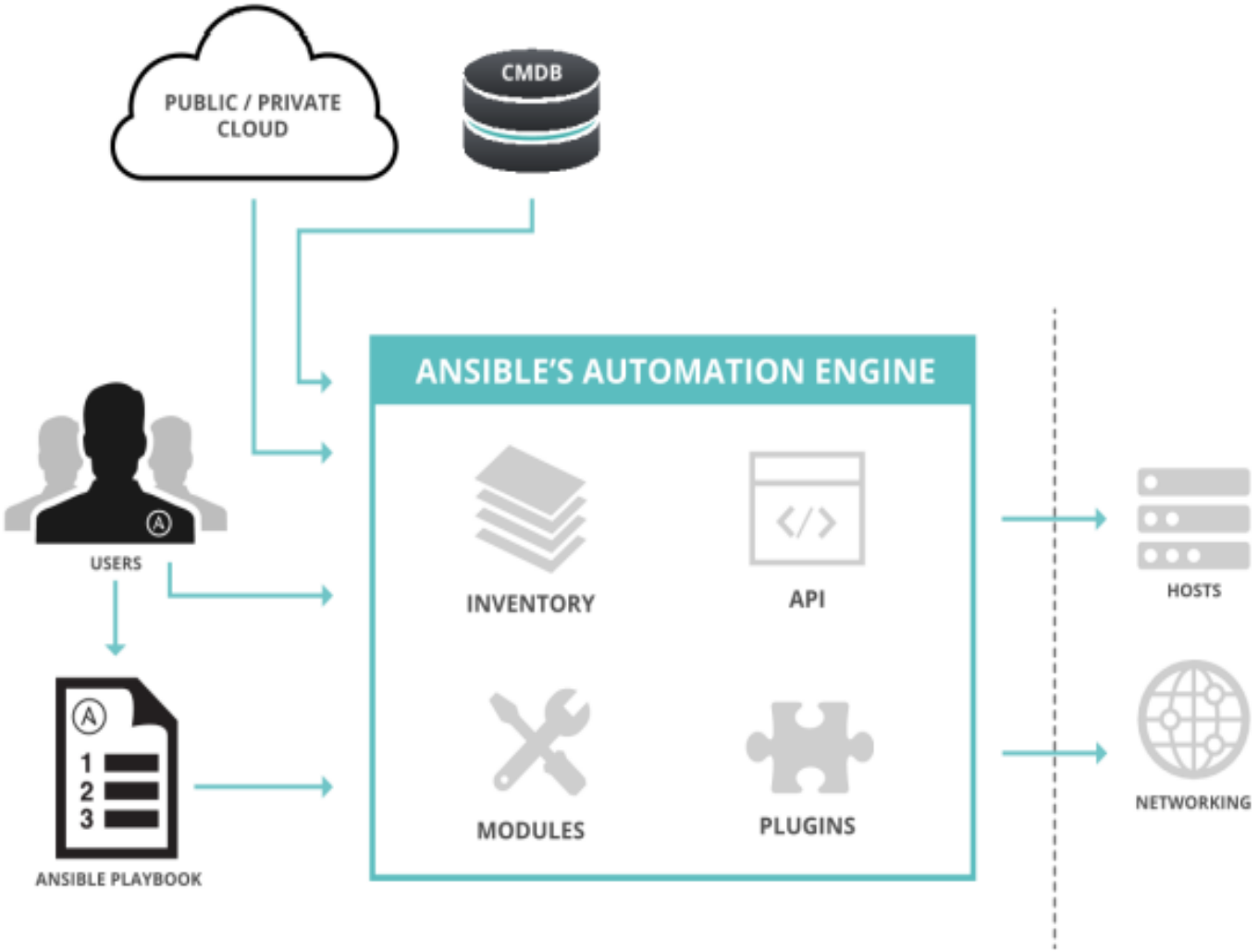




Introduction – contd…

- Powerful CM & IaC tool
- Agentless (No Managing the Management / Upgrades are "Zero Downtime”)
- Written in Python / PowerShell / Shell / Ruby
- **Languages Support**
  - Languages - .py/.sh/.ps1/.rb
  - Jinja2 Template - File extension can be ".j2" format
  - YAML - (YAML Ain't Markup Language) file extension can be ".yaml" or ".yml"
  - Modules - 1378 (Version 2.4.2.0)
- **Platform Support**
  - OS - Linux / Unix / Windows
  - Networking & Security - Juniper / Cisco / F5-Load balancers
  - Provisioning - Bare-Metal & Virtual Infrastructures
  - Containerization - Docker & Kubernetes
  - Cloud - Public (AWS / Azure / Google) / Private (OpenStack)
  - DevOps & Microservices - Atlassian / Splunk / Git / Jenkins /REST API
- **Pre-Requisite**
  - OpenSSH
  - Python 2.4 or later (For python 2.4 & 2.5, then python-simplejson is mandatory on managed nodes)

Architecture



Installation & directory structure

- **Installed through yum, apt-get & PIP**
  - RHEL & CentOS → EPEL(6.X) | By-Default(7.X)
  - Ubuntu / Debian → apt-get
  - Pip install → pip install --user ansible

- **Config files** - Parameters can be overridden & flexible to change directory
  - **File name > /etc/ansible/ansible.cfg**

- **Inventory** - Inventory is a collection of hosts(nodes) with associated data and groupings that Ansible can connect and manage
  - Hosts (Nodes)
  - Groups
  - Inventory-Specific data (Variables)
  - Static & Dynamic(AWS/Azure/Google - user defined) Inventory
  - **File name > /etc/ansible/hosts**

Config file

```
[defaults]
inventory = ./inventory
remote_user = someuser
ask_pass = false

[privilege_escalation]
become = true
become_method = sudo
become_user = root
become_ask_pass = false
```

Inventory file

```
[web]
Server1
Server2

[db]
192.168.0.1
192.168.[4:7].[0:255]

[datacenter:children]
web
db

[insecuressh:children]
datacenter

[insecuressh:vars]
ansible_ssh_common_args='-o
    StrictHostKeyChecking=no'
```

Modules

- Piece of code copied to the target system and execute the tasks declared.
- By-default modules are written in Python, but it can be written in any language.
- **Modules URL:** https://docs.ansible.com/ansible/latest/modules/modules_by_category.html
- **To list all module:**
  - *ansible-doc –l*
- **To View module options:**
  - *ansible-doc <Module Name>*

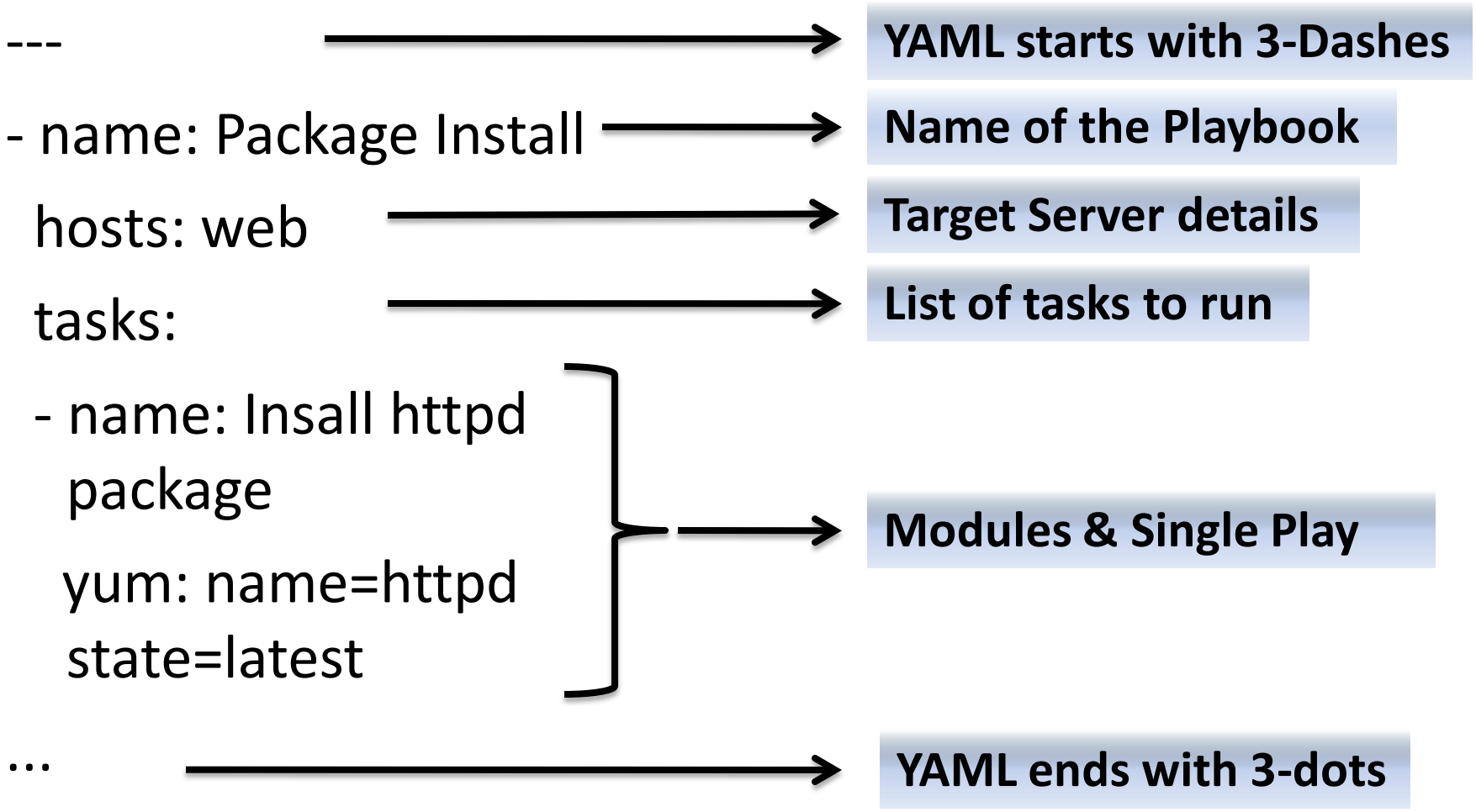
Ansible Ad-Hoc Commands

- To check remote host connectivity
  - *ansible test –m ping –u test*
- Checking uptime for remote machines
  - *ansible test -m shell -a "uptime" -u test*
- List all hosts
  - *ansible all --list-hosts*
- List hosts under specific group-host name
  - *ansible test -list*
- List all Group name & Group with hostnames
  - *ansible localhost -m debug -a 'var=groups.keys()'*
  - *ansible localhost -m debug -a 'var=groups‘*

Ansible Facts

- Gathers facts about remote host by default without any additional package
- Uses “setup” module
  - http://docs.ansible.com/ansible/setup_module.html
- Can collect reports from
  - *NIX machines
  - Networking Devices
  - Cloud Vendors
  - Containerization

Simple Playbook



Simple Playbook – contd…

```
- name: Install and start Apache HTTPD
  hosts: web
  tasks:
    - name: Install httpd package
      yum: name=httpd state=present
    - name: Sample index.html is present
      copy: src=files/index.html dest=/var/www/html/index.html
    - name: Start & Enable httpd service
      service: name=httpd state=started enabled=true
```

Conditions & Template

```
- name: Install and start Apache HTTPD
  hosts: web
  tasks:
    - name: Install httpd package
      yum: name=httpd state=present
    - name: Sample index.html is present
      template: src=templates/index.j2 dest=/var/www/html/index.html
      when:
          - ansible_distribution_version==“7.5”
```

Loops & Simple Variable

```
- hosts: web,db
  become: true
  vars:
    #created by - python -c 'import crypt; print crypt.crypt("test123,
     "$1$SomeSalt")'
    password: <Enter your Encrypted Password here>
  tasks:
    -  user: name="{{ item }}" password={{password}} state=present
       with_items:
            - user1
            - user2
            - user3
            - user4
```

Tags Example

```
tasks:
  - name: Package {{ item }} is installed
    yum:
      name: "{{ item }}"
      state: installed
    with_items:
      - httpd
      - memcached
    tags:
      - packages
```

debug & Ignore_errors

```
tasks:
   - name: Check local time
     command: date
     register: result
   - debug: var=result

   - name: Check foobar.com reachability
     command: ping –c1 www.foobar.com
     ignore_errors: yes
```

Ansible Roles

- Roles provide a framework for fully independent, or interdependent
- Collections of variables, tasks, files, templates, and modules.
- It's primary mechanism for breaking a playbook into multiple files, this simplifies writing **complex playbooks.**
- Roles require a particular directory structure.
- Unused directories need not exist, remove if any unused dir's.
- "main.yml" serves as the entry point for the role.
- Files in the tasks, templates, and files directories may be referenced without path within the role.
- When a role is called in a playbook, Ansible looks for the role definition in ${PWD}/roles/
- If ${PWD}/roles does not contain the sought role, then /etc/ansible/roles is checked.
- The full path to a role may also be specified with the role keyword to use a non-default path.

Roles – contd…

- **tasks** - Contains the main list of tasks to be executed by the role.
- **handlers** - Contains handlers, which may be used by this role or even anywhere outside this role.
- **defaults** - Default variables for the role (see Variables for more information).
- **vars** - Other variables for the role (see Variables for more information).
- **files** - Contains files which can be deployed via this role.
- **templates** - Contains templates which can be deployed via this role.
- **meta** - Defines some meta data for this role.


Questions?


Thank
you!!