
HANDS-ON
ANSIBLE

WHO WE ARE?

Let's introduce ourselves

- Say your name
- Say your role
- Say your background
- Experience with Ansible, puppet and other configuration management tools

ABOUT ME

OMPRAGASH VISWANATHAN

ANSIBLE ENGINEER, RED HAT

IRC & SLACK: Ompragash

ansibler@hotmail.com

AGENDA

- What is Ansible
- Architecture Overview
- Core Components
- References

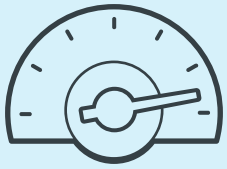
WHAT IS ANSIBLE?

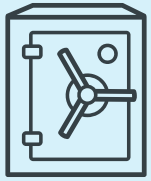
Ansible is an open source IT automation, configuration management, deployment, and orchestration tool.

- In which language Ansible it written?

Ansible is **written in Python 2**.

Python 3 support from version 2.5





SIMPLE

Human readable automation

No special coding skills needed

Tasks executed in order

Get productive quickly

POWERFUL

App deployment

Configuration management

Workflow orchestration

Orchestrate the App lifecycle

AGENTLESS

Agentless architecture

Uses OpenSSH and WinRM

No exploits or updates

More efficient and more sucure

REQUIREMENTS

- **CONTROL NODE**

① Ansible can be run from any machine with Python 2.6 or 2.7

- **MANAGED NODES**

① Linux/Unix - SSH, Python 2.4 or later

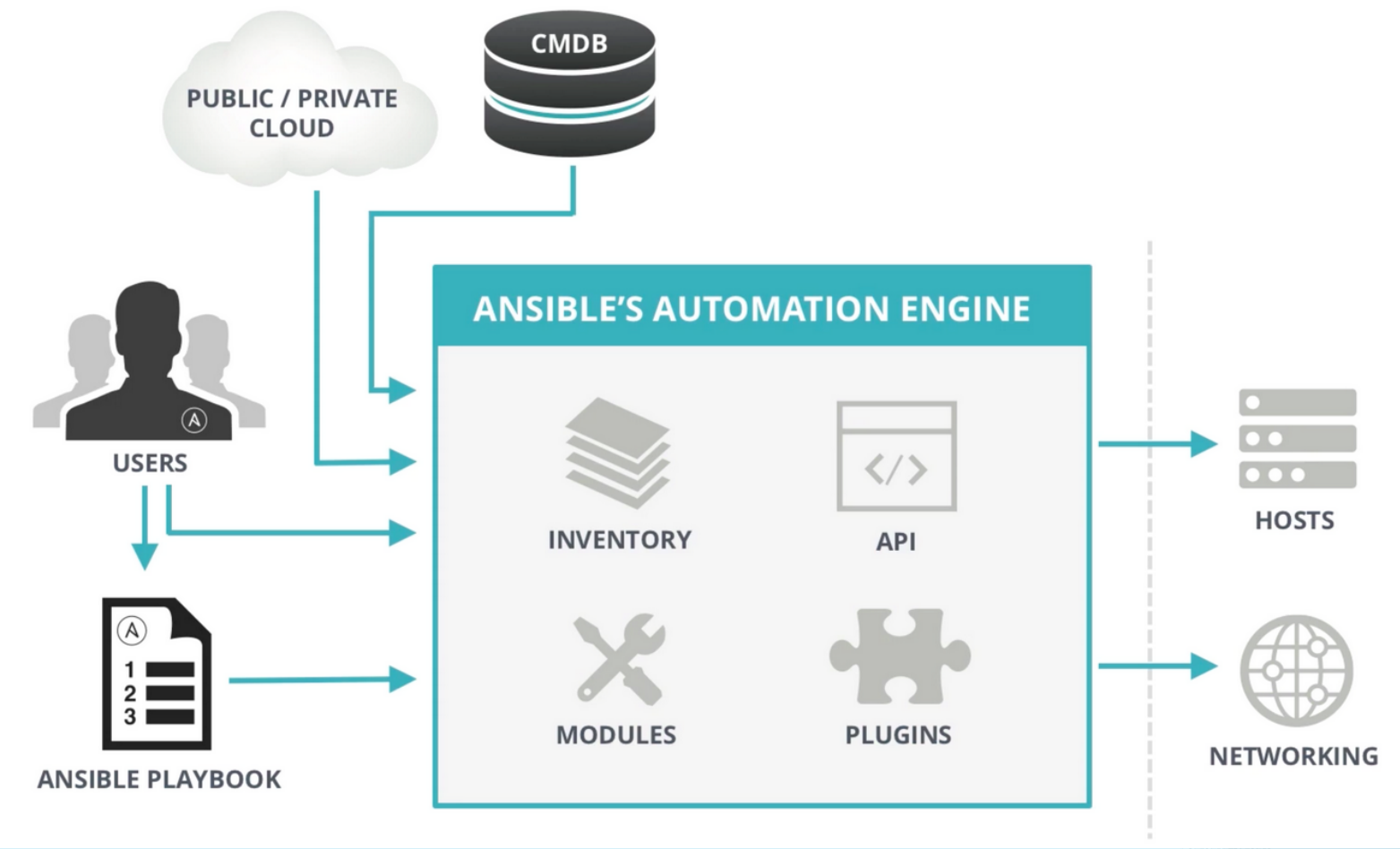
1.1 SSH

1.2 Python 2.4 or later

1.2.1 If running less than Python 2.5 on the remotes nodes, package "python-simplejson" is required. **Note**: RHEL 5.x has Python 2.4 only.

② Windows (Ansible 1.7+) - enable and configure PowerShell remoting 3.0+ (WinRM)

ARCHITECTURE OVERVIEW



ANSIBLE

INVENTORY

- Define which hosts ansible manages
- Inventory file has list of all managed host names one line per host
- Files are organized as hosts and groups
- A set of hosts can be under a group name
- A host can be in more than one group

TWO TYPES OF INVENTORIES:

Static - Defined in simple text files.

Dynamic - Generated for outside providers, some examples include pulling inventory from a cloud provider (OpenStack, AWS, etc), LDAP, Cobbler, or a piece of expensive enterprisey CMDB software.

STATIC INVENTORY EXAMPLE

```
[mariadb]
mariadb01
mariadb02

[pgsql]
pgsl01
pgsl02

[frontend]
nginx01
nginx02
httpd01

[databases:children]
mariadb
pgsql
```

DYNAMIC INVENTORY EXAMPLE

- These are Ansible's inventories whose information has been dynamically generated by an external source.
- A number of existing scripts are available from Ansible's GitHub site at https://github.com/ansible/ansible/tree/devel/contrib/inventory

```
#!/usr/bin/env python

'''
EC2 external inventory script
==================================

Generates inventory that Ansible can understand by making API request to
AWS EC2 using the Boto library.

NOTE: This script assumes Ansible is being executed where the environment
```

- Use "ansible-inventory -i ec2.py --list" to list host details

MODULES

- Modules control system resources - services, packages, files, system commands, etc.
- Module can be executed directly in cli or through playbooks.
- Language independent - Return JSON format data.
- Idempotent - avoids change to system unless needed

```
ansible python module location = /usr/lib/python2.7/site-packages/ansible
```

PLUGINS

- Gears in the engine
- Python that plugs into the core engine
- Adaptability for various uses and platforms

```
Available Plugins

cache, callback, connection, inventory, lookup,
shell, module, strategy, vars

Usage: ansible-doc [-l|-F|-s] [options] [-t <plugin type> ] [plugin]
```

WHY YAML?

Ansible playbooks are written in YAML language

YAML language was designed to represent data structures in easy-to-write, human-readable format

- YAML File Syntax
  - Start of document: ---
  - Optional marker at the end of document: ...
- Space characters **(not Tabs)** used for indentation

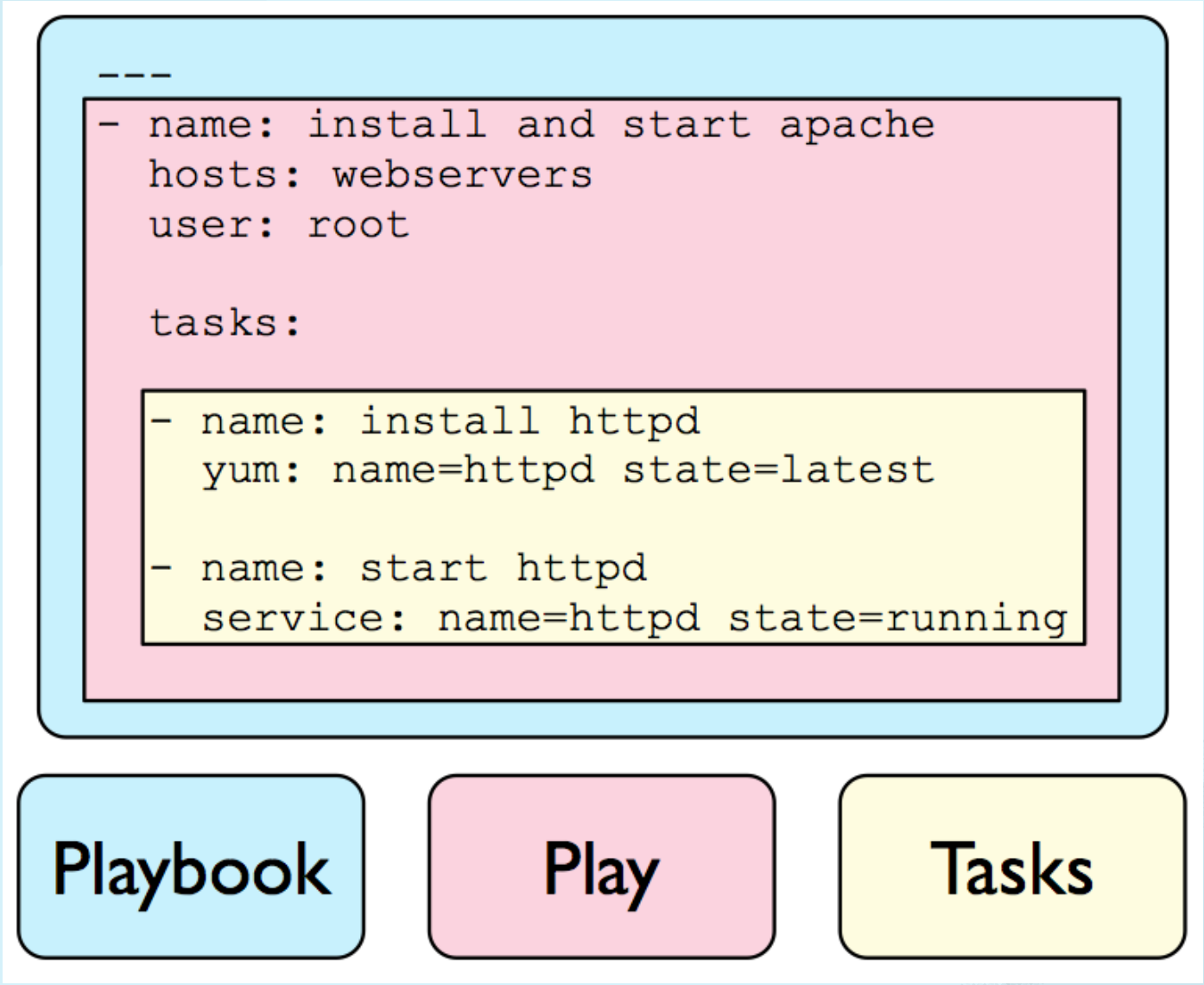
- Indentation rules:
  - Elements at same level in hierarchy must have same indentation
  - Child elements must be indented further than parents
  - No rules about exact number of spaces to use

PLAYBOOKS

Playbooks are Ansible's configuration, deployment, and orchestration language and they are expressed in YAML format and have a minimum of syntax.

- Playbook contain Plays
- Plays contain Tasks
- Tasks call Modules
- Tasks run sequentially

PLAYBOOKS EXAMPLE



PLAYBOOKS WITH MULTI PLAYS EXAMPLE

```
---
# This is a simple playbook with two plays

-  name: first play
   hosts: web.example.com
   tasks:
   - name: first task
     service:
       name: httpd
       enabled: true

-  name: second play
   hosts: database.example.com
   tasks:
   - name: first task
     service:
       name: mariadb
       enabled: true
...
```

WHO IS USING ANSIBLE?

ADDITIONAL REFERENCES

GETTING STARTED

- Would you like to learn Ansible? It's easy to get started:

ansible.com/get-started

- Want to learn more?

ansible.com/whitepapers

Best Practices Essentials:

https://www.ansible.com/blog/ansible-best-practices-essentials

Ansible Examples - This repository contains examples and best practices for building Ansible Playbooks

https://github.com/ansible/ansible-examples