Ansible Introduction

QUALITY THOUGHT

INFOSYSTEMS (INDIA) PVT. LTD.

What Is Ansible?

Image Credit: Courtesy of Summit Entertainment

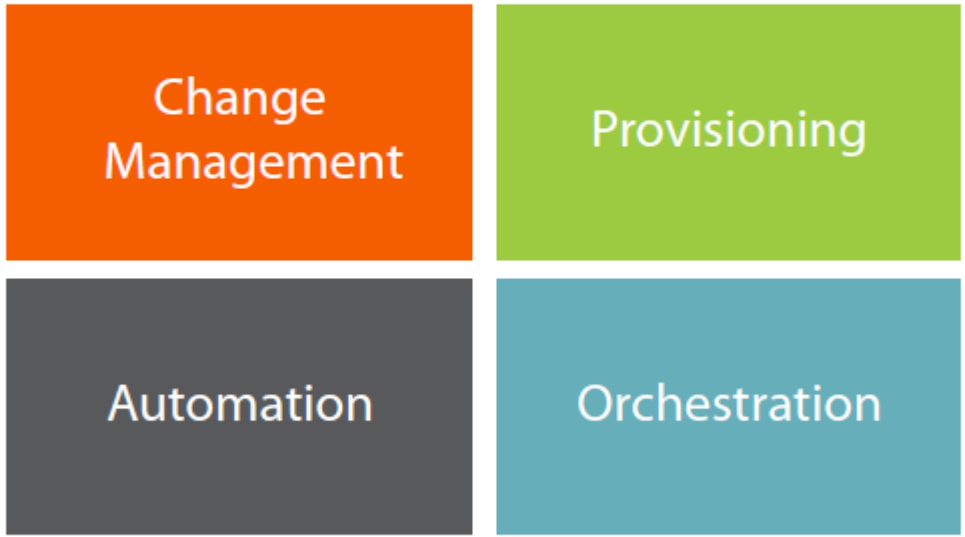
What is an Ansible?

Ansible is, in short, an IT automation, configuration management and provisioning tool.

It uses 'playbooks' to deploy, manage, build, test and configure anything from full server environments to web sites to custom compiled source code for applications.

It brings together aspects of environment management that have been traditionally separate and managed independently.

What Is Ansible?



Change Management

Define a "System State"

- Enforce the System State

System State

- Apache Web Installed
- Apache Web at version x.xx.x
- Apache Web Started

A function is idempotent if repeated applications has the same affect as a single application

IDEMPOTENCE

Provisioning

Prepare a system to make it ready

- Transition from one state to a different state

Examples

- Make an FTP Server
- Make an Email Server
- Make a DB Server

1. Install web software
2. Copy configurations
3. Copy web files
4. Install security updates
5. Start web service

Automation

Define tasks to be executed automatically

- Ordered Tasks
- Make decisions
- Ad-hoc tasks

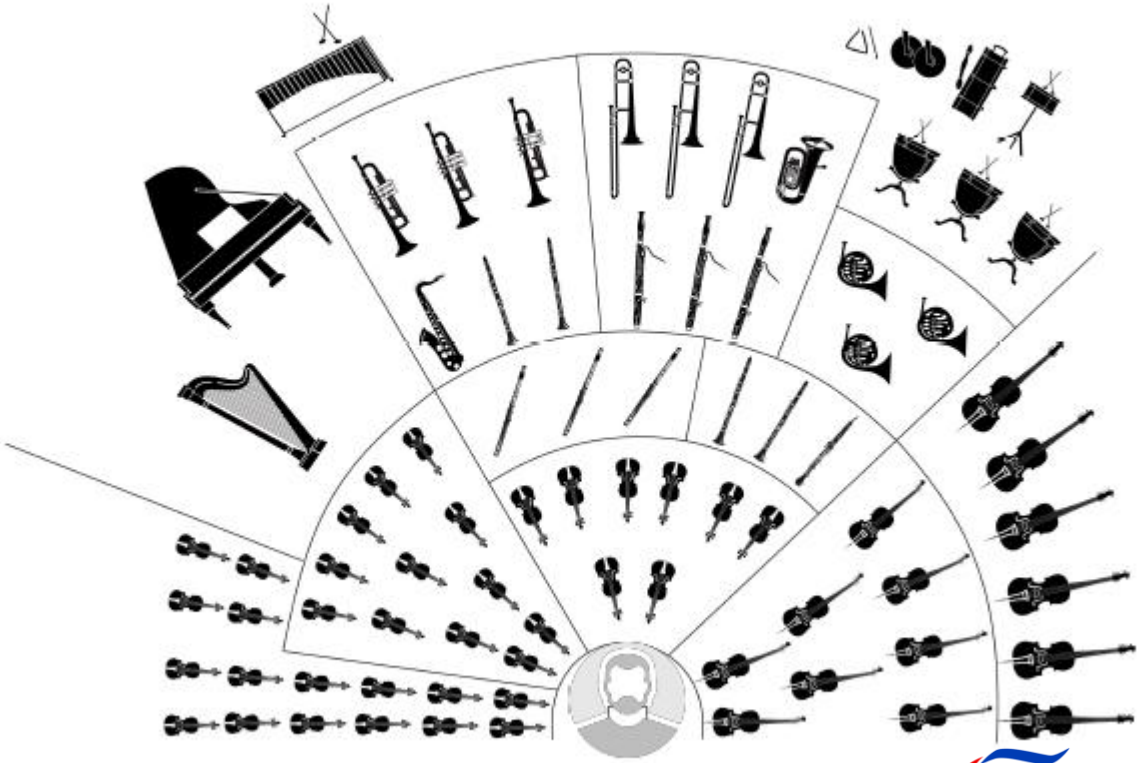
Set it and Forget it

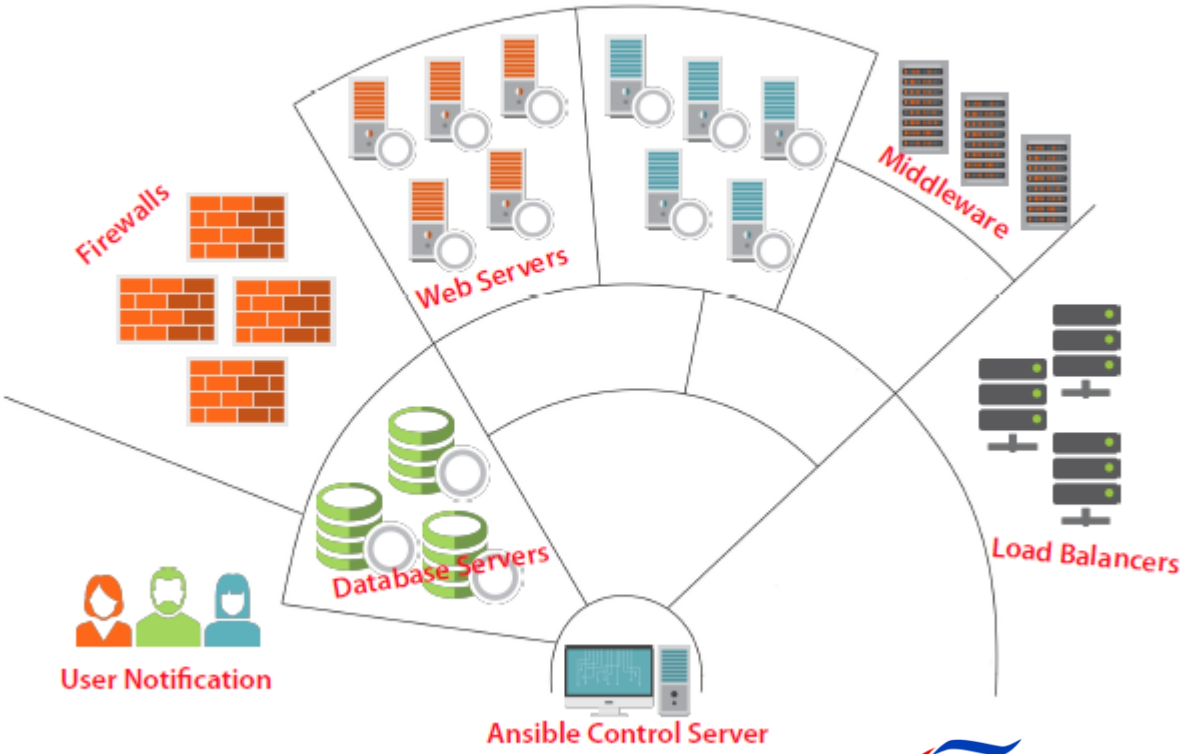
- Run the task
- Get a cup of coffee
- Walk back to desk seeing tasks finished
- Sip your coffee and feel productive

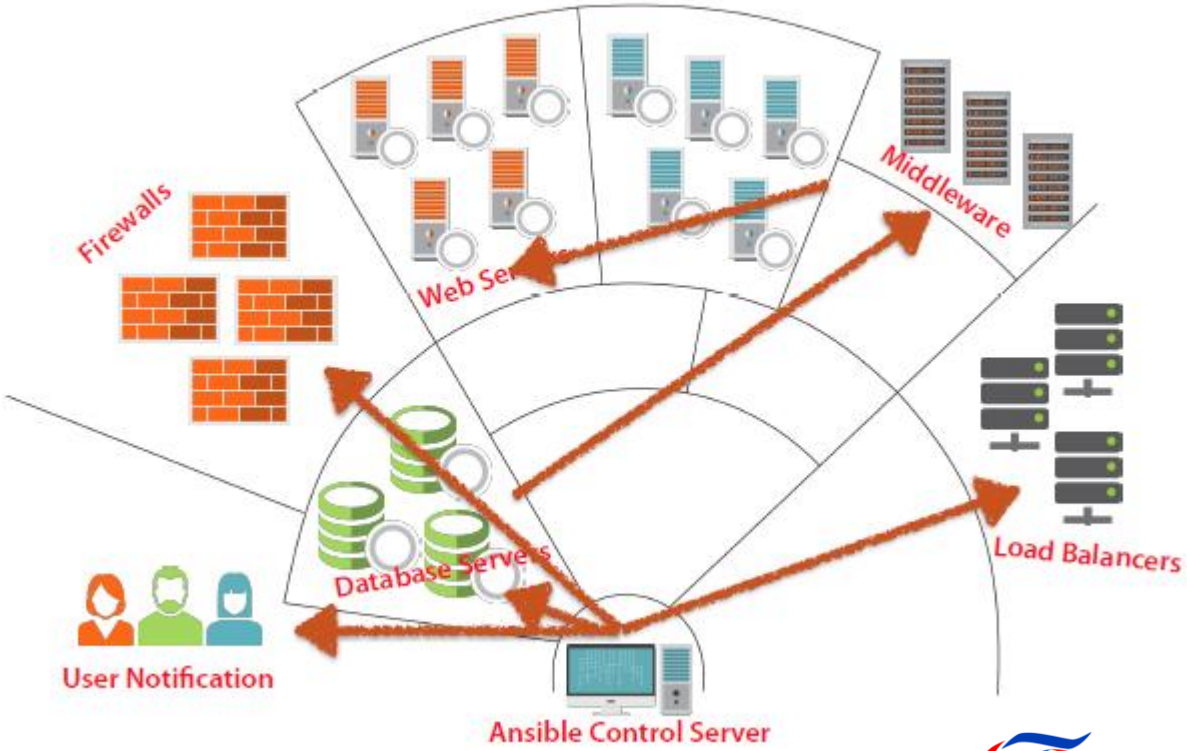
Orchestration

Coordinates automation BETWEEN systems

- Task 1 - System 1
- Task 2 - System 2
- Task 3 - System 3
- Task 4 - System 1







Why Ansible?

What makes it so different?

It's clean!

- No agents
- No database
- No residual software
- No complex upgrades

YAML

Ansible Execution

No programming required

NOT a markup language

Structured

Easy to read and write

Built-in security

- Uses SSH
- Root / Sudo usage
- Encrypted vault
- No PKI needed

What Other Tools?

Basically, Ansible 'plays in the sandbox' of a large number of deployment and configuration management tools such as:

- Jenkins
- Salt
- Puppet
- Chef
- Fabric

However, Ansible works at a high enough level that it can also be used in conjunction with one or more of these tools. Ansible is very often called an 'orchestration' tool since it can function independently as well as 'control' one or more of the tools listed above.

Ansible Vs. The World

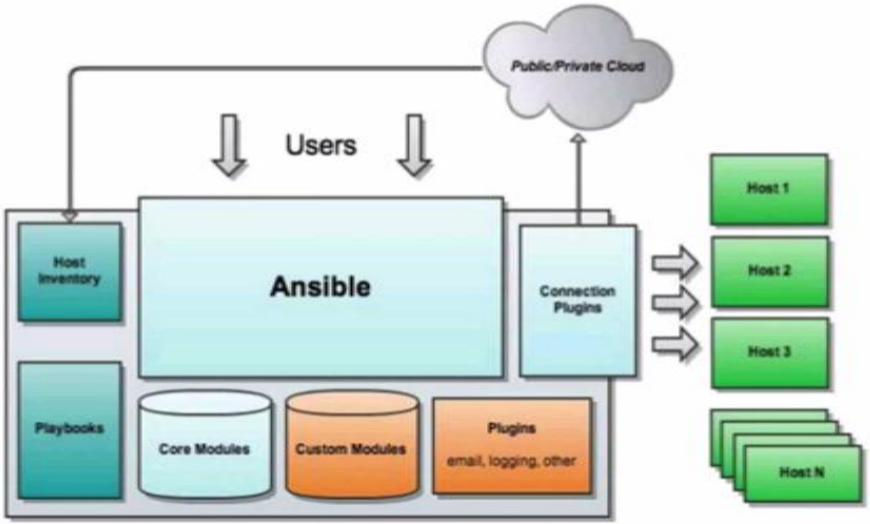
There are some key differences in these technologies, let's talk about a few:

- Server/Management Nodes
  - Puppet/Chef will usually contain a 'master' or 'controller' server in the setup. Ansible, operating only with SSH, does not. Any system with Ansible can function in that role at any time based on the task or deployment type.
- Workflow – Push vs. Pull
  - Since most configuration management tools have a 'master' server, they use the 'pull' method (i.e. the client 'checks in' with the server to pull it's configuration). Ansible uses the 'push' method, requiring no client installation or configuration (other than general Python).

Ansible Vs. The World (Continued)

- Resource Definitions and Execution Ordering
  - Puppet (for example) instructions are not applied in order (in other words, not 'top to bottom' in how they appear in the manifest). Ansible uses pure in order execution, which can be easy to read as well as convert from other languages or scripts.
- Language
  - Ansible is built upon Python and the huge standard of inclusive functionality that comes with it. Puppet (Ruby) and Chef or Salt are not quite so inclusive.
- Syntax
  - Puppet is based on custom DSL while Ansible is based on YAML standard.

Ansible architecture



Taking control of your environment with a single tool has a number of advantages.

With Ansible, you can control server deployment configuration, making everything consistent. With modules and plugins, you can build or 'hook' into other applications and control them as well!