Contents

[Configuration Management 1](#_Toc68154605)

[Introduction 2](#_Toc68154606)

[Ansible Terminology 2](#_Toc68154607)

[Advantages 3](#_Toc68154608)

[YAML 3](#_Toc68154609)

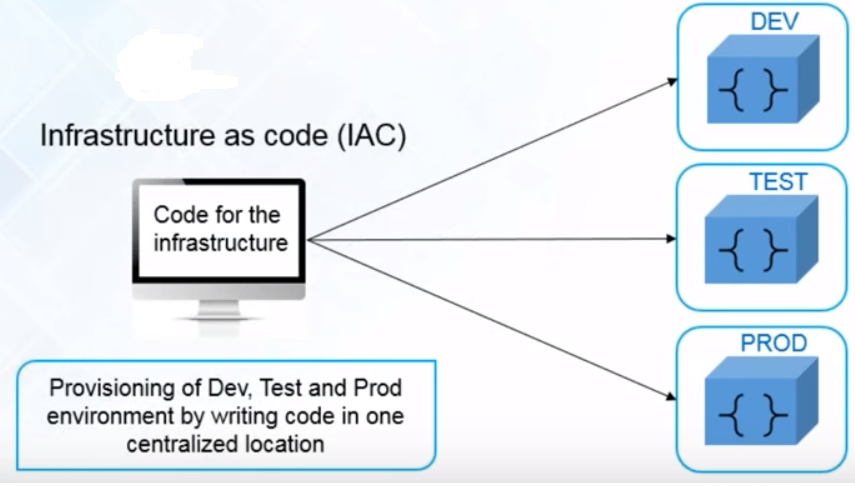
[Data Formats 4](#_Toc68154610)

[Variables: 4](#_Toc68154611)

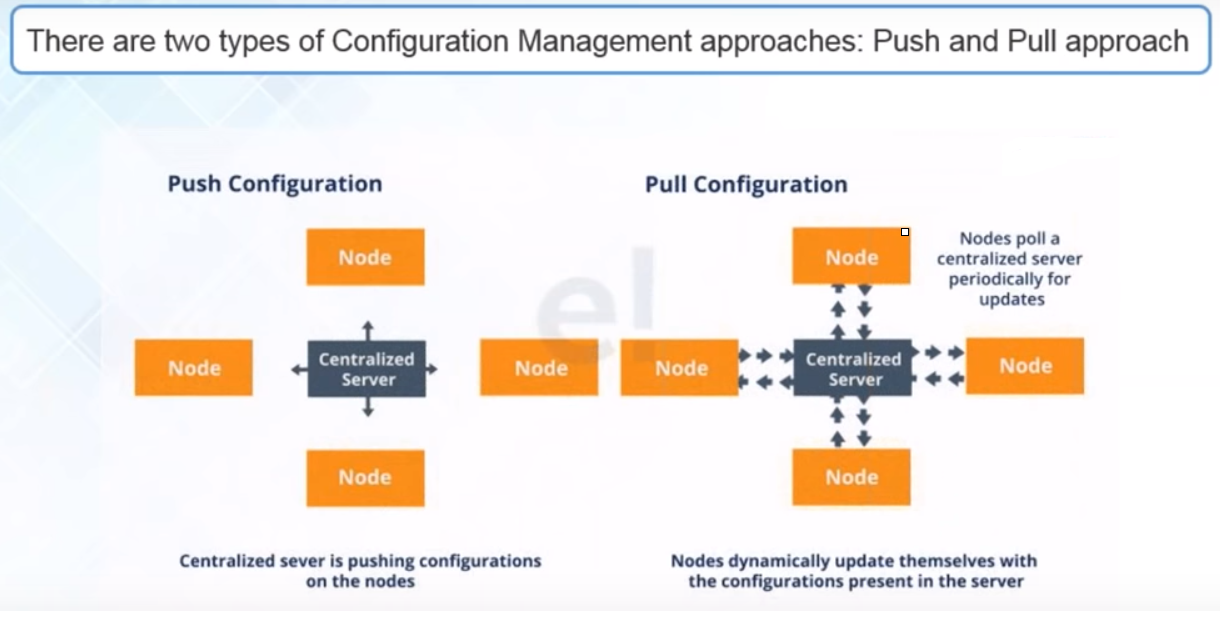
[Facts 5](#_Toc68154612)

# Configuration Management

* Configuration Management is the practice of handling changes systematically so that a system maintains its integrity over time. It allows access to an accurate historical record of system state.



Configuration Management Types



# Introduction

* Ansible is an open source IT Configuration Management, Deployment & Orchestration tool.
* It aims to provide large productivity gains to a wide variety of automation challenges.
* This tool is very simple to use yet powerful enough to automate complex multi-tier IT application environments.
* What Problems we have before Ansible?
* Ansible is a helpful tool that allows you to create groups of machines, describe how these machines should be configured or what actions should be taken on them.
* Ansible issues all commands from a central location to perform these tasks.
* No other client software is installed on the node machines. It uses SSH to connect to the nodes.

# Ansible Terminology

* **Controller Machine**: The machine where Ansible is installed, responsible for running the provisioning on the servers you are managing.
* **Inventory**: An initialization file that contains information about the servers you are managing.
* **Playbook**: The entry point for Ansible provisioning, where the automation is defined through tasks using YAML format.
* **Task**: A block that defines a single procedure to be executed, e.g. Install a package.
* **Module**: A module typically abstracts a system task, like dealing with packages or creating and changing files. Ansible has a multitude of built-in modules, but you can also create custom ones.
* **Role**: A pre-defined way for organizing playbooks and other files in order to facilitate sharing and reusing portions of a provisioning.
* **Play**: A provisioning executed from start to finish is called a play*.*In simple words, execution of a playbook is called a play.
* **Facts**: Global variables containing information about the system, like network interfaces or operating system.
* **Handlers**: Used to trigger service status changes, like restarting or stopping a service.

# Advantages

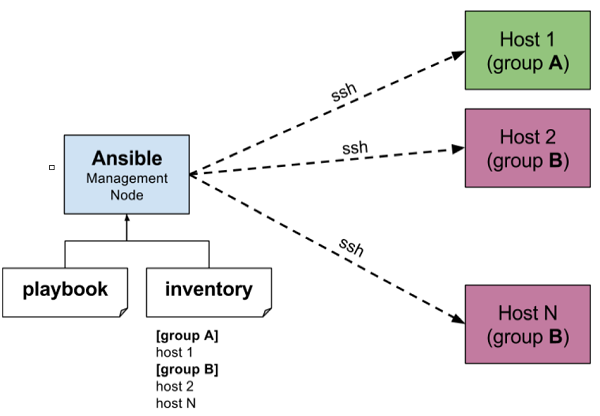
**Simple:**Ansible uses a simple syntax written in YAML called***playbooks***. YAML is a human-readable data serialization language. It is extraordinarily simple. So, no special coding skills are required

**Agentless:**There are no agents/software or additional firewall ports that you need to install on the  client systems or hosts which you want to automate. Ansible further reduces the effort required for your team to start automating right away.

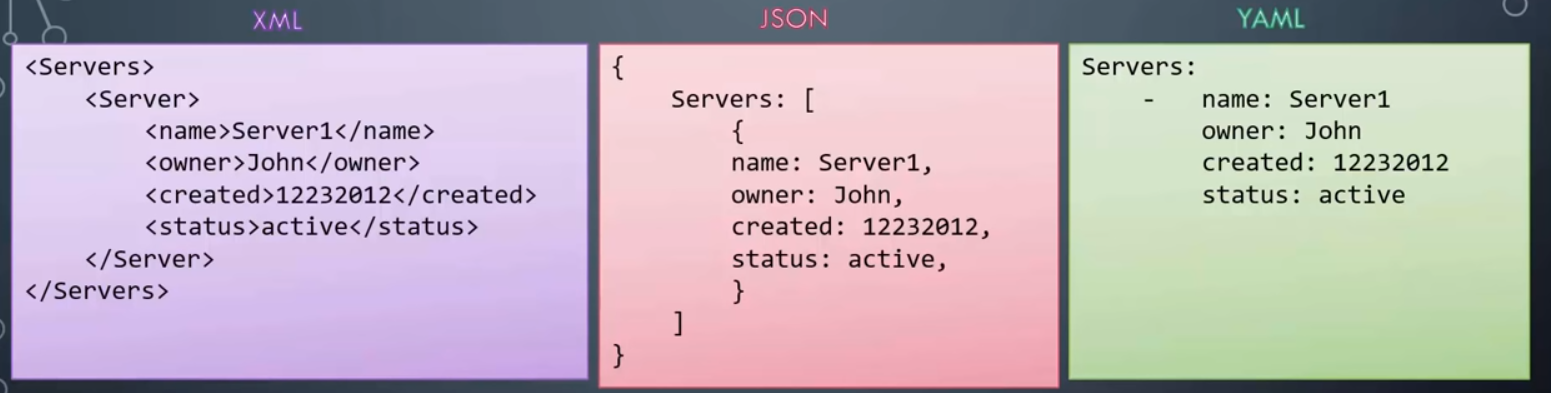
**Powerful & Flexible:** Ansible has powerful features that can enable you to model even the most complex IT workflows. can manage the infrastructure, networks, operating systems and services that you are already using, as Ansible provides you with hundreds of modules to manage them. Together Ansible’s capabilities allow you to orchestrate the entire application environment regardless of where it is deployed.

# YAML

* YAML Ain't Markup Language
* is a data serialization language that matches user’s expectations about data.
* It designed to be human friendly and works perfectly with other programming languages.
* YAML is case sensitive
* The files should have .yaml extension
* YAML does not allow the use of tabs while creating YAML files; spaces are allowed instead
* Comments in YAML begins with the (#) character.
* Indentation of whitespace is used to denote structure.
* YAML supports single line comments, does not support multi line comments
* Keys and Values are separated by colon(:) and space



## Data Formats



# Variables:

* Allows you to customize behavior for systems, since not all systems are the same.
* Variables are how we deal with the differences between systems
* Variables can be defined in the Inventory/Playbook.
* Variables can be referenced using the jinga2 templating system
  + Dest = {{ remote\_path }}

# Facts

* Ansible facts gather information from the node where ansible is executed.
* To collect facts ansible uses a module called as setup
* You can use these facts in playbook variables
* Gathering facts can be disabled in a playbook
  + It is not always required
  + Can speed up execution
    - hosts:mainhosts
    - gather\_facts:no