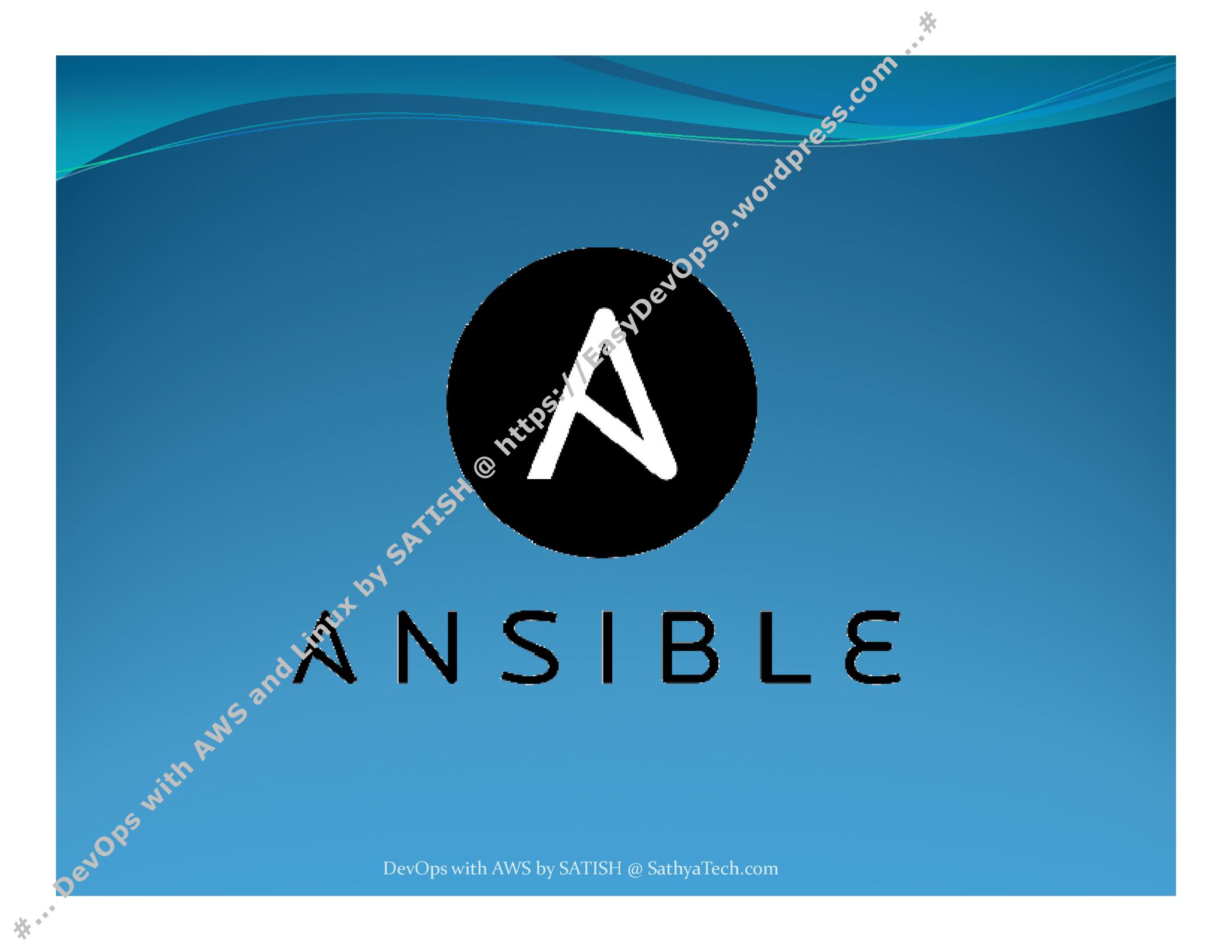




# ANSIBLE

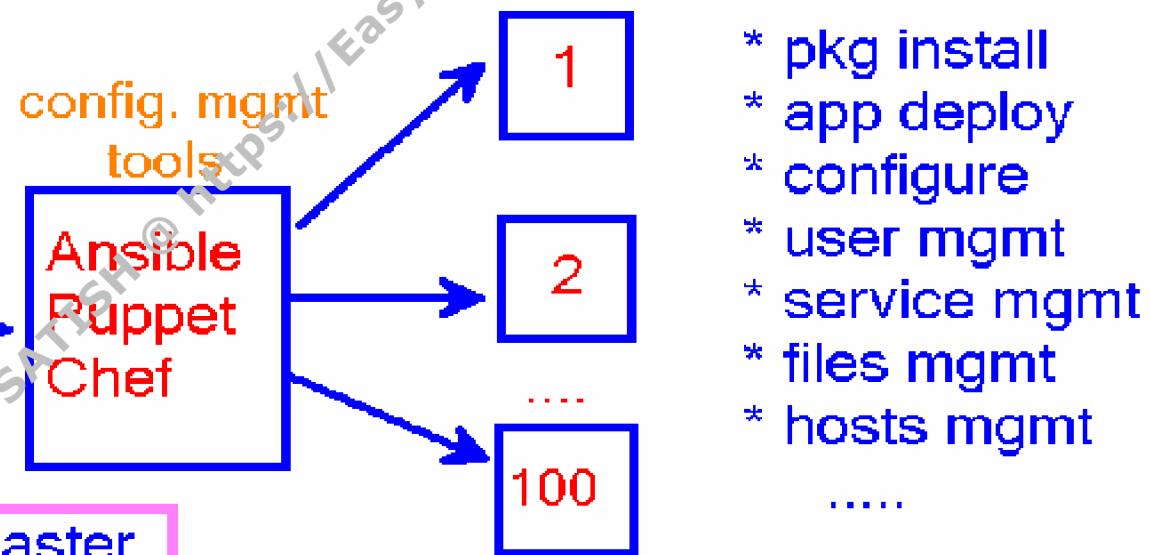
DevOps with AWS by SATISH @ SathyaTech.com



# Configuration Management tools

- \* Ansible : Playbooks (YAML)
- \* Chef : Cookbooks (RUBY)
- \* Puppet : Manifests (DSL)

linux servers



\* Ansible : Master

\* Puppet : Master , Agent

\* Chef : Server, client, Workstation

# Configuration Management tools

	<b>Ansible</b>	<b>Puppet</b>	<b>Chef</b>
<b>Components:</b>	<b>ansible master</b>	<b>master, agent</b>	<b>server,client workstation(dk)</b>
<b>Language:</b>	<b>Yaml / Python</b>	<b>DSL</b>	<b>Ruby</b>
<b>Scripts:</b>	<b>Playbooks</b>	<b>Manifest</b>	<b>Cookbooks</b>
<b>default file:</b>	<b>main.yml</b>	<b>site.pp</b>	<b>default.rb</b>
<b>node info:</b>	<b>Setup</b>	<b>Facter</b>	<b>Ohai</b>
<b>model:</b>	<b>push</b>	<b>pull</b>	<b>pull</b>
<b>connection:</b>	<b>ssh(secure shell)</b>	<b>ssl (secure socket Layer)</b>	<b>ssh</b>
<b>Fund.Unit :</b>	<b>modules</b>	<b>resources</b>	<b>recipes</b>

# Configuration Management tools

core-components

Ansible --> Master  
Puppet --> Master, Agent  
Chef --> Server, Client, ws

scripts

ansible --> playbooks  
chef --> cookbooks  
puppet --> manifests

model

Ansible --> Push  
Puppet --> Pull  
Chef --> Pull

WS: Chef Work Station

scripting language

Ansible --> YAML  
Puppet --> DSL  
Chef --> Ruby

connection process

Ansible --> SSH  
puppet --> SSL  
chef --> SSH

fund. units

Ansible --> module  
Puppet --> resources  
Chef --> recipes

YAML: yet another markup language

SSH : secure shell

DSL : domain specific language

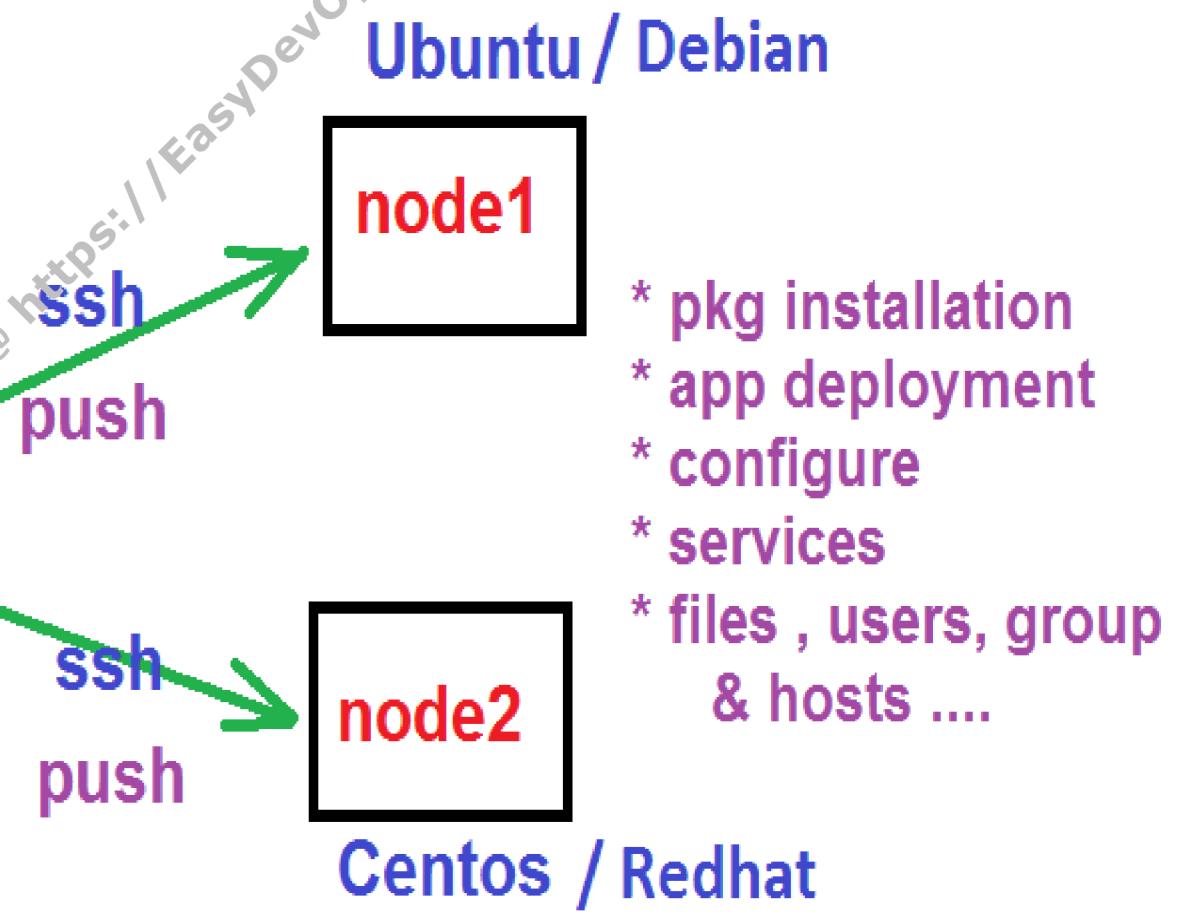
SSL : secure socket layer

# Introduction

- Configuration management systems are designed to make controlling large numbers of servers easy for administrators and operations teams.
- They allow you to control many different systems in an automated way from one central location.
- Ansible is a radically simple IT automation platform that makes your applications and systems easier to deploy.

# Ansible Master-Node

- \* playbooks
- \* roles
- \* variables
- \* handlers
- \* tasks
- \* plays
- \* templates
- \* loops
- \* ad-hoc commands



# What is configuration management

- Package Installation
- Configuration of servers
- Application deployment
- Continuous testing of already installed application
- Provisioning  
(provisioning means "providing" or making something available)
- Orchestration
- Automation of tasks

# Why Ansible ?

- It is a free open source application
- Agent-less – No need for agent installation and management
- Python/yaml based
- Highly flexible and configuration management of systems.
- Large number of ready to use modules for system management
- Custom modules can be added if needed
- Configuration roll-back in case of error
- Simple and human readable

# Examples for Ansible modules:

- \* **ping** : to check a node connection / reachability
- \* **apt** : to manage packages for Ubuntu nodes
- \* **yum** : to manage packages for Redhat nodes
- \* **package**: to manage packages for All nodes (ubnt/cent..)
- \* **setup** : to get a node info / attributes
- \* **debug**: to dispaly messages
- \* **user** : to manage users (add, drop...)
- \* **group**: to manage groups (add, drop..)
- \* **file** : to manage files (create, delete...)
- \* **copy** : to copy a file to a specific node
- \* **template**: to copy a file to a node with required changes
- \* **command** : to run linux commands (default module)
- \* **shell** : to run linux shell scripts
- \* **raw** : to run linux admin command / scripts
- \* **git** : to clone the code from specific git repo
- \* **get\_url** : to download a required packages / files
- \* **unarchive**: to extract packages
- \* **cron** : to schedule cron jobs

# Ansible Architecture

