ANSIBLE

S1: Introduction to Ansible

HRISHIKESH MOHAN

CONFIGURATION MANAGEMENT

Configuration management is the systematic approach to handling changes in system settings, software, and hardware configurations to maintain consistency and reliability across IT infrastructure. It encompasses:

- System State Management: Ensuring systems remain in their intended, desired state
- Change Control: Tracking and managing modifications to system configurations
- Consistency Enforcement: Maintaining uniform settings across multiple systems
- Documentation: Recording all configuration changes for audit trails

BENEFITS OF CONFIGURATION MANAGEMENT

- Improved System Recoverability: Create baselines for functional configurations that can be restored when issues arise
- Reduced System Disruption: Minimize downtime through consistent, automated configuration deployment
- Enhanced Security: Enforce security policies uniformly across all systems
- Accelerated Development: Automate testing and deployment processes
- Simplified Troubleshooting: Compare current configurations against known baselines for faster issue resolution

ANSIBLE ARCHITECTURE AND CORE CONCEPTS

- Agentless Architecture: No software installation required on managed hosts—uses SSH/WinRM
- Declarative YAML Syntax: Human-readable playbooks that describe desired system states
- Idempotent Operations: Running the same command multiple times produces consistent results
- Extensive Module Library: Over 1,300 built-in modules for various automation tasks
- Platform Agnostic: Manages heterogeneous environments without platform-specific agents

ANSIBLE ARCHITECTURE COMPONENTS

- Control Node: The central system where Ansible is installed and executed.
- Managed Nodes: Target systems that Ansible configures and manages.
- Inventory: Configuration file listing all managed hosts and their groupings.
- Modules: Small programs that perform specific tasks on managed nodes.

INSTALLATION PREREQUISITES AND PROCESS

Before installing Ansible, ensure your environment meets these prerequisites:

- Control Node: Linux, macOS, or WSL on Windows
- Python: Version 3.6+ on control node and all managed nodes
- SSH Access: Passwordless SSH key authentication to managed hosts
- Network Connectivity: Control node can reach managed nodes on SSH port (22)

Installation steps have been discussed and implemented in previous class.

SSH KEY AUTHENTICATION SETUP

- Step 1: Generate SSH Key Pair
  - ssh-keygen -t rsa -b 4096 -C "your_email@domain.com"
- Step 2: Copy Public Key to Managed Hosts
  - ssh-copy-id username@remote_host_ip
- Step 3: Set Correct Permissions
  - ssh username@remote_host "chmod 700 ~/.ssh && chmod 600 ~/.ssh/authorized_keys"
- Step 4: Test Passwordless Connection
  - ssh username@remote_host

INVENTORY FILE CONFIGURATION

- The inventory file is Ansible's address book, defining which hosts to manage and how to group them. It supports multiple formats such as INI and yaml.
- Common Inventory Variables
  - ansible_host: Actual IP address or hostname to connect to
  - ansible_user: SSH username for connection
  - ansible_port: SSH port (default: 22)
  - ansible_ssh_private_key_file: Path to SSH private key

```
# /etc/ansible/hosts or custom inventory file

[webservers]
web1.example.com
web2.example.com ansible_host=192.168.1.10

[databases]
db1.example.com ansible_user=dbadmin
db2.example.com ansible_port=2222

[production:children]
webservers
databases

[all:vars]
ansible_ssh_private_key_file=~/.ssh/id_rsa
```

CONNECTIVITY TEST WITH PING MODULE

Ansible's ping module is not an ICMP ping, it's a connectivity and Python verification test that:

- Tests SSH connectivity to managed hosts
- Verifies Python interpreter availability
- Confirms Ansible can execute modules on remote systems
- Returns success/failure status with diagnostic information

```
# Ping all hosts in inventory
ansible all -m ping

# Ping specific host group
ansible webservers -m ping

# Ping with custom inventory file
ansible all -i custom_inventory.ini
-m ping

# Ping specific hosts
ansible
web1.example.com,web2.example.com -
m ping
```

COMMON CONNECTIVITY PROBLEMS

Issue	Symptom	Solution
SSH key not configured	Password prompt appears	Set up SSH key authentication
Wrong inventory host entry	Host unreachable error	Verify hostname/IP in inventory
Firewall blocking SSH	Connection timeout	Open port 22 on managed hosts
Python not installed	Module execution failure	Install Python on managed nodes

THANK YOU