

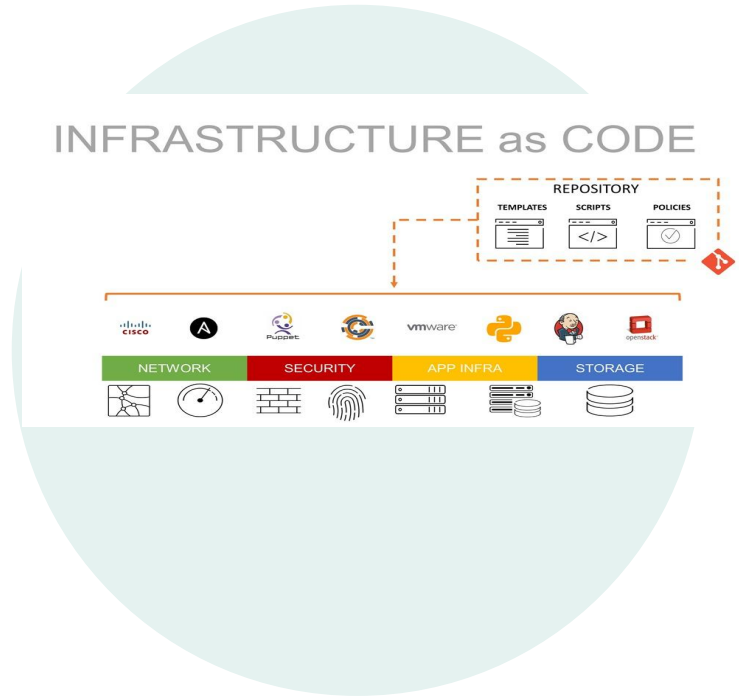


Configuration Management

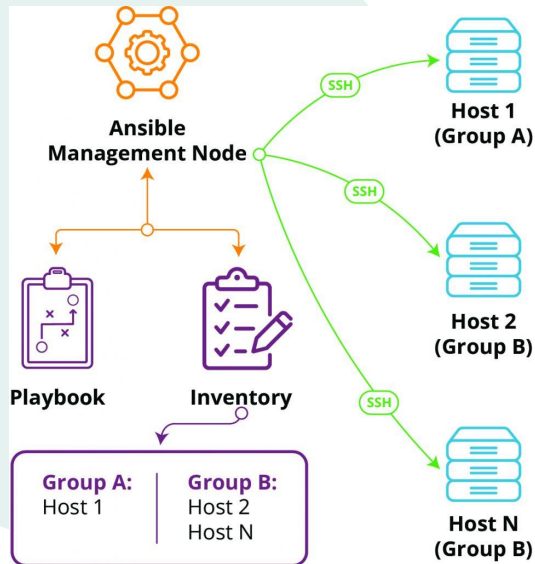
IaC & Ansible

Introduction to Infrastructure as Code (IaC)

Infrastructure as Code (IaC) is a modern approach to managing IT infrastructure through code rather than manual processes. **Significance:** It enables automation, consistency, and scalability in software development. **Engagement:** How do you perceive the role of IaC in your projects? Share your thoughts!



Overview of Ansible and Its Components



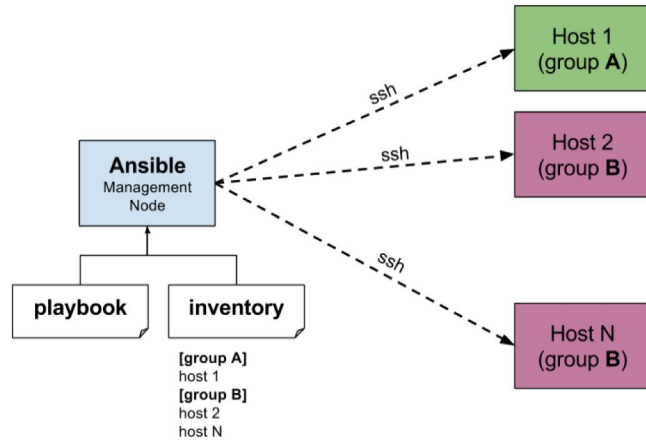
Ansible is a powerful and versatile tool for implementing **Infrastructure as Code (IaC)**, which streamlines the management of IT infrastructure. It consists of several key components:

- **Modules:** These are reusable scripts designed for executing specific tasks efficiently, allowing for modular automation;
- **Inventory:** This is a comprehensive list of managed nodes that Ansible interacts with, enabling precise control over the infrastructure;
- **Playbooks:** These are YAML files that define automation tasks in a structured manner, outlining the steps required to achieve desired configurations and deployments.

Setting Up Ansible

In this section, we'll cover the installation and initial configuration of Ansible.

- **Operating System:** Linux, macOS, or Windows (via WSL)
- **Python:** Required for Ansible to function



Writing Playbooks for Configuration Management

"Ansible playbooks are vital for efficient configuration management.

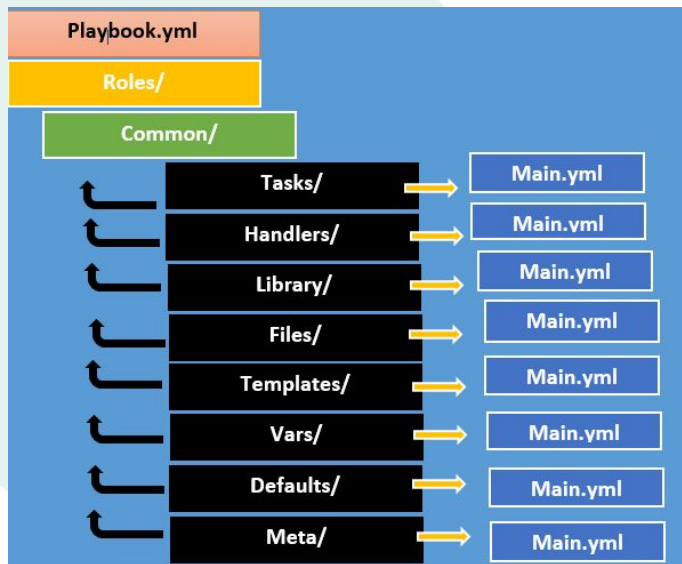
- **Format:** YAML, user-friendly and straightforward.
- **Components:** Tasks, handlers, and variables essential for operation."

```
1  ---
2  - hosts: all
3    become: yes
4    gather_facts: yes
5
6  tasks:
7    - name: Create centos folder
8      file:
9        path: /test_centos
10       state: directory
11       when: ansible_facts['distribution']=="CentOS"
12
13    - name: Create ubuntu folder
14      file:
15        path: /test_ubuntu
16        state: directory
17        when: ansible_facts['distribution']=="Ubuntu"
18
19    - name: Create debian folder
20      file:
21        path: /test_debian
22        state: directory
23        when: ansible_facts['distribution']=="Debian"
24
```

Advanced Configuration Management Techniques

We will explore advanced techniques to enhance playbook functionality.

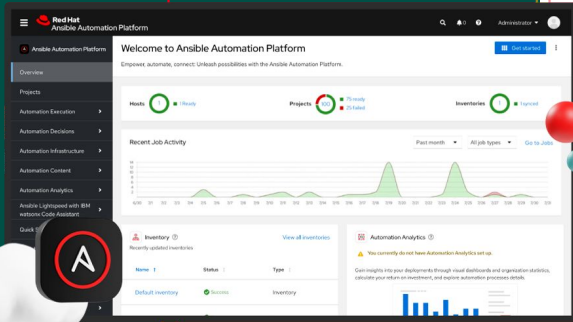
- **Roles:** Organize playbooks into reusable components
- **Templates:** Dynamically generate configuration files



Automating Infrastructure with Ansible

In this slide, we will demonstrate how to automate infrastructure deployments using Ansible.

- **Provisioning servers**
- **Deploying applications**
- **Managing configurations**

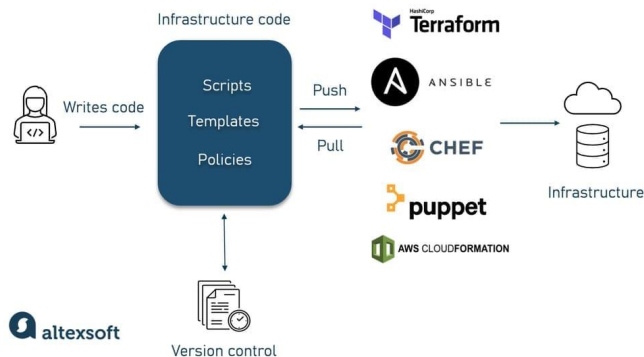


Section Recap: Infrastructure as Code and Ansible Overview

Let's summarize the main ideas discussed so far about IaC and Ansible functionalities.

- Definition and significance of **IaC**
- Overview of **Ansible** components
- Importance of **playbooks** and **automation**

HOW INFRASTRUCTURE AS CODE WORKS

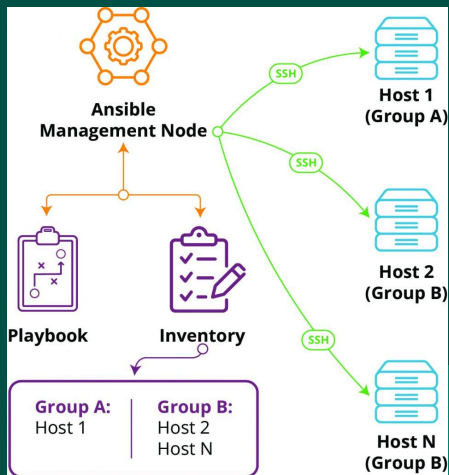


Real-World Use Cases of Ansible

Here, we will explore practical applications of Ansible in the industry.

- **Streamlined** deployment processes
- **Reduced** configuration drift
- **Enhanced** collaboration among teams

[7 Ansible Use Cases – by Spacelift](#)



Mind Map

Ansible Use Cases



Configuration management



CI/CD & application deployment



Cloud provisioning & management



Network automation



Security & compliance automation



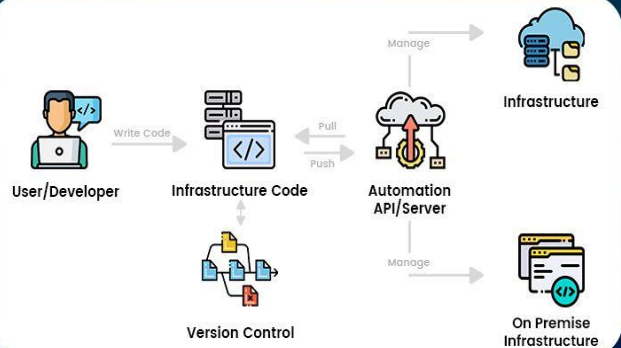
Disaster recovery automation



Complex workflow automation

spacelift

IaC Workflow



www.radixweb.com



Best Practices for Using Ansible

To maximize efficiency and readability in your projects, consider implementing these best practices that can make a significant difference:

- Use **clear** and descriptive naming conventions that accurately reflect the content and purpose of your files, making it easier for anyone to understand their function at a glance.
- Organize files **logically** by grouping related items together and establishing a consistent structure that allows for quick navigation and retrieval of information.
- Include **comments** for clarity throughout your code or documentation, providing helpful explanations that guide users and collaborators through your thought process and decisions.

A hand with the index finger pointing upwards towards a blue rectangular button with rounded corners. The button has the text 'BEST PRACTICE' in white, bold, uppercase letters. The background is dark blue with a subtle gradient.

BEST PRACTICE

Conclusion and Q&A

In conclusion, we have covered:

- The fundamentals of **Infrastructure as Code**
- Ansible's **components** and setup
- Writing **playbooks** and automating infrastructure

Engagement: Now, let's open the floor for any questions you might have! Feel free to clarify any doubts or share your insights on the material presented.

INFRASTRUCTURE as CODE

