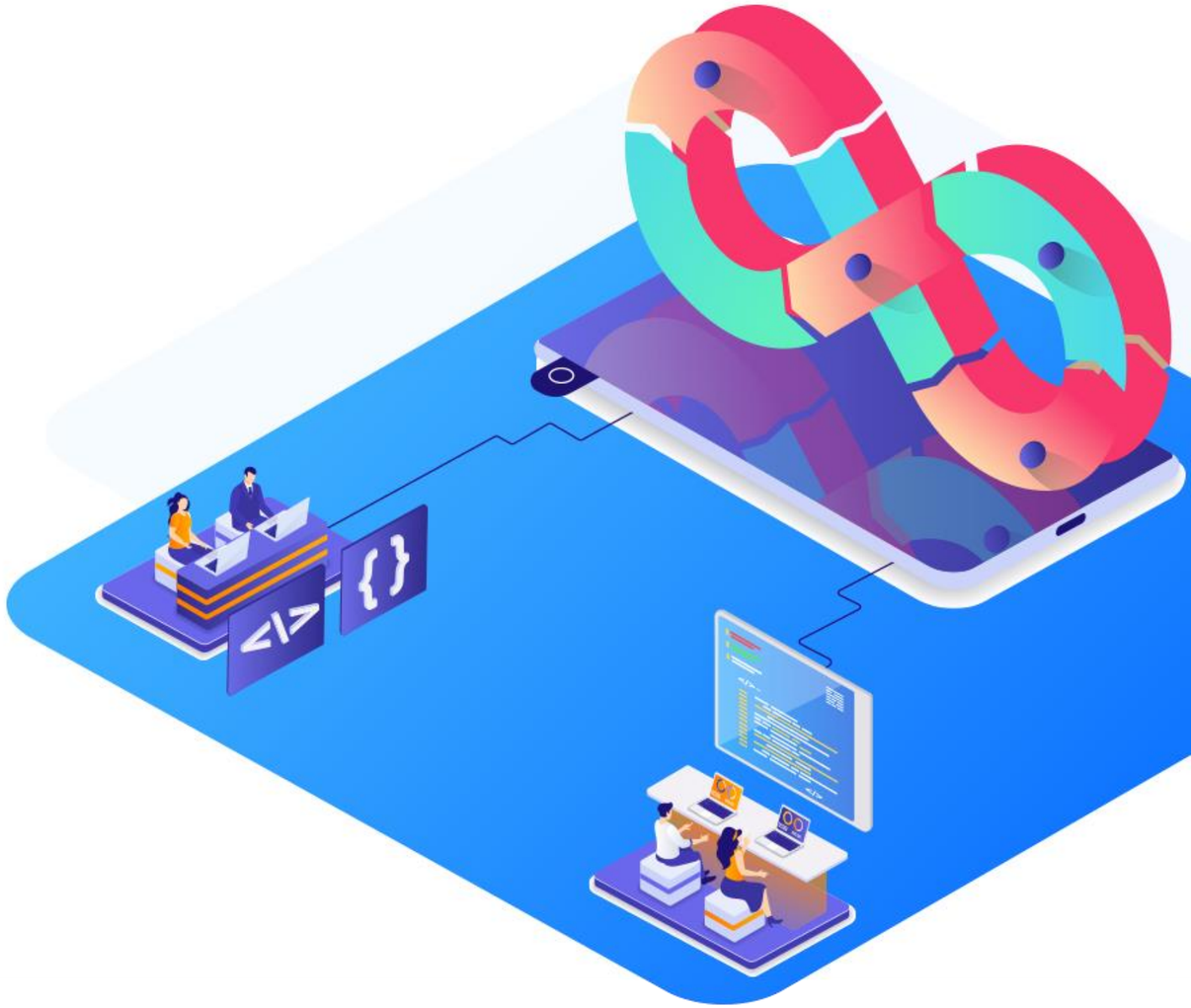
Configuration Management with Ansible and Terraform



Ansible Configuration

Learning Objectives

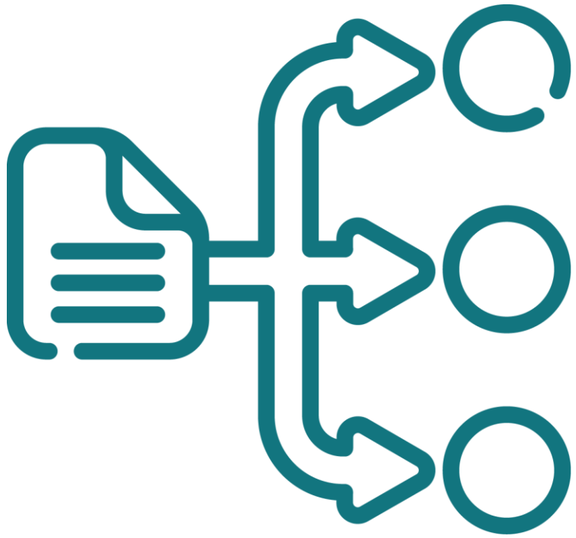
By the end of this lesson, you will be able to:

- Illustrate the fundamentals of Ansible inventory to effectively categorize and organize the Ansible resources
- Identify the different types of Ansible inventory to help organizations allocate resources more optimally, ensuring efficient infrastructure management
- Describe the concept of inventory file formats to manage and maintain inventories efficiently, ensuring effective and error-free infrastructure management with Ansible
- Demonstrate how to add variables to inventory to help users tailor configurations and actions to specific hosts or groups, enhancing flexibility and control

Managing Ansible Inventory

Ansible Inventory

The Ansible inventory file stores the hostnames or private IP addresses of all the nodes or hosts connected to the Ansible server.



The default location of the Ansible inventory file is **/etc/ansible/hosts**.

Ansible Inventory

Example of an inventory file in INI format:

```
 mail.example.com


[webservers]
foo.example.com
bar.example.com

[Observers]
one.example.com
two.example.com
three.example.com
```

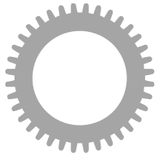
Ansible Inventory

Example of an inventory file in YAML format:

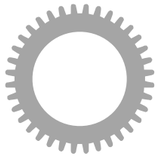
```
all:
  hosts:
        mail.example.com:
   children:
        webservers:
             hosts:
                  foo.example.com:
                  bar.example.com:
          dbservers:
              hosts:
                  one.example.com:
                  two.example.com:
                three.example.com:
```

Ansible Inventory

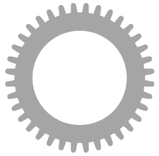
Key points regarding Ansible inventory are:



The place where hosts or groups are defined is known as the inventory. Ansible runs against various managed nodes, or **hosts**, in the infrastructure all at once.



After defining the inventory, users can use patterns to select the hosts or groups against which they want Ansible to run.



Users can specify a different inventory file at the command line using the **-i <path>** option.

Types of Ansible Inventory

There are two types of Ansible inventory:

Static Inventory:
A static inventory file is a plain text file that contains a list of managed hosts defined by hostnames or IP addresses under a host group.

Dynamic Inventory:
A script in a high-level language is called a dynamic inventory. It is helpful in cloud setups like AWS, where server IPs change with stops and restarts.

Ansible Inventory

Example of a static inventory file in YAML:

```
[webservers]
 173.82.115.165

[database_servers]
173.82.220.239

[datacenter:children]
webservers
database_servers
```

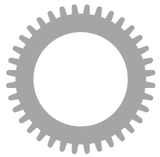
Note

The inclusion of managed hosts in a host group is optional. Users can simply use their hostnames or IP addresses to compile a list.

Host and Groups

Hosts and groups are the nodes connected to the Ansible server.

A host is a single node connected to the Ansible server.



Groups are collections of hosts named as a single entity.

Adding Variables to Inventory

1. Inventory stores variable values related to a specific host or group.

2. Inventory can contain variable values associated with a specific host or group.

3. Users can add additional managed nodes to their Ansible inventory; then, they can also keep the variables in distinct host and group variable files.

Adding Variables to Inventory

Example of assigning a variable to one machine: host variables

```
 [atlanta]
host1 http_port=80 maxRequestsPerChild=808
host2 http_port=303maxRequestsPerChild=909
```

Adding Variables to Inventory

Example of assigning a variable to one machine: host variables

YAML format

```
atlanta:
    hosts:
        hosts1:
            http_port  80
            maxRequestsPerChild: 808
         host2:
            http_port: 303
            maxRequestsPerChild: 909
```

Adding Variables to Inventory

Example of assigning a variable to one machine: group variables

INI format

```
[atlanta]
 host1
 host2

[atlanta:vars]
 ntp_server=ntp.atlanta.example.com
 proxy=proxy.atlanta.example.com
```

Adding Variables to Inventory

Example of assigning a variable to one machine: group variables

YAML format

```
 atlanta:
     hosts:
          host1:
          host2:

     vars:
          ntp_server:
ntp.atlanta.example.com
          proxy: proxy.atlanta.example.com
```

Organizing Host and Group Variables

1. Variables can be stored in the main inventory file. However, maintaining separate files for host and group variables might simplify the organization of variable values.

2. Host and group variable files require YAML syntax.

3. A file can have extensions such as **.yml**, **.yaml**, and **.json**, or it may have no extension at all. Both types of files are valid.

Inventory File Formats

Inventory File Formats: INI and YAML

Ansible environment supports two types of formats.

1. INI
2. YAML

INI and YAML

Example of inventory in INI format:

```
server1.example.com
server1.example.com
[linux]
server3.example.com
server4.example.com
[windows]
server5.example.com
server6.example.com
```

INI and YAML

Example of inventory in YAML format:

```
linux:
      hosts:
            server1.example.com:
            server2.example.com:
            server3.example.com:
 windows:
       hosts:
            server1.example.com:
            server2.example.com:
            server3.example.com:
```

Assisted Practice

Creating static host inventory **Duration: 15 Min.**

Problem Statement:

You have been assigned a task to create a static host inventory for managing and automating infrastructure tasks efficiently across multiple servers using Ansible.

Outcome:

You will have a robust static host inventory setup in Ansible, facilitating efficient infrastructure management and automation across multiple servers.

Note: Refer to the demo document for detailed steps: 01_Creating_Static_Host_Inventory

Assisted Practice: Guidelines

Steps to be followed:

1. Generate SSH key pair on the main node
2. Copy the SSH key on the other two nodes
3. Update the inventory or host file with the host IP address
4. Establish connectivity between the hosts specified in the host file and the Ansible server

Assisted Practice

Working with host and groups **Duration: 15 Min.**

Problem Statement:

You have been tasked to work with hosts both individually and in groups for streamlined management and deployment across your network infrastructure.

Outcome:

The management and deployment of hosts across your network infrastructure will become more streamlined, efficient, and scalable.

Note: Refer to the demo document for detailed steps: 02_Working_with_Hosts_and_Groups

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to be followed:

1. List the hosts present in your host file
2. Create a file on the host
3. Remove the file from the host
4. Create and remove the file on the hosts using the group
5. Update the hosts using the group

Quick Check

You are managing a production environment with multiple servers. You need to list all the hosts in your Ansible inventory to ensure they are correctly configured before running any playbooks. Which command should you use?

A. ansible all -m ping

B. ansible-inventory --list

C. ansible-playbook --list-hosts

D. ansible --list

Quick Check

Before deploying a critical update, you want to ensure your Ansible inventory is correctly structured. Which command helps you visualize the structure of your inventory?

A. ansible-playbook --check

B. ansible-inventory --graph

C. ansible all -m setup

D. ansible-playbook --syntax-check

Key Takeaways

- The Ansible inventory file stores the hostnames or private IP addresses of all the nodes or hosts connected to the Ansible server.
- The place where hosts or groups are defined is known as the inventory. Ansible runs against various managed nodes, or **hosts**, in the infrastructure all at once.
- A file can have extensions such as **.yml**, **.yaml**, and **.json**, or it may have no extension at all. Both types of files are valid.
- After defining the inventory, users can use patterns to select the hosts or groups against which they want Ansible to run.

Simplifying Inventory with Ranges

Duration: 25 Min.

Project agenda: To perform simplification of inventory with ranges for defining multiple hosts in a more concise and manageable way, especially when dealing with many hosts

Description: As a DevOps engineer at a tech company, you manage a large fleet of servers. In your current project, you maintain a cluster of 100 web servers needing frequent updates. Using range patterns like web[01:100].example.com in Ansible inventory files, you will simplify and streamline inventory management. This reduces redundancy, makes files easier to manage, and ensures efficient, consistent updates across all servers with minimal effort.

LESSON-END PROJECT

Simplifying Inventory with Ranges

Duration: 25 Min.

Perform the following:

1. Install Ansible on the server
2. Create an inventory file
3. Test the inventory
4. Create a simple playbook
5. Run the playbook

Expected Deliverables: A simplified Ansible inventory file using ranges to define multiple hosts, a working Ansible playbook that runs tasks against the defined inventory groups, and successful execution and verification of the playbook with appropriate logs.

LESSON-END PROJECT

Thank you