# Ansible Documentation for Cisco Network Automation

### Overview

Ansible is an open-source automation tool that simplifies configuration management, application deployment, and task automation. This documentation provides a comprehensive guide to setting up Ansible for managing Cisco network devices such as routers and switches.

### Table of Contents

1. \*\*Prerequisites\*\*

2. \*\*Installing Ansible\*\*

3. \*\*Setting Up Directory Structure\*\*

4. \*\*Creating Inventory File\*\*

5. \*\*Configuring Ansible Control Node\*\*

6. \*\*Creating Playbooks\*\*

7. \*\*Executing Playbooks\*\*

8. \*\*Example Playbooks\*\*

9. \*\*Advanced Topics\*\*

10. \*\*Troubleshooting\*\*

### 1. Prerequisites

Before you begin, ensure the following prerequisites are met:

- Ansible control node (can be a Linux or macOS machine)

- Cisco routers and switches accessible via SSH or Telnet

- Knowledge of Cisco IOS command-line interface (CLI)

- Basic understanding of YAML syntax

### 2. Installing Ansible

Ansible can be installed on a variety of platforms. Here’s how to install it on a Debian-based Linux system (e.g., Ubuntu):

```bash

sudo apt update

sudo apt install ansible

```

Verify the installation:

```bash

ansible --version

```

### 3. Setting Up Directory Structure

Create a directory structure for your Ansible project:

```bash

mkdir ansible

cd ansible

mkdir inventory playbooks backup

touch ansible.cfg

```

### 4. Creating Inventory File

Edit the Ansible inventory file to include your Cisco devices:

\*\*File: `inventory/hosts`\*\*

```ini

[routers]

router1 ansible\_host=192.168.1.1

router2 ansible\_host=192.168.1.2

[switches]

switch1 ansible\_host=192.168.1.3

switch2 ansible\_host=192.168.1.4

```

Replace IP addresses (`ansible\_host`) with actual IP addresses or hostnames of your Cisco devices.

### 5. Configuring Ansible Control Node

Edit the Ansible configuration file (`ansible.cfg`) to define settings for your project:

\*\*File: `ansible.cfg`\*\*

```ini

[defaults]

inventory = ./inventory/hosts

remote\_user = your\_remote\_username

private\_key\_file = /path/to/your/private\_key

```

Adjust `remote\_user` and `private\_key\_file` according to your environment. Replace `/path/to/your/private\_key` with the path to your SSH private key if using SSH authentication.

### 6. Creating Playbooks

#### Network Setup Playbook (`network\_setup.yml`)

\*\*File: `playbooks/network\_setup.yml`\*\*

```yaml

---

- name: Configure Cisco Network Devices

hosts: routers:switches

gather\_facts: no

become: yes

tasks:

# Insert tasks from previously provided playbook example (network\_setup.yml)

# Ensure to customize IP addresses, VLANs, and other settings as needed.

```

#### Backup Playbook (`backup.yml`)

\*\*File: `playbooks/backup.yml`\*\*

```yaml

---

- name: Backup Cisco Network Devices

hosts: routers:switches

gather\_facts: no

become: yes

tasks:

# Insert tasks from previously provided playbook example (backup.yml)

# Adjust file paths and configurations according to your environment.

```

### 7. Executing Playbooks

Run the playbooks using the `ansible-playbook` command:

```bash

ansible-playbook playbooks/network\_setup.yml --ask-pass --ask-become-pass

ansible-playbook playbooks/backup.yml --ask-pass --ask-become-pass

```

Replace `--ask-pass` with `--private-key` if using SSH keys for authentication. Adjust `--ask-become-pass` with `--become-user` and `--become-method` based on your sudo setup.

### 8. Example Playbooks

The provided example playbooks configure basic network settings and perform backups. Customize these playbooks for your specific Cisco network requirements.

### 9. Advanced Topics

Explore advanced Ansible features such as roles, Jinja templating, and Ansible Galaxy for managing larger and more complex network infrastructures.

### 10. Troubleshooting

- Ensure proper connectivity between Ansible control node and Cisco devices.

- Check SSH/Telnet access and credentials.

- Review Ansible output for errors or warnings (`ansible-playbook` logs).

---

This documentation provides a foundational understanding of using Ansible to automate Cisco network device configurations. Customize and expand upon these examples to suit your organization’s specific network infrastructure needs.