

# Ansible Network Automation

This playbook describes the process of using Ansible to automate network configuration tasks, which helps in maintaining consistency across multiple devices and reduces the likelihood of human error.

## Step 1: Prerequisites

Ensure that Ansible is installed on your control node (your workstation). Confirm that all target network devices are reachable from this node and have Python installed if necessary. Setup SSH keys for a seamless authentication process between the control node and the network devices.

## Step 2: Inventory Setup

Create an inventory file in INI or YAML format that lists all the hostnames or IP addresses of the network devices you intend to manage with Ansible. Group the devices by type or function if needed for more structured management.

## Step 3: Playbook Creation

Write an Ansible playbook that defines the desired state of your network devices. This includes tasks that specify which modules to run along with the necessary variables and settings.

## Step 4: **Variables Assignment**

Assign variables for any dynamic content such as device-specific configuration parameters. These can be defined within the playbook itself or separate variable files for better organization.

## Step 5: **Dry Run**

Initiate a 'dry run' by executing your playbook with the `--check` flag. This will simulate changes without actually applying them, allowing you to verify the expected outcomes.

## Step 6: **Execution**

Run the playbook without any test flags to apply the configurations. Monitor the output for any errors or changes that did not behave as expected.

## Step 7: **Review**

Verify the changes by logging into the network devices or using Ansible to fetch their current state. Ensure that the configurations match the desired state defined in the playbook.

# General Notes

## **Modules**

Use the Ansible documentation to choose the correct modules for your device type and configuration tasks. Modules can vary between network vendors.

## Backups

Before applying changes, consider using Ansible to back up current configurations to prevent any potential data loss.

## Version Control

Keep your playbooks and variable files under a version control system. This provides a history of changes and the ability to revert to previous versions if necessary.

Powered by: **PlaybookWriter.com**