

# Write and execute a simple Ansible playbook to install a package on remote servers

---

## Table of Contents

1. [Introduction](#)
  2. [Problem Statement](#)
  3. [Prerequisites](#)
    - [Software Required](#)
    - [Hardware Requirement](#)
  4. [Writing the Ansible Playbook](#)
  5. [Running the Playbook](#)
  6. [Verifying Installation](#)
  7. [Supported Reference](#)
- 

## Introduction

This guide will explain how to create and run a simple Ansible playbook to install **Chocolatey**, a popular package manager for Windows, on a remote Windows machine. Ansible playbooks are written in YAML format and allow you to automate tasks such as package installation, configuration, and server management.

---

## Problem Statement

Managing packages across multiple remote Windows servers manually can be time-consuming and error-prone. Ansible can automate the process, making it easier to install, update, or remove software packages on multiple servers from a single control node.

In this guide, we'll create an Ansible playbook that installs Chocolatey on a remote Windows machine.

---

## Prerequisites

Completion of all previous lab guides (up to Lab Guide-01) is required before proceeding with Lab Guide-02.

Before proceeding, ensure that you have completed the following:

### Software Required

- **Windows Subsystem for Linux (WSL):** Installed on your control node.
- **Python 3.8 or later:** Installed on your WSL environment.
- **Ansible 2.9 or later:** Installed and configured.
- **Windows Remote Server (with WinRM configured):** A Windows machine that Ansible will manage.

### Hardware Requirement

- **Control Node:** A Windows machine with WSL enabled.

- **Target Node:** A remote Windows machine where Chocolatey will be installed.
- 

## Writing the Ansible Playbook

The Ansible playbook will contain the steps required to install Chocolatey on the remote Windows host. Follow these steps to create the playbook:

### Step 1: Create the Ansible Playbook

1. In your WSL environment, create a new directory for the playbook:

```
mkdir ~/ansible_playbooks  
cd ~/ansible_playbooks
```



2. Create a new file named `install_chocolatey.yml`:

```
nano install_chocolatey.yml
```

3. Add the following YAML content to define the playbook:

```
---  
- name: Install Chocolatey on Windows  
  hosts: windows  
  tasks:  
    - name: Ensure Chocolatey is installed  
      win_chocolatey:  
        name: chocolatey  
        state: present
```



This playbook does the following:

- **name:** Describes the task ("Install Chocolatey on Windows").
- **hosts:** Specifies the group of remote machines to target (in this case, `windows` as defined in the Ansible inventory).
- **tasks:** Lists the tasks to execute. Here, we are using the `win_chocolatey` Ansible module to ensure that Chocolatey is installed.

4. Save the file by pressing `CTRL + O` and then `CTRL + X` to exit the editor.
- 

## Running the Playbook

Now that the playbook is written, let's run it to install Chocolatey on the remote Windows server.

## Step 1: Execute the Playbook

Run the following command to execute the `install_chocolatey.yml` playbook:

```
ansible-playbook install_chocolatey.yml
```

## Step 2: Verify the Output

After running the command, you should see output similar to the following, indicating that Ansible has connected to the remote Windows machine and executed the task:

```
PLAY [Install Chocolatey on Windows]
*****
TASK [Ensure Chocolatey is installed]
*****
ok: [<ip-address>]
PLAY RECAP
*****
<ip-address>          : ok=1    changed=0    unreachable=0    failed=0
```



The `ok=1` indicates that the task has successfully executed without making changes since Chocolatey may already be present. If the playbook installs Chocolatey for the first time, `changed=1` will appear instead.

---

## Verifying Installation

After running the playbook, verify that Chocolatey was installed successfully on the remote Windows machine:

### Step 1: Check Chocolatey on the Remote Server

1. Log in to the remote Windows machine.
2. Open a PowerShell window with Administrator access.
3. Run the following command to check the version of Chocolatey installed:

```
choco --version
```



If Chocolatey was installed successfully, the version number will be displayed.

## Supported Reference

For further reading and references on Ansible and managing Windows hosts, visit:

- [Ansible Documentation](#)
  - [Chocolatey Documentation](#)
  - [WSL Documentation](#)
-