# Project Report

# Ansible Playbook Implementation

D.H.D.Madhushani

# Table of Contents

# 1. Introduction

**Project Title**: Automating Tasks with Ansible Playbooks

This project focuses on leveraging Ansible, a powerful IT automation tool, to simplify system administration tasks and demonstrate its core features. Ansible playbooks, written in YAML, allow users to automate various tasks such as software installation, service configuration, and file management.

The project involves creating and executing multiple Ansible playbooks to explore key concepts, including idempotence, variables, loops, and conditionals. These playbooks automate real-world scenarios, such as installing and configuring a web server, demonstrating the idempotence of tasks, and dynamically managing operations based on conditions.

By completing this project, we gain hands-on experience in writing efficient playbooks, understanding Ansible modules, and applying automation to ensure consistency and reliability across tasks. It showcases how Ansible can streamline repetitive processes, saving time and minimizing errors in system administration.

**Objective**:

- To demonstrate the practical application of Ansible playbooks.
- To automate the deployment of services, configuration management, and task execution using Ansible.
- To understand and implement concepts like idempotence, variables, loops, and conditionals in Ansible.

# 2. Project Description

## 2.1 Overview of Ansible

**What is Ansible?**

Ansible is an open-source IT automation tool that simplifies configuration management, application deployment, and other tasks through its YAML-based playbooks.

**Why Ansible?**

- Easy to learn.
- Agentless.
- Supports idempotence to prevent redundant changes.

# 3. Project Details

## 3.1 Playbook 1: Web Server Automation

**Playbook Name**: webserver.yml

webserver.yml automates the setup and configuration of a web server. It installs Apache, ensures the service is running and enabled at startup, and creates a simple index.html file with "Hello World!" content. This playbook demonstrates Ansible's capability to manage software installation and configuration effectively.

- **Steps**:
    1. **Install Apache**: Ensure httpd is installed.
    2. **Start Service**: Enable and start the Apache service.
    3. **Configure File**: Create an index.html file with the text "Hello World!".

- **Modules Used**:
    - ansible.builtin.package: To install the Apache web server.
    - ansible.builtin.service: To manage the Apache service.
    - ansible.builtin.lineinfile: To create and write to the index.html file.

## 3.2 Playbook 2: Idempotence Demonstration

Playbook Name: idempotent.yml

idempotent.yml focuses on the concept of idempotence in Ansible. It uses the ansible.builtin.shell module to write "hello" to a file (/tmp/hello.txt). Running the playbook

multiple times highlights how the shell module always results in a "changed" status, showcasing the importance of using modules that support state checking.

- **Steps**:
    1. Use the ansible.builtin.shell module to create a file at /tmp/hello.txt containing the text "hello".
    2. Run the playbook multiple times to observe the changed status behavior.
- **Key Concept**:
    - Idempotence ensures tasks do not produce unintended changes when executed multiple times.

## 3.3 Playbook 3: Variables, Loops, and Conditionals

- **Objective**: Use variables, loops, and conditionals to dynamically manage tasks.
- **Playbook Name**: vcl.yml

vcl.yml demonstrates the use of variables, loops, and conditionals in Ansible. It defines a list of numbers (nums), iterates through the list using a loop, and applies a conditional statement to display numbers greater than 10. This playbook emphasizes Ansible's dynamic and flexible approach to task execution.

- **Steps**:
    1. Define a list of numbers as a variable (nums).
    2. Iterate over the list using a loop.
    3. Use a conditional to display only numbers greater than 10.
- **Modules Used**:
    - ansible.builtin.debug: To display output dynamically.

# 4. Implementation

## 4.1 Tools and Environment

- **Operating System**: Windows Subsystem for Linux (WSL).
- **Ansible Version**: [Specify the installed version using ansible --version].
- **Editor**: Nano/VSCode for writing YAML files.
- **Host Inventory**:
    - localhost ansible_connection=local.

## 4.2 Command Execution

- **Running Playbooks**:

ansible-playbook -i inventory webserver.yml

ansible-playbook -i inventory idempotent.yml

ansible-playbook -i inventory vcl.yml

# 5. Results

- **Web Server Automation**:
    - The Apache web server was successfully installed and configured.
    - Verified the index.html file using curl.
- **Idempotence**:
    - Re-running idempotent.yml always showed a changed status because of the shell module.
- **Variables, Loops, and Conditionals**:
    - Displayed only numbers greater than 10 from the defined list.

# 6. Conclusion

This project provided hands-on experience with Ansible playbooks and demonstrated how to automate system administration tasks effectively. The playbooks covered key Ansible concepts and showcased the power of automation in real-world scenarios.

# 7. References


1. https://www.linkedin.com/learning/learning-ansible-24687086