Prog No 7:

Configuration Management with Ansible: Basics of Ansible: Inventory, Playbooks, and Modules, Automating Server Configurations with Playbooks, Hands-On: Writing and Running a Basic Playbook

VIRTUAL BOX INSTALLATION.

1. Search – Turn Windows features ON or OFF
2. CHECK BOX – Virtual Machine Platform and Windows Subsystem for Linux.
3. Click on OK

SEARCH – MICROSOFT STORE

1. Type Ubuntu in the search box.
2. Select the Latest version
3. Click on **GET** (IF NOT INSTALLED), else the **OPEN** button reflects
4. After downloading OPEN
5. TERMINAL OPEN.
6. WILL ASK FOR USERNAME AND PASSWORD.

What Is Ansible?

Ansible is an **open-source automation tool** used for:

- **Configuration Management** (e.g., install software, edit config files)
- **Application Deployment**
- **Orchestration** (e.g., restart services in order)
- **Provisioning** (e.g., set up new servers in the cloud)

YAML stands for **"YAML Ain't Markup Language"** (originally **"Yet Another Markup Language"**). It's a **human-readable data serialization language**, often used for:

- Configuration files (e.g., Ansible playbooks, Docker Compose, Kubernetes)
- Data exchange between languages with different data structures
- Application settings

Key Features of YAML:

- Easy to read and write
- Uses **indentation** (spaces) to define structure (not tabs)
- Avoids brackets {} and quotes when not needed
- Based on **key-value pairs**, **lists**, and **nesting**

Key-Value Pairs

```
name: John
age: 30
is_student: false
```

Lists

```
fruits:
  - apple
 - banana
 - orange
```

Nested Structures

```
person:
  name: Alice
  contact:
    email: alice@example.com
    phone: 1234567890
```

Experiment Steps

ansible-playbook playbooks/apache-install.yml

steps:

1. sudo apt update
2. sudo apt upgrade -y
3. sudo apt install ansible -y
4. ansible --version

Step 1: Create an Inventory File

1. Open your text editor to create a file called hosts.ini:
2. nano hosts.ini
3. Add the following content to define the local host:

```
[local]
 localhost ansible_connection=local
```

Explanation:

- [local] is a group name.
- localhost is the target host.
- ansible_connection=local tells Ansible to execute commands on the local machine without SSH.

4.Save the file by pressing Ctrl+O then Enter, and exit with Ctrl+X.

Only for reference

Inventory File Name	Command
hosts.ini	ansible-playbook -i hosts.ini setup.yml
inventory	ansible-playbook -i inventory setup.yml
prod_hosts.txt	ansible-playbook -i prod_hosts.txt deploy.yml

Term	Meaning
hosts.ini	A file that lists your inventory (servers or groups of servers)
nano, vim, notepad	Editors, you use to create/edit hosts.ini

Step 2: Create the Playbook File

1. Open your text editor to create a file called setup.yml:
2. nano setup.yml

CODE:

```
- name: Basic Server Setup
```

```
hosts: local
become: true  # Optional: if your tasks need root access
tasks:
  - name: Example task
    debug:
      msg: "Hello, this is a basic setup"
```

Step 3: Execute the Playbook

In your terminal, run the following command:

ansible-playbook -i hosts.ini setup.yml

OR

ansible-playbook -i hosts.ini setup.yml --ask-become-pass

Part	Meaning
ansible-playbook	The Ansible command runs playbooks (YAML files with tasks).
-i hosts.ini	Specifies the **inventory file** (hosts.ini) that contains the list of target hosts.
setup.yml	This is the **playbook** you are running, which contains the automation tasks.
--ask-become-pass	Prompts for the **sudo password** (become password), if become: true is used in your playbook.

If you want to have a different playbook name, it is possible to have

If your playbook is named abc.yml, just run:

ansible-playbook -i hosts.ini abc.yml

ansible-playbook -i hosts.ini abc.yml --ask-become-pass

Prog No 9:

Introduction to Azure DevOps: Overview of Azure DevOps Services, Setting Up an Azure DevOps Account, and Project

Overview of Azure DevOps
Azure DevOps is a comprehensive suite of cloud-based services to support the entire software development lifecycle. It provides tools for planning, developing, testing, delivering, and monitoring applications.

Here are the primary services offered:

• **Azure Repos:**
A set of version control tools that allow you to host Git repositories or use Team Foundation Version Control (TFVC). It offers collaboration features such as pull requests, branch policies, and code reviews.

• **Azure Pipelines:**
A CI/CD service that helps automate builds, tests, and deployments. It supports multiple languages, platforms, and can run on Linux, Windows, or macOS agents.

• **Azure Boards:**
A work tracking system that helps teams manage work items, sprints, backlogs, and Kanban boards. It facilitates agile planning and reporting.

• **Azure Test Plans:**
Provides a solution for managing and executing tests, capturing data about defects, and tracking quality.

• **Azure Artifacts:**
Allows you to create, host, and share packages (such as Maven, npm, NuGet, and Python packages) with your team, integrating package management into your CI/CD pipelines.

These services integrate with each other and with popular third-party tools to create a cohesive DevOps ecosystem.

Step 1: Sign Up for an Azure DevOps Account

1. Open Your Web Browser:

   Navigate to the Azure DevOps website: https://dev.azure.com

2. Sign In or Create a Microsoft Account:

   - If you already have a Microsoft account (such as Outlook, Hotmail, or Office 365), click “Sign in”.
   - If you do not have a Microsoft account, click “Create one!” and follow the instructions to create a new Microsoft account.

Step 2: Create an Azure DevOps Organization

1. Click on the Azure DevOps organization or search for it in the search bar.
   - Click on my Azure DevOps organizations
   - Click on Create a New Organization and click on Continue
   - Enter a unique name for your organization (e.g., Your Company or your name DevOps) or MyPersonalOrg).
   - Select a Region: India. (it automatically shows India)
   - Enter the characters you see
   - Click “Continue” or “Create

Step 3. Creating an Azure DevOps Project

1. Configure Your Project:

   Project Name: Enter a descriptive name for your project (e.g., HelloDevOps).

   Description: Optionally, provide a brief description (e.g., “A sample project to demonstrate Azure DevOps services”).

2. Visibility:
   - Choose “Private” if you want to restrict access to your project.
   - Choose “Public” if you are okay with the project being accessible to anyone.
   - Advanced Options (Optional): You can choose a version control system (Git is the default) and a work item process (Agile, Scrum, or Basic). For most beginners, the defaults are recommended.
   - Click “Create

Step 4: Explore Your Project Dashboard

1.Project Overview:

Once your project is created, you will be directed to the project dashboard. Here you will see navigation options for:

- Repos: Where your code is stored.
- Pipelines: For build and release automation.
- Boards: For work tracking and agile planning.
- Test Plans: For managing and running tests.
- Artifacts: For hosting packages.

2. Familiarize Yourself with the Interface:

   Click through each section (e.g., Repos, Pipelines, Boards) to get a sense of the available features.