Ansible Playbooks

Certainly! Here are a few examples of Ansible playbooks that demonstrate different tasks and configurations:

Example 1: Basic Server Setup

This playbook installs common packages and sets up basic configurations on a group of servers.

```
---
- name: Basic Server Setup
  hosts: servers
  become: yes  # Run tasks with sudo

  tasks:
    - name: Update apt cache
      apt:
        update_cache: yes

    - name: Install necessary packages
      apt:
        name: "{{ item }}"
        state: present
      with_items:
        - vim
        - git
        - curl

    - name: Configure timezone
      timezone:
        name: America/New_York

    - name: Ensure NTP service is running
      service:
        name: ntp
        state: started
        enabled: yes
```

Example 2: Deploying a Web Application

This playbook deploys a web application (in this case, a simple HTML file) to a group of web servers.

```
---
- name: Deploy Web Application
  hosts: webservers
  become: yes  # Run tasks with sudo

  tasks:
    - name: Copy index.html to web servers
      copy:
        src: /local/path/to/index.html
        dest: /var/www/html/index.html
        owner: www-data
        group: www-data
        mode: '0644'

    - name: Ensure Apache service is running
      service:
        name: apache2
        state: started
        enabled: yes
```

Example 3: Database Server Configuration

This playbook configures a PostgreSQL database server with custom configurations and users.

```
---
- name: Configure PostgreSQL Database Server
  hosts: db_servers
  become: yes  # Run tasks with sudo

  tasks:
    - name: Install PostgreSQL database server
      apt:
        name: postgresql
        state: present

    - name: Configure PostgreSQL authentication
      template:
        src: pg_hba.conf.j2
        dest: /etc/postgresql/12/main/pg_hba.conf
        owner: postgres
        group: postgres
        mode: '0640'
      notify:
        - Restart PostgreSQL

  handlers:
    - name: Restart PostgreSQL
      service:
        name: postgresql
        state: restarted
```

Example 4: Application Deployment with Docker

This playbook deploys a Dockerized application using Ansible and Docker modules.

```
---
- name: Deploy Dockerized Application
  hosts: docker_servers
  become: yes  # Run tasks with sudo

  tasks:
    - name: Pull Docker image
```

```
      docker_image:
        name: myapp_image
        source: pull

    - name: Run Docker container
      docker_container:
        name: myapp_container
        image: myapp_image
        state: started
        ports:
          - "8080:80"
```

Example 5: Continuous Integration Setup

This playbook sets up a Jenkins server for continuous integration.

```
---
- name: Setup Jenkins CI Server
  hosts: jenkins_server
  become: yes  # Run tasks with sudo

  tasks:
    - name: Install Java
      apt:
        name: openjdk-11-jdk
        state: present

    - name: Add Jenkins repository key
      apt_key:
        url: https://pkg.jenkins.io/debian-stable/jenkins.io.key
        state: present

    - name: Add Jenkins repository
      apt_repository:
        repo: deb https://pkg.jenkins.io/debian-stable binary/
        state: present
        filename: jenkins

    - name: Install Jenkins
      apt:
        name: jenkins
        state: present
```

These examples illustrate different use cases of Ansible playbooks, from basic server setup and application deployment to configuring specialized services like databases and CI servers. Ansible's flexibility and declarative approach make it suitable for automating a wide range of IT tasks and configurations in both small-scale and large-scale environments.