  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
**Step-by-step guidelines** to implement all three scenarios (Continuous Deployment, IaC, and Self-Healing Services) **locally in Visual Studio Code**. This will include **prerequisites, folder setup, playbooks, and pipeline simulation**.

**Prerequisites**

Before you start, make sure your local machine (where VS Code is installed) has the following:

1. **Operating System**: Ubuntu/Debian recommended (you can also use WSL2 on Windows).
2. **VS Code** installed.
3. **Git** installed:

sudo apt update

sudo apt install git -y

git --version

1. **Ansible** installed:

sudo apt update

sudo apt install ansible -y

ansible --version

1. **Optional (for Jenkins pipeline simulation)**: Jenkins installed locally or you can simulate via scripts.

**Step 1: Create Project Structure in VS Code**

Open VS Code and create a folder, for example:

ci-cd-ansible/

├── inventory.ini

├── ansible.cfg

├── app/

│ └── index.html

├── playbooks/

│ ├── deploy\_app.yml

│ ├── infrastructure.yml

│ └── self\_healing\_service.yml

├── scripts/

│ └── run\_pipeline.sh

└── .git/

**Step 2: Configure Inventory**

**inventory.ini** (local testing):

[staging]

localhost ansible\_connection=local

[webservers]

localhost ansible\_connection=local

Replace localhost with real server IPs if you want to test on remote servers.

**Step 3: Create Ansible Configuration**

**ansible.cfg**:

[defaults]

inventory = inventory.ini

host\_key\_checking = False

retry\_files\_enabled = False

**Step 4: Scenario 1 - Continuous Deployment of Web Applications**

**Playbook: deploy\_app.yml**

---

- name: Deploy web application

hosts: staging

become: yes

tasks:

- name: Install Nginx

apt:

name: nginx

state: present

- name: Copy application files

copy:

src: ../app/

dest: /var/www/html/

owner: www-data

group: www-data

mode: '0755'

- name: Ensure Nginx is running

service:

name: nginx

state: started

enabled: yes

**App Code: app/index.html**

<!DOCTYPE html>

<html>

<head>

<title>Local CI/CD Web App</title>

</head>

<body>

<h1>Hello from Ansible-deployed web server!</h1>

</body>

</html>

**Step 5: Scenario 2 - Infrastructure as Code (IaC)**

**Playbook: infrastructure.yml**

---

- name: Configure infrastructure

hosts: staging

become: yes

tasks:

- name: Update apt packages

apt:

update\_cache: yes

- name: Install Docker

apt:

name: docker.io

state: present

- name: Ensure Docker service is running

service:

name: docker

state: started

enabled: yes

**Step 6: Scenario 3 - Self-Healing for Service Failures**

**Playbook: self\_healing\_service.yml**

---

- name: Self-healing for web service

hosts: webservers

become: yes

tasks:

- name: Gather service facts

ansible.builtin.service\_facts:

- name: Restart the web service if it is stopped

ansible.builtin.service:

name: nginx

state: started

when: ansible\_facts.services['nginx'].state != 'running'

register: service\_restart

- name: Notify if the web service was restarted

ansible.builtin.debug:

msg: "The web service was stopped and has been restarted."

when: service\_restart.changed

Here, I replaced httpd with nginx to match our web server.

**Step 7: Simulate CI/CD Pipeline Locally**

**Shell Script: scripts/run\_pipeline.sh**

#!/bin/bash

echo "Starting Local CI/CD Pipeline..."

echo "Step 1: Deploy Infrastructure..."

ansible-playbook playbooks/infrastructure.yml

echo "Step 2: Deploy Application..."

ansible-playbook playbooks/deploy\_app.yml

echo "Step 3: Run Self-Healing Check..."

ansible-playbook playbooks/self\_healing\_service.yml

echo "Pipeline Finished Successfully!"

Make the script executable:

chmod +x scripts/run\_pipeline.sh

**Step 8: Test Locally in VS Code**

1. Open terminal in VS Code.
2. Run pipeline manually:

./scripts/run\_pipeline.sh

1. Open http://localhost in a browser to see the deployed app.
2. Stop Nginx manually and rerun the pipeline to see **self-healing in action**:

sudo systemctl stop nginx

./scripts/run\_pipeline.sh

* The pipeline will restart Nginx automatically.
*  **Scenario 1**: Continuous Deployment → deploy\_app.yml
*  **Scenario 2**: IaC → infrastructure.yml
*  **Scenario 3**: Self-Healing → self\_healing\_service.yml
*  **Pipeline Simulation** → run\_pipeline.sh
*  **Optional Git Hook** → Auto-trigger pipeline on commit