

K J Somaiya School of Engineering

Department of Computer Engineering
 Course: DevOps

SMIT SANTOSH PATIL 16010122139 LYCOMPS_C1

Year: 2025-2026

DEVOPS _IA_ASSIGNMENT

TOPIC: - WOODPECKER CI/CD

Introduction

This documentation explains the complete setup and execution of a **local CI/CD pipeline using Woodpecker CI**, connected with **GitHub** and running via **Docker Compose**.

A static **HTML/CSS website** was used as the demo application.

The purpose of this implementation is to demonstrate how **automated builds, validations, and packaging** can be triggered whenever changes are pushed to a repository.

Objectives

- To set up a **self-hosted CI/CD pipeline** using **Woodpecker CI**.
- To connect **GitHub** as the version control and pipeline trigger.
- To configure **Docker-based execution agents**.
- To **automate validation, build, test, and packaging** steps of a sample application.
- To **understand and resolve practical CI/CD configuration challenges**.

Tools & Technologies

<u>Tool / Technology</u>	<u>Purpose</u>
Woodpecker CI	CI/CD Orchestrator
GitHub	Code repository and Webhook trigger
Docker Compose	Hosts Woodpecker Server & Agent
Ngrok	Exposes local server for OAuth & Webhooks
Static HTML/CSS Site	Sample Application

File Structure

```
woodpecker-static-demo/
├ .env
├ docker-compose.yml
├ .woodpecker.yaml
└ site/
  └ index.html
  └ styles.css
```

Implementation Steps

1 Install Dependencies

- Install **Docker Desktop**
- Install **Git**
- Install **Ngrok** and configure authtoken

2 Start Ngrok Tunnel

```
.\ngrok.exe http 8000
```

Copy the generated **HTTPS URL** → Used later in `.env` and GitHub OAuth.

3 Configure `.env`

```
WOODPECKER_HOST=https://<YOUR_NGROK_URL>
WOODPECKER_GITHUB_CLIENT=<OAUTH_CLIENT_ID>
WOODPECKER_GITHUB_SECRET=<OAUTH_CLIENT_SECRET>
WOODPECKER_AGENT_SECRET=change_me_demo_secret
```

4 Create GitHub OAuth App

- Go to **GitHub** → **Settings** → **Developer Settings** → **OAuth Apps**
- Create new app with:
 - **Homepage URL** = `https://<YOUR_NGROK_URL>`
 - **Callback URL** = `https://<YOUR_NGROK_URL>/authorize`
- Copy **Client ID** and **Secret**

5 Run Woodpecker Server & Agent

```
docker compose down
docker compose up -d
```

6 Login to Woodpecker CI

- Visit: `https://<YOUR_NGROK_URL>`
- Click **Login with GitHub**
- Authorize OAuth App

7. Push Sample Code to GitHub

```
git init
git add .
git commit -m "Initial commit"
git branch -M main
git remote add origin https://github.com/<username>/woodpecker-static-demo.git
git push -u origin main
```

8 Enable Repository in Woodpecker

- Go to **Woodpecker UI → Repositories → Enable**

9. CI/CD Pipeline Configuration

```
.woodpecker.yaml:

kind: pipeline
type: docker
name: default

steps:
  - name: validate
    image: alpine:3.20
    commands:
      - echo "Validating repo contents..."
      - ls -lah
      - test -f site/index.html
      - test -f site/styles.css

  - name: build
    image: node:20-alpine
    commands:
      - sh -c "echo Pretend build: copy static files to dist/"
      - sh -c "mkdir -p dist"
      - sh -c "cp -r site/* dist/"
      - sh -c "ls -la dist"

  - name: test
    image: alpine:3.20
    commands:
      - sh -c "echo Running basic grep test..."
      - sh -c "grep -q Hello dist/index.html"
      - sh -c "echo All good!"

  - name: package
    image: alpine:3.20
    commands:
      - sh -c "tar -czf dist.tar.gz dist"
      - sh -c "ls -lh dist.tar.gz"
```

Screenshots

The screenshot shows the GitHub OAuth application settings page for the "Woodpecker Local Demo" application. The application logo section includes a placeholder for a logo with a "Drag & drop" instruction and an "Upload new logo" button. The application name is set to "Woodpecker Local Demo". The homepage URL is listed as <https://apocalyptic-unputative-latish>. The application description is optional and left blank. The authorization callback URL is also listed as <https://apocalyptic-unputative-latish/authorize>. A checkbox for "Enable Device Flow" is present but unchecked. A green "Update application" button is at the bottom.

Application logo

Drag & drop

Upload new logo

You can also drag and drop a picture from your computer.

Application name *

Woodpecker Local Demo

Something users will recognize and trust.

Homepage URL *

<https://apocalyptic-unputative-latish>

The full URL to your application homepage.

Application description

Application description is optional

This is displayed to all users of your application.

Authorization callback URL *

<https://apocalyptic-unputative-latish/authorize>

Your application's callback URL. Read our [OAuth documentation](#) for more information.

Enable Device Flow

Allow this OAuth App to authorize users via the Device Flow.
Read the [Device Flow documentation](#) for more information.

Update application

The screenshot shows the GitHub application overview page for the "Woodpecker Local Demo" application. It displays ownership information, marketplace listing options, user management, and client secret details.

Optional features

Advanced

Woodpecker Local Demo

smitpatil0411 owns this application. [Transfer ownership](#)

You can list your application in the [GitHub Marketplace](#) so that other users can discover it. [List this application in the Marketplace](#)

0 users [Revoke all user tokens](#)

Client ID

Ov23liTwP02nNCpg4NEM

Client secrets [Generate a new client secret](#)

Make sure to copy your new client secret now. You won't be able to see it again.

a51c4c4c0469ab9d1d4ae5a2d9945808a80de208 [Copy](#)
Added now by **smitpatil0411**
Never used
You cannot delete the only client secret. Generate a new client secret first. [Delete](#)

Screenshot of a web browser showing the Woodpecker CI/CD application interface.

The top navigation bar includes tabs for "Woodpecker CI/CD", "Download ngrok", "New Tab", "Your Authtoken - ngr...", "Download history", "Login - Woodpecker", and a "Verify it's you" button.

The main content area displays a green logo featuring a white woodpecker head and the text "Login to Woodpecker with" followed by a "github.com" button.

The second screenshot shows the "Repos" page. The header includes a woodpecker icon, the text "3.9.0", and a user profile icon.

The page features a search bar labeled "Search..." and a list of repositories:

- smitpatil0411/CLOUD_COMPUTING_IA2 Enable
- smitpatil0411/Movie_review_analysis Enable
- smitpatil0411/TechTrends-Ecommerce Enable
- smitpatil0411/html-portfolio Enable
- smitpatil0411/woodpecker-static-demo Enable
- smitpatil0411/yumfood Enable

The screenshot shows the Woodpecker UI interface. At the top, there's a green header bar with the Woodpecker logo (a stylized bird), the word "Repos", and a user profile icon. Below the header, the repository name "smitpatil0411 / woodpecker-static-demo" is displayed. A button labeled "pipeline none" is shown, along with icons for running a pipeline and settings. A prominent button labeled "Run pipeline" with a play icon is visible. Below these, there are tabs for "Activity", "Branches", and "Pull requests", with "Activity" being the active tab. A message box states "No pipelines have been started yet." In the bottom right corner of the main area, there's a green box with the text "Repository enabled".

This screenshot shows a detailed view of a pipeline. The top navigation bar includes the Woodpecker logo, "Repos", and a user profile icon. The URL in the browser bar is "apocalyptically-unputative-latisha.ngrok-free.d...". A green banner at the top right says "Repository enabled". The main content area shows a pipeline named "Pipeline #7 MANUAL PIPELINE @ main". It indicates the pipeline was "just now" and is currently "not started yet". Below this, there are tabs for "Tasks", "Errors" (with a count of 1), "Config", and "Debug". A specific error message is highlighted: "generic pipeline definition not found".



Repos

0



← smitpatil0411 / woodpecker-static-demo

Pipeline #12 Force build after clean YAML



⌚ 1 minute ago ⏲ started

Tasks

Errors

Config

Changed files 1

Debug



smitpatil0411

main

e89a87e74d

No pipeline steps available!

Pipeline restarted

Pipeline restarted

```
PS D:\Devops\woodpecker-static-demo> git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 16 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 425 bytes | 425.00 KiB/s, done.
Total 4 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/smitpatil0411/woodpecker-static-demo.git
  b7135a6..e89a87e main -> main
PS D:\Devops\woodpecker-static-demo>
```

```

PS D:\Devops\woodpecker-static-demo> docker compose down
>> docker compose up -d
time="2025-09-28T22:08:36+05:30" level=warning msg="D:\\Devops\\woodpecker-static-demo\\docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please
✓ Container woodpecker-agent           Removed      0.5s
✓ Container woodpecker-server          Removed      0.6s
✓ Network woodpecker-static-demo_default Removed     0.6s
time="2025-09-28T22:08:38+05:30" level=warning msg="D:\\Devops\\woodpecker-static-demo\\docker-compose.yml: the attribute `version` is obsolete, please
remove it to avoid potential confusion"
[+] Running 3/3
✓ Network woodpecker-static-demo_default Created     0.1s
✓ Container woodpecker-server          Started     0.7s
✓ Container woodpecker-agent           Started     0.7s
PS D:\Devops\woodpecker-static-demo>

```

o ngrok

❖ Create instant endpoints for local containers within Docker Desktop → <https://ngrok.com/r/docker>

Session Status	online
Account	smit.p@somaiya.edu (Plan: Free)
Version	3.30.0
Region	India (in)
Latency	15ms
Web Interface	http://127.0.0.1:4040
Forwarding	https://apocalyptic-unputative-latisha.ngrok-free.dev -> http://localhost:8000
Connections	ttl opn rt1 rt5 p50 p90
	19 0 0.00 0.00 91.25 718.21

HTTP Requests

23:15:15.725 IST GET	/api/stream/events	200 OK
23:03:14.374 IST GET	/api/stream/events	200 OK

Ln 32, Col 1 Spaces: 2

```

.env | ! .woodpecker.yaml | docker-compose.yml ✘ | index.html | # styles.css |
  ↴ docker-compose.yml
  1  version: "3.9"
  2
  3  services:
  4    woodpecker-server:
  5      image: woodpeckerci/woodpecker-server:v3
  6      container_name: woodpecker-server
  7      ports:
  8        - "8000:8000"          # Web UI + OAuth callback
  9      volumes:
 10        - woodpecker-server-data:/var/lib/woodpecker
 11      environment:
 12        - WOODPECKER_OPEN=true
 13        - WOODPECKER_HOST=${WOODPECKER_HOST}
 14        - WOODPECKER_GITHUB=true
 15        - WOODPECKER_GITHUB_CLIENT=${WOODPECKER_GITHUB_CLIENT}
 16        - WOODPECKER_GITHUB_SECRET=${WOODPECKER_GITHUB_SECRET}
 17        - WOODPECKER_AGENT_SECRET=${WOODPECKER_AGENT_SECRET}
 18
 19    woodpecker-agent:
 20      image: woodpeckerci/woodpecker-agent:v3
 21      container_name: woodpecker-agent
 22      depends_on:
 23        - woodpecker-server
 24      volumes:
 25        - /var/run/docker.sock:/var/run/docker.sock
 26      environment:
 27        - WOODPECKER_SERVER=woodpecker-server:9000
 28        - WOODPECKER_AGENT_SECRET=${WOODPECKER_AGENT_SECRET}
 29
 30      volumes:
 31        - woodpecker-server-data:

```

```
! .env          ! .woodpecker.yaml X docker-compose.yml      index.html
! .woodpecker.yaml
1   kind: pipeline
2   type: docker
3   name: default
4
5   steps:
6     - name: validate
7       image: alpine:3.20
8       commands:
9         - echo "Validating repo contents..."
10        - ls -lah
11        - test -f site/index.html
12        - test -f site/styles.css
13
14     - name: build
15       image: node:20-alpine
16       commands:
17         - sh -c "echo Pretend build: copy static files to dist/"
18         - sh -c "mkdir -p dist"
19         - sh -c "cp -r site/* dist/"
20         - sh -c "ls -la dist"
21
22     - name: test
23       image: alpine:3.20
24       commands:
25         - sh -c "echo Running basic grep test..."
26         - sh -c "grep -q Hello dist/index.html"
27         - sh -c "echo All good!"
28
29     - name: package
30       image: alpine:3.20
31       commands:
32         - sh -c "tar -czf dist.tar.gz dist"
33         - sh -c "ls -lh dist.tar.gz"
34
```

Hello Woodpecker 🙌

This is a super simple static site used to demo a local Woodpecker CI/CD pipeline.

Issues and Resolutions

Issue	Cause	Resolution
cannot unmarshal 'map[...] into string	Invalid YAML format	Converted commands to proper YAML list using - and wrapped in sh -c
pipeline definition not found	Incorrect file naming or location	Ensured exact filename .woodpecker.yaml at repository root
Git push error: src refspec main does not match	Branch mismatch	Renamed master → main

Lessons Learned

Woodpecker CI is strict about YAML syntax — indentation & formatting must be perfect.
Ngrok is essential for local testing with GitHub OAuth.
Docker-based CI/CD is clean and isolated, no need to install runtimes on host.
Static site deployment can be automated easily with just a few CI steps.

Conclusion

This project demonstrates a **fully functional local CI pipeline** using **Woodpecker + GitHub + Docker + Ngrok**.

Every **push** to GitHub now automatically:

- ✓ Validates code
- ✓ Builds and packages files
- ✓ Runs automated tests