

CS 816 Software Production Engineering Mini Project

Scientific Calculator with DevOps Pipeline

Name: Swaroop A Ram Rayala

Roll No.: IMT2022587

GitHub: [Swaroop3](#)

October 11, 2025

Contents

1	Introduction	2
2	What and Why of DevOps	2
3	Toolchain Overview	2
4	Implementation Details	3
4.1	Source Control Management	3
4.2	Automated Testing	3
4.3	Build	4
4.4	Continuous Integration (Jenkins)	4
4.5	Containerization	6
4.6	Image Publishing (Docker Hub)	7
4.7	Configuration Management & Deployment (Ansible)	7
4.8	Webhook Relay via ngrok	8
5	Results and Validation	8
6	Challenges and Learnings	9
7	Future Enhancements	9
8	References	9

1 Introduction

- Developed a menu-driven scientific calculator providing square root, factorial, natural logarithm, and exponentiation.
- Delivered both CLI (`spe-calculator`) and FastAPI HTTP endpoints (e.g. `/sqrt?value=9`).
- Implemented a full DevOps toolchain: GitHub, pytest, packaging, Jenkins CI/CD, Docker, Docker Hub, Ansible deployment, and ngrok-based webhook automation.
- Repository: github.com/Swaroop3/SPE_mini

2 What and Why of DevOps

- **Collaboration:** GitHub centralizes code, issues, and history.
- **Quality:** Pytest and linting (Ruff) run on every commit to prevent regressions.
- **Flow:** Jenkins orchestrates build → test → package → containerize → push → deploy.
- **Repeatability:** Docker images and Ansible playbooks ensure consistent runtime across environments.
- **Feedback:** Webhooks (via ngrok) trigger Jenkins instantly upon push, tightening feedback loops.

3 Toolchain Overview

Stage	Tool(s)	Purpose / Notes
Source Control	Git, GitHub	Repo: <code>Swaroop3/SPE_mini</code> ; branch <code>master</code> . Conventional commits used (e.g. <code>feat:</code> , <code>ci:</code>). <code>pytest</code> covers calculator functions and API; <code>ruff check</code> enforces style.
Testing	Pytest, Ruff	<code>python -m build</code> produces <code>(dist/spe_calculator-0.1.0-py3-none-any.whl)</code> and <code>sdist</code> .
Build		Pipeline job <code>SPE-mini-pipeline</code> , triggered by GitHub webhooks via ngrok tunnel.
CI/CD	Jenkins 2.516.1	Image runs FastAPI app via Uvicorn, exposes port 8000.
Containerization	Docker, Dockerfile	Repo <code>swaroop4/spe-calculator</code> ; Jenkins pushes both build-number tag and <code>latest</code> .
Registry	Docker Hub	Playbook <code>configs/ansible/playbooks/deploy_calculator</code> pulls image and runs container.
Config Mgmt / Deployment	Ansible (<code>community.docker</code>)	Tunnel <code>https://unshy-otelia-unfermentative.ngrok.io</code> forwards GitHub events to Jenkins.
Webhook Relay	ngrok	

4 Implementation Details

4.1 Source Control Management

- Repository: github.com/Swaroop3/SPE_mini; remote configured in Jenkins with credential `spe_mini_git`.
- Branch strategy: single trunk (`master`) for this mini project; commits follow Conventional Commit format (e.g. `ci: enable ansible deployment via venv`).
- Commands:

```
git status  
git add .  
git commit -m "feat: bootstrap calculator project tooling"  
git push
```

- Evidence: screenshot of commit graph and Jenkins webhook-triggered build.

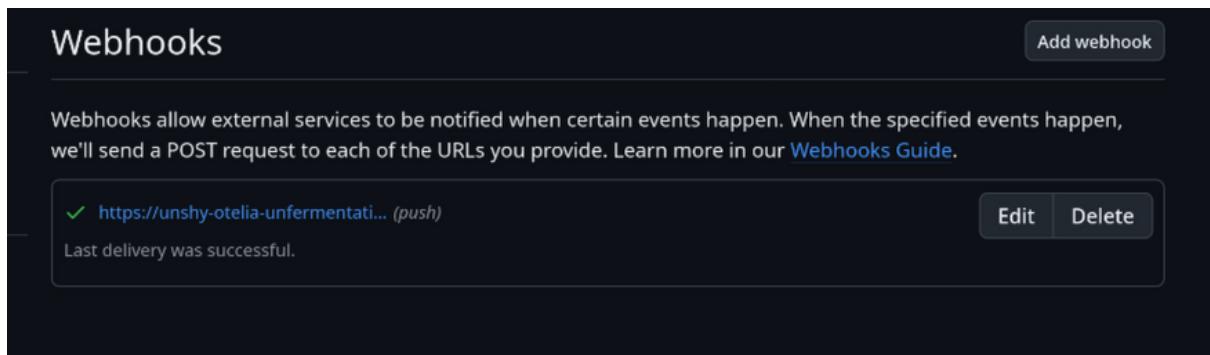


Figure 1: GitHub webhook delivery triggering Jenkins

4.2 Automated Testing

- Unit tests (`tests/test_calculator.py`) validate deterministic math results and error handling.
- API integration tests (`tests/test_api.py`) exercise FastAPI endpoints via `TestClient`.
- Jenkins stage:

```
stage('Unit Tests') {  
    sh 'mkdir -p reports'  
    sh '. ${VENV_DIR}/bin/activate && pytest --junitxml=reports/pytest.xml'  
    junit 'reports/pytest.xml'  
}
```

- Local command: `pytest --maxfail=1 --disable-warnings -q`.
- Evidence: attach Jenkins test report screenshot (14/14 passed).

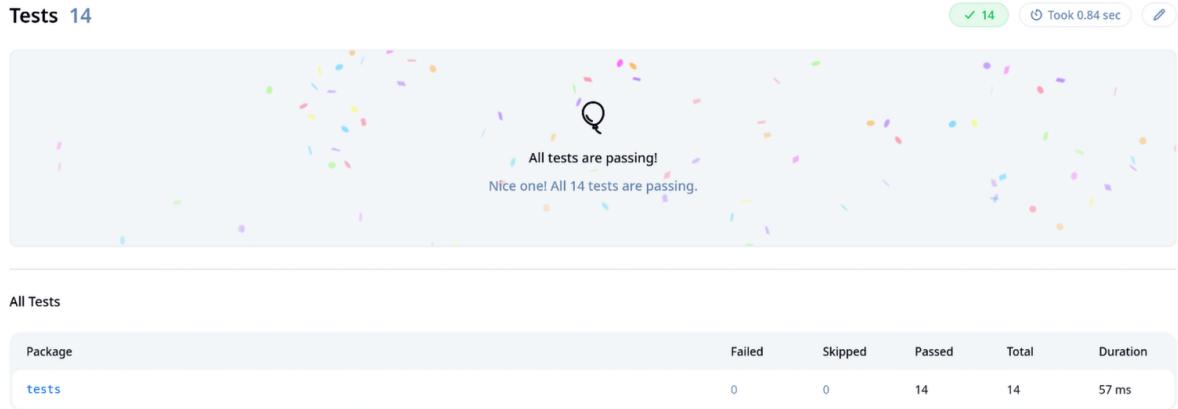


Figure 2: Jenkins unit test summary (14 tests passed)

4.3 Build

- Project metadata defined in `pyproject.toml` (setuptools backend).

- Command (Jenkins & local):

```
python -m build
```

- Artifacts archived by Jenkins (`dist/spe_calculator-0.1.0.tar.gz`, `.whl`).
- Evidence: Jenkins artifact list snapshot.



Figure 3: Jenkins pipeline run showing all stages green

4.4 Continuous Integration (Jenkins)

- Jenkinsfile (repo root) drives Pipeline:

```
pipeline {
    agent any
    environment {
        VENV_DIR = ".venv"
        DOCKER_IMAGE = "docker.io/swaroop4/spe-calculator"
        DOCKERHUB_CREDENTIALS = "dockerhub"
        DOCKER_HOST = "unix:///run/user/1000/docker.sock"
    }
    stages { ... }
}
```

- Key stages: Checkout, Setup Python, Lint, Unit Tests, Build Package, Build Image, Push Image, Deploy via Ansible.
- Webhook: GitHub → Jenkins via ngrok; Jenkins job configured with “GitHub hook trigger for GITScm polling”.
- Evidence: screenshot of pipeline stage view (all stages green).

4.5 Containerization

- Dockerfile:

```
FROM python:3.13-slim
WORKDIR /app
RUN useradd --create-home --shell /bin/bash appuser
COPY pyproject.toml README.md /app/
COPY spe_calculator /app/spe_calculator
RUN pip install --upgrade pip && pip install .
USER appuser
EXPOSE 8000
ENTRYPOINT ["uvicorn", "spe_calculator.api:app", "--host", "0.0.0.0", "--port", "8000"]
```

- Build command (Jenkins stage): docker build -t docker.io/swaroop4/spe-calculator:\$BUILD

.

- Local verification:

```
docker run --rm -p 8000:8000 docker.io/swaroop4/spe-calculator:latest
curl "http://localhost:8000/sqrt?value=9"
```

- Evidence: CLI output from curl and container logs.

The screenshot shows the Docker Hub repository page for `swaroop4/spe-calculator`. The repository was last pushed 10 minutes ago and has a size of 79.1 MB. The `General` tab is selected, showing the repository's name, description, category, and settings. Below the tabs, there is a section for `Tags`, which lists three tags: `latest`, `21`, `20`, `19`, and `18`. Each tag entry includes the tag name, OS (represented by a bell icon), type (Image), pull status (less than 1 day), and push time. A link to `See all` tags is at the bottom of the table. On the right side of the page, there is a `DOCKER SCOUT INACTIVE` button with an `Activate` link.

Tag	OS	Type	Pulled	Pushed
latest	🔔	Image	less than 1 day	10 minutes
21	🔔	Image	less than 1 day	10 minutes
20	🔔	Image	less than 1 day	15 minutes
19	🔔	Image	less than 1 day	26 minutes
18	🔔	Image	less than 1 day	about 4 hours

Figure 4: Docker Hub repository with build and latest tags

4.6 Image Publishing (Docker Hub)

- Jenkins Push stage uses stored credential dockerhub (username swaroop4).

```
withCredentials([usernamePassword(credentialsId: env.DOCKERHUB_CREDENTIALS, ...)]) {  
    sh 'docker push ${FULL_IMAGE}'  
    sh 'docker push ${DOCKER_IMAGE}:latest'  
}
```

- Registry URL: docker.io/swaroop4/spe-calculator
- Evidence: Docker Hub screenshot showing tags 20, latest.

```
(.venv) swaroop@fedora:~/Desktop/SPE$ scripts/deploy_with_ansible.sh \  
  --extra-vars "calculator_image=docker.io/swaroop4/spe-calculator:latest \  
  project_root=$PWD" \  
  build_from_source=false"  
[WARNING]: Deprecation warnings can be disabled by setting 'deprecation_warnings=False' in ansible.cfg.  
[DEPRECATION WARNING]: community.general.yaml has been deprecated. The plugin has been superseded by the option 'result_format=yaml' in callback plugin ansible.builtin.default from ansible-core 2.13 onwards. This feature will be removed from collection 'community.general' version 12.0.0.  
PLAY [Deploy scientific calculator container] *****  
TASK [Gathering Facts] *****  
[WARNING]: Host 'localhost' is using the discovered Python interpreter at '/home/swaroop/Desktop/SPE/.venv/bin/python3.13', but future installation of another Python interpreter could cause a different interpreter to be discovered. See https://docs.ansible.com/ansible-core/2.19/reference\_appendices/interpreter\_discovery.html for more information.  
ok: [localhost]  
TASK [Check Docker availability] *****  
ok: [localhost]  
TASK [Ensure project root exists] *****  
ok: [localhost]  
TASK [Abort when project root is missing] *****  
skipping: [localhost]  
TASK [Check for Dockerfile] *****  
ok: [localhost]  
TASK [Build image from local Dockerfile] *****  
skipping: [localhost]  
TASK [Pull image from registry when Dockerfile missing] *****  
ok: [localhost]  
TASK [Run scientific calculator container] *****  
changed: [localhost]  
PLAY RECAP *****  
localhost : ok=6   changed=1   unreachable=0   failed=0   skipped=2   rescued=0   ignored=0
```

Figure 5: Ansible deployment play recap

4.7 Configuration Management & Deployment (Ansible)

- Playbook: configs/ansible/playbooks/deploy_calculator.yml
- Jenkins runs:

```
scripts/deploy_with_ansible.sh --extra-vars \  
  '${extraVars}'
```

- Helper script ensures DOCKER_HOST for rootless docker, installs community.docker.
- Critical tasks: ensure repo path exists, pull image (`build_from_source=false`), run container mapping host port 8000.
- Manual command for reproducibility:

```
scripts/deploy_with_ansible.sh \  
  --extra-vars "calculator_image=docker.io/swaroop4/spe-calculator:latest \  
  project_root=/home/swaroop/Desktop/SPE \  
  build_from_source=false"
```

- Evidence: Ansible output (PLAY RECAP with `changed=1`) and `docker ps` result.

4.8 Webhook Relay via ngrok

- Config file: `configs/ngrok/ngrok.yml` with Jenkins tunnel on 8080.
- Start command: `scripts/start_ngrok.sh`.
- GitHub Webhook configuration:
 - Payload URL: `https://unshy-otelia-unfermentative.ngrok-free.dev/github-webhook/`
 - Content type: `application/json`
 - Event: Push
- Evidence: GitHub webhook delivery history and Jenkins build triggered by push.

5 Results and Validation

- Jenkins build #20 success: lint, tests (14 pass), package, Docker push, Ansible deploy.

- Deployed container responds:

```
$ curl "http://localhost:8000/sqrt?value=25"
{"operation":"sqrt","result":5.0}
$ curl "http://localhost:8000/power?base=2&exponent=8"
{"operation":"power","result":256.0}
```

- CLI transcript:

```
$ spe-calculator
Scientific Calculator
1. Square Root (x)
...
Select an option (1-5): 2
Enter a non-negative integer: 5
5! = 120
```

- Docker Hub tag list confirms `swaroop4/spe-calculator:20` and `:latest`.
- Ansible play recap:

```
localhost : ok=6  changed=1  failed=0
```

```
swaroop@fedora:~/Desktop/SPE$ docker ps
CONTAINER ID   IMAGE          COMMAND       CREATED      STATUS      PORTS     NAMES
59ebb60bea64   swaroop4/spe-calculator:latest   "uvicorn spe_calcula..."   6 minutes ago   Up 6 minutes   0.0.0.0:8000->8000/tcp   scientific-calculator
```

Figure 6: Running container and API verification

```
swaroop@fedora:~/Desktop/SPE$ curl "http://localhost:8000/power?base=2&exponent=5"
{"operation": "power", "result": 3}
url "http://localhost:8000/sqrt?value=9"
{"operation": "sqrt", "result": 3.0}
swaroop@fedora:~/Desktop/SPE$
```

Figure 7: CLI demonstration of calculator operations

6 Challenges and Learnings

- **Rootless Docker Access:** Jenkins needed `DOCKER_HOST=unix:///run/user/1000/docker.sock` and local `DOCKER_CONFIG` to authenticate cleanly.
- **Ansible Defaults:** Original variable defaults referenced themselves, causing recursive loops; resolved with derived variables and explicit `build_from_source` flag.
- **Credential Hygiene:** Docker Hub login inside Jenkins required isolated config instead of default credential helper.
- **Webhook Exposure:** ngrok provided easy tunneling but demands updates when the URL changes; longer term solution could be a static domain or hosted Jenkins.

7 Future Enhancements

- Add integration tests covering API error paths (e.g., invalid JSON).
- Publish code coverage reports to Jenkins/Allure.
- Add health checks and monitoring dashboards (Prometheus + Grafana).
- Parameterize Ansible deploy to target remote hosts via SSH inventory.
- Secure Jenkins ngrok tunnel with Webhook secret validation.

8 References

- FastAPI docs: <https://fastapi.tiangolo.com/>
- Pytest docs: <https://docs.pytest.org/>
- Jenkins Declarative Pipeline: <https://www.jenkins.io/doc/book/pipeline/syntax/>
- Docker documentation: <https://docs.docker.com/>
- Ansible community.docker collection: <https://docs.ansible.com/ansible/latest/collections/community/docker/>
- ngrok docs: <https://ngrok.com/docs>

Appendix

- GitHub repository: https://github.com/Swaroop3/SPE_mini
- Jenkins job (local): <http://localhost:8080/job/SPE-mini-pipeline>
- Docker Hub: <https://hub.docker.com/r/swaroop4/spe-calculator>
- Jenkinsfile: `Jenkinsfile` at repository root.
- Ansible playbook: `configs/ansible/playbooks/deploy_calculator.yml`
- Report template source: `docs/report-template.md`