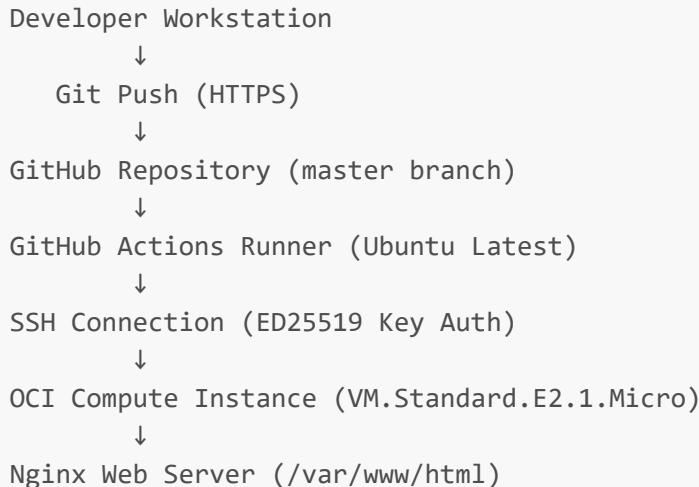


## Overview

This project implements continuous deployment for a static website hosted on Oracle Cloud Infrastructure. The pipeline uses GitHub Actions to automatically deploy code changes via SSH, eliminating manual intervention while maintaining security controls.

---

## Architecture



---

## Infrastructure

**Cloud Platform:** Oracle Cloud Infrastructure (OCI)

**Compute:** VM.Standard.E2.1.Micro (Always Free tier)

**Operating System:** Ubuntu 24.04 LTS

**Web Server:** Nginx

**Region:** Netherlands Northwest (Amsterdam)

### Network Configuration:

- Public IP: Assigned
  - Security List: Custom ingress rules
  - VCN: Default OCI configuration
- 

## Security Controls

### SSH Hardening

- **Password Authentication:** Disabled
- **Root Login:** Prohibited
- **Public Key Authentication:** Required (ED25519)

### Firewall Configuration (UFW)

- Port 22 (SSH): Allowed from anywhere
- Port 80 (HTTP): Allowed from anywhere
- Port 80 (HTTP IPv6): Allowed from anywhere
- Default Policy: Deny incoming, allow outgoing

## Access Control

- GitHub Actions authentication via encrypted secrets
- SSH private key stored in GitHub repository secrets
- Server credentials never exposed in workflow files

---

## CI/CD Pipeline

### Workflow Trigger

```
on:  
  push:  
    branches: [ master ]
```

### Deployment Process

1. GitHub detects push to master branch
2. Workflow spins up Ubuntu runner
3. SSH connection established using private key from secrets
4. Commands execute on remote server:
  - Navigate to web directory
  - Pull latest code from GitHub
  - Reload Nginx (zero downtime)
5. Changes live within 10-15 seconds

### Secrets Configuration

Three repository secrets required:

- `SSH_PRIVATE_KEY`: ED25519 private key for authentication
- `SERVER_IP`: OCI instance public IP
- `SERVER_USER`: SSH username

---

## Deployment Evidence

### Successful Pipeline Execution

The screenshot shows the GitHub Actions interface for a repository named 'Deployment-Test'. The left sidebar is collapsed, and the main area displays 'All workflows' under the 'Actions' tab. A search bar at the top right says 'Type ⌘ to search'. Below it is a filter bar for 'Workflow runs' with dropdowns for Event, Status, Branch, and Actor, and a 'Filter workflow runs' input field.

**9 workflow runs**

Event	Status	Branch	Actor	Time	Duration	More
✓ confirmed github actions and fixed html	Success	master	escanut	Today at 6:59 PM	9s	...
✓ confirmed github actions	Success	master	escanut	Today at 6:57 PM	10s	...
✓ tested modified html	Success	master	escanut	Today at 6:55 PM	12s	...
✓ Updated yaml file	Success	master	escanut	Today at 6:52 PM	10s	...

Multiple successful deployments showing consistent 9-12 second execution times

## Repository Secrets

The screenshot shows the 'Repository secrets' page for the same repository. The left sidebar is expanded, showing sections like Actions, Models, Webhooks, Copilot, Environments, Codespaces, Pages, Security, Advanced Security, Deploy keys, and Secrets and variables. The 'Actions' section is currently selected. At the top right is a 'Preview' button. The main area shows a message 'This environment has no secrets.' and a 'Manage environment secrets' button. Below this is a table titled 'Repository secrets' with a 'New repository secret' button.

Name	Last updated	Actions
SERVER_IP	4 hours ago	edit delete
SERVER_USERNAME	4 hours ago	edit delete
SSH_PRIVATE_KEY	3 hours ago	edit delete

At the bottom, there's a footer with links: © 2026 GitHub, Inc. Terms Privacy Security Status Community Docs Contact Manage cookies Do not share my personal information.

Encrypted credentials stored securely in GitHub

## Infrastructure Status

The screenshot shows the Oracle Cloud Instances page. At the top, there's a search bar and a dropdown for the region: 'Netherlands Northwest (Amsterdam)'. Below the header, the title 'Instances' is displayed. A note states: 'An **instance** is a compute host. Choose between virtual machines (VMs) and bare metal instances. The image that you use to launch an instance determines its operating system and other software.' A search bar and a 'Search' button are present. Under 'Applied filters', it says 'Compartment One1010 (root)'. There are buttons for 'Create instance' and 'Actions'. A table lists the instance details:

Name	State	Public IP	Private IP	Shape	OCPUs	Memory (GB)	Availability domain
Always Free	Running	141.144.206.53		VM.Standard.E2.1.Micro	1	1	AD-1

Always Free tier compute instance in running state

## Security Configuration

The screenshot shows a Parrot OS desktop environment. In the foreground, a terminal window displays the output of the command 'sudo ufw status numbered':

```
sudo ufw status numbered
Status: active
[ 1] OpenSSH (OpenSSH v8.2p1-4+deb10u1) ALLOW IN  Anywhere
[ 2] 22/tcp (OpenSSH v8.2p1-4+deb10u1) ALLOW IN  Anywhere
[ 3] 80/tcp (Apache2 - www-data) ALLOW IN  Anywhere
[ 4] OpenSSH (v6) (OpenSSH v8.2p1-4+deb10u1) ALLOW IN  Anywhere (v6)
[ 5] 22/tcp (v6) (OpenSSH v8.2p1-4+deb10u1) ALLOW IN  Anywhere (v6)
[ 6] 80/tcp (v6) (Apache2 - www-data) ALLOW IN  Anywhere (v6)
```

In the background, another terminal window shows the command 'lsblk' and a file browser window titled 'index.html'.

Active firewall with restrictive ingress rules

A screenshot of a Parrot OS desktop environment. In the top panel, there are icons for Applications, Places, System, and a terminal window titled 'Parrot Terminal'. The date and time 'Sat Jan 3, 21:28' are also displayed. Below the top panel, there is a menu bar with File, Edit, View, Search, Terminal, Tabs, and Help. A terminal window is open with the command `sudo grep "PasswordAuthentication\|PermitRootLogin\|PubkeyAuthentication" /etc/ssh/sshd_config`, showing the configuration for SSH authentication. In the background, a browser window displays a local file named 'index.html' containing some basic HTML code.

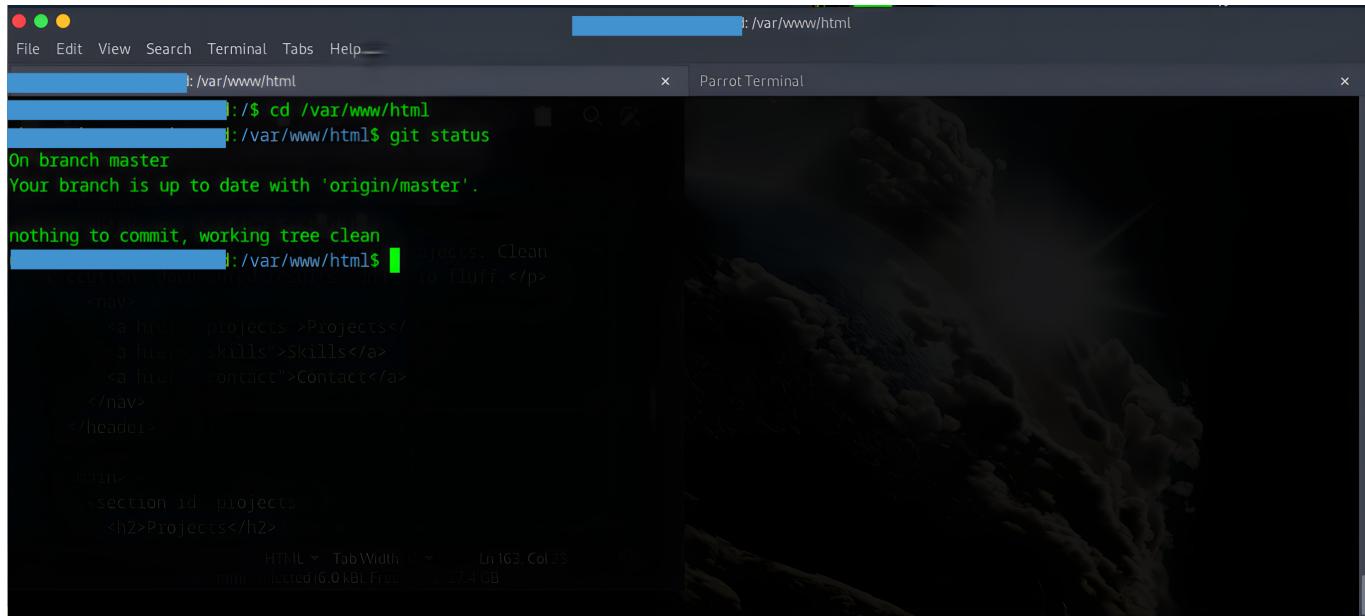
*Hardened SSH configuration with password authentication disabled*

## Live Deployment

A screenshot of a web browser window showing a live deployed website. The address bar indicates the site is not secure and shows the IP address 141.144.206.53. The main heading is 'Victor | CI/CD Test'. Below it, a subtitle reads 'Junior cybersecurity and IT projects. Clean execution, documented results, and zero fluff.' There are three project cards: 'Cloud VM Hardening', 'CI/CD Static Site', and 'Basic Network Recon'. Each card has a brief description and a list of tags. Below the projects is a 'Skills' section with tabs for Core, Tools, and Workflow.

*Deployed website accessible via public IP*

## Git Repository Status



A screenshot of a terminal window titled "Parrot Terminal". The window has two tabs: one for "File" and another for "Terminal". The terminal tab shows the command line and its output. The command `git status` is run, and the output indicates that the branch is up-to-date with 'origin/master' and there is nothing to commit. The working tree is clean. Below the terminal, the content of a file named "index.html" is displayed in a code editor. The file contains HTML code for a navigation bar and a section for projects.

Clean working tree synced with remote repository

---

## Technical Implementation

### GitHub Actions Workflow

Location: [.github/workflows/deploy.yml](#)

The workflow uses GitHub's hosted runners to establish SSH connections and execute remote commands. All commands run within a single SSH session to maintain context and reduce connection overhead.

### Server Configuration

- Web root: [/var/www/html](#)
  - Git repository: Cloned directly into web root
  - Nginx: Reloads configuration on each deployment
- 

## Performance Metrics

Based on production deployment data:

Metric	Value
Average Deployment Time	10-12 seconds
Successful Deployments	9/9 shown
SSH Connection Time	< 2 seconds
Git Pull Time	< 3 seconds
Nginx Reload Time	< 1 second

This deployment uses one compute instance with standard storage allocation.

---

# Operational Considerations

## Monitoring

- GitHub Actions provides execution logs for all deployments
- Nginx access and error logs available on server
- UFW logs track connection attempts

## Backup Strategy

Git serves as the primary backup mechanism. Full repository history maintained on GitHub with the ability to roll back to any previous commit.

---

# Reproduction Steps

## Prerequisites

- Oracle Cloud account (Always Free tier)
- GitHub account
- SSH key pair (ED25519 recommended)
- Basic Linux command line knowledge

## Setup Process

1. Provision OCI compute instance (Ubuntu 24.04)
  2. Configure security list for HTTP/HTTPS traffic
  3. Harden SSH configuration
  4. Enable UFW firewall
  5. Install Nginx and Git
  6. Clone repository to `/var/www/html`
  7. Generate deployment SSH key pair
  8. Add public key to server's `authorized_keys`
  9. Store private key and credentials in GitHub secrets
  10. Create GitHub Actions workflow file
  11. Push to master branch to test deployment
- 

# Technologies Used

- **Cloud:** Oracle Cloud Infrastructure (OCI)
  - **CI/CD:** GitHub Actions
  - **Web Server:** Nginx
  - **Operating System:** Ubuntu 24.04 LTS
  - **Security:** UFW, SSH key authentication
  - **Version Control:** Git, GitHub
- 

# Project Context

This deployment demonstrates practical DevOps implementation suitable for client environments requiring automated deployments without enterprise tooling complexity. The architecture scales to support multiple sites on a single instance or can be replicated across multiple instances for different applications.

The security posture aligns with standard cloud infrastructure requirements including SSH hardening, firewall configuration, and credential management through encrypted secrets.

---