Technical Case Study: Automated GitOps Pipeline

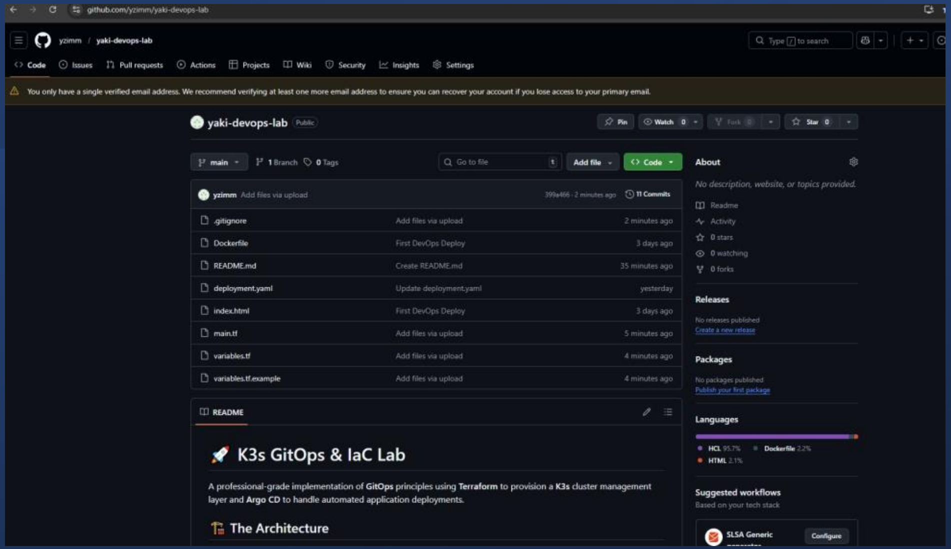
Engineer: Yaki Zimmerman

Date: February 2026

Stack: K3s, Terraform, Argo CD, GitHub, Ubuntu

Executive Summary

This project demonstrates a production-grade **GitOps implementation** designed to eliminate manual intervention in Kubernetes deployments. By utilizing **Infrastructure-as-Code (IaC)** via Terraform, I provisioned a self-healing environment where the cluster state is strictly governed by a GitHub "Source of Truth".



You only have a single verified email address. We recommend verifying at least one more email address to ensure you can recover your account if you lose access to your primary email.

yaki-devops-lab

No description, website, or topics provided.

No releases published
Create a new release

No packages published
Publish your first package

yzimm Add files via upload 399a466 · 2 minutes ago 11 Commits

.gitignore	Add files via upload	2 minutes ago
Dockerfile	First DevOps Deploy	3 days ago
README.md	Create README.md	35 minutes ago
deployment.yaml	Update deployment.yaml	yesterday
index.html	First DevOps Deploy	3 days ago
main.tf	Add files via upload	5 minutes ago
variables.tf	Add files via upload	4 minutes ago
variables.tf.example	Add files via upload	4 minutes ago

🚀 K3s GitOps & IaC Lab

A professional-grade implementation of **GitOps** principles using **Terraform** to provision a **K3s** cluster management layer and **Argo CD** to handle automated application deployments.

🏗 The Architecture

Architecture Overview

Infrastructure Layer:
Ubuntu VM running a lightweight K3s cluster.
Orchestration & Security: Terraform automates the deployment of Argo CD via Helm.
It manages sensitive data using bcrypt password hashing and Kubernetes secrets for GitHub access.
The GitOps Loop: Argo CD monitors the repository and performs automatic reconciliation
to ensure the live environment matches the code.

Security & Compliance

Secret Masking: Implemented .gitignore to prevent leakage of terraform.tfvars and cluster configs.

Collaboration: Provided a variables.tf.example to ensure project portability and secure team onboarding.

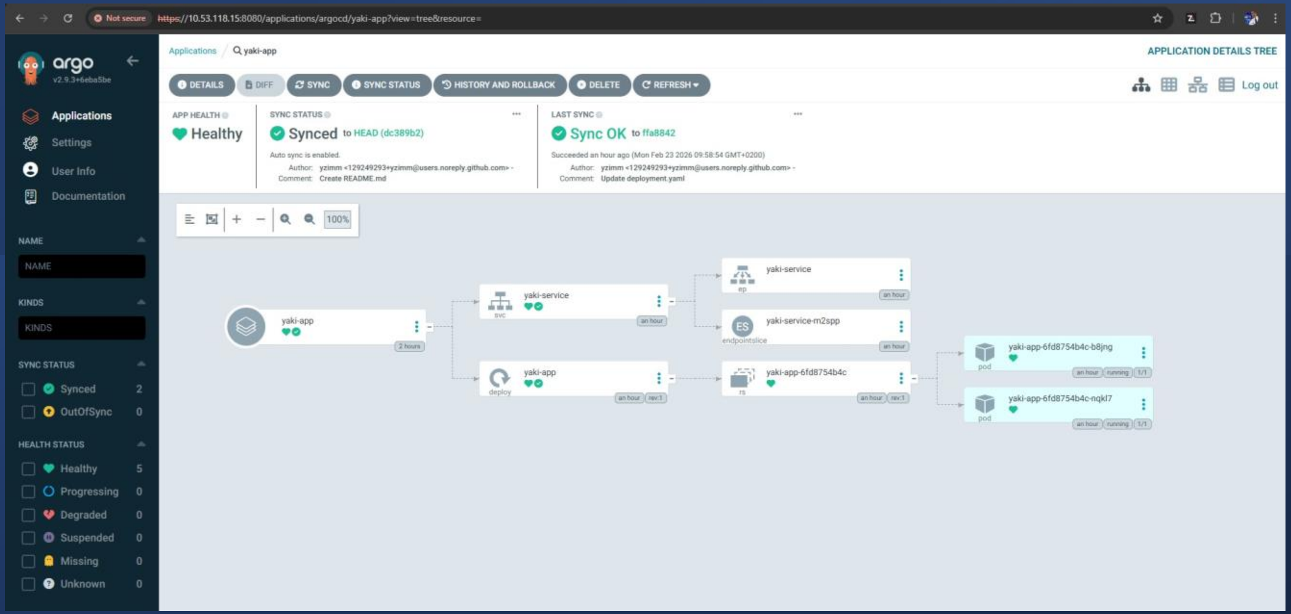
Problem Solving & Troubleshooting

Resolution of Path Conflicts: Successfully debugged a ComparisonError by realigning Terraform application paths with repository structure, moving from "Unknown" to "Healthy" status.

Networking: Configured secure port-forwarding on 0.0.0.0 to bridge VM internal services with external management UIs.

Evidence of Success

Sync Status: The application is verified as "Healthy" and "Synced".
Object Mapping: Automated creation and management of Service, ReplicaSet, and Pod objects.
Zero-Touch Deployment: A single commit to GitHub triggers a full environment update without manual CLI commands.



Not secure https://10.53.118.15:8080/applications/argocd/yaki-app?view=tree&resource=

APP HEALTH
Healthy

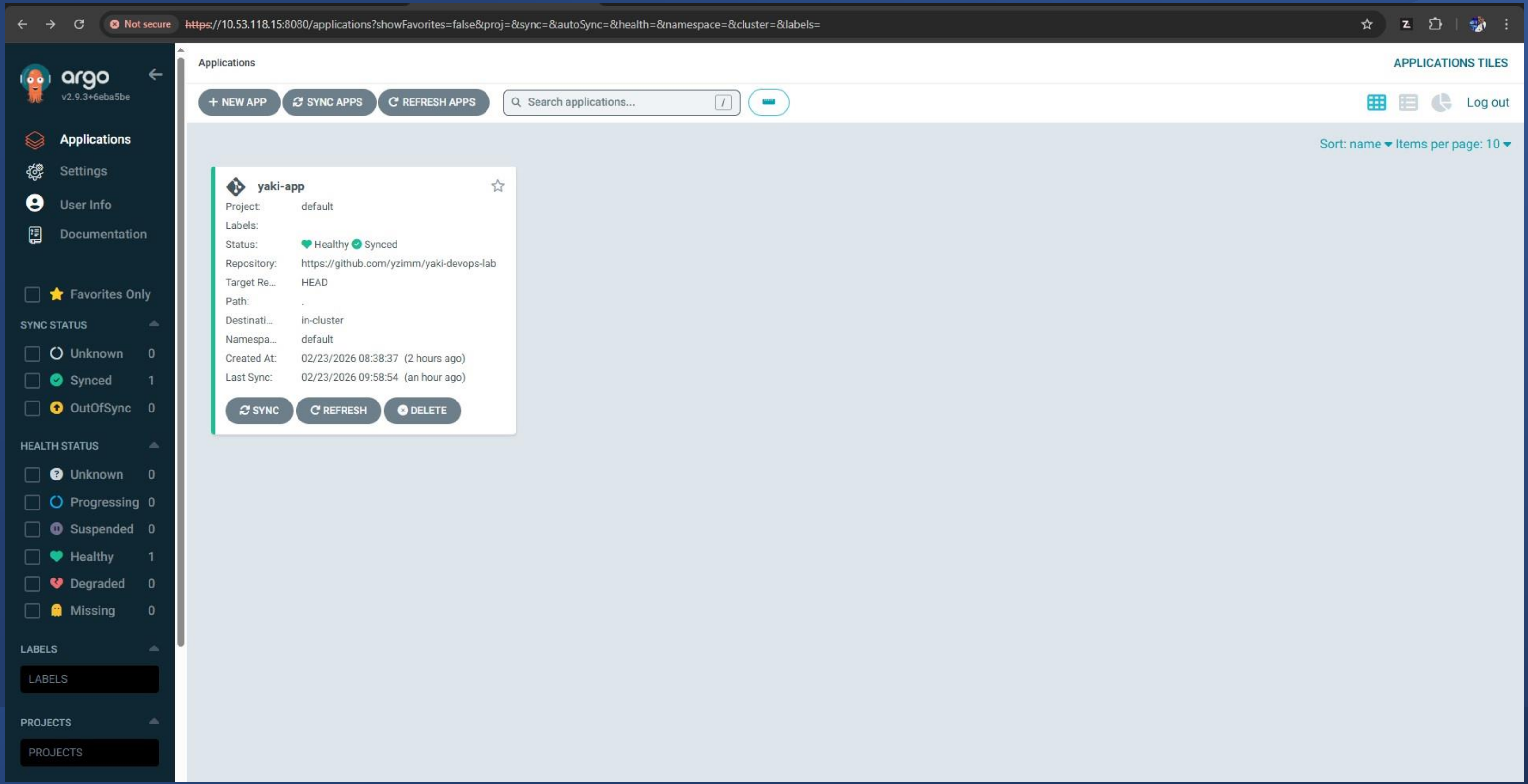
SYNC STATUS
Synced to HEAD (dc389b2)
Auto sync is enabled.
Author: yzimm <129249293+yzimm@users.noreply.github.com> -
Comment: Create README.md

LAST SYNC
Sync OK to ffa8842
Succeeded an hour ago (Mon Feb 23 2026 09:58:54 GMT+0200)
Author: yzimm <129249293+yzimm@users.noreply.github.com> -
Comment: Update deployment.yaml

SYNC STATUS
□ Synced 2
□ OutOfSync 0

HEALTH STATUS
□ Healthy 5
□ Progressing 0
□ Degraded 0
□ Suspended 0
□ Missing 0
□ Unknown 0

Visual Proof



~~https~~://10.53.118.15:8080/applications?showFavorites=false&proj=&sync=&autoSync=&health=&namespace=&cluster=&labels=

Applications

yaki-app

Project:	default
Labels:	
Status:	Healthy Synced
Repository:	https://github.com/yzimm/yaki-devops-lab
Target Re...	HEAD
Path:	.
Destinati...	in-cluster
Namespa...	default
Created At:	02/23/2026 08:38:37 (2 hours ago)
Last Sync:	02/23/2026 09:58:54 (an hour ago)

SYNC STATUS

□ Unknown	0
□ Synced	1
□ OutOfSync	0

HEALTH STATUS

□ Unknown	0
□ Progressing	0
□ Suspended	0
□ Healthy	1
□ Degraded	0
□ Missing	0

LABELS

PROJECTS

Future Roadmap

Remote State: Migrate Terraform state to an S3 backend for team-based scaling.
Ingress Routing: Replace port-forwarding with a professional Nginx Ingress Controller.
Advanced CI/CD: Integrate GitHub Actions for automated terraform plan validation.

Thank You

- END-