Forgejo Deployment Troubleshooting & Resolution Report

Generated on: April 06, 2025 at 06:29 PM

# Overview

This document provides a detailed account of the challenges encountered while deploying Forgejo using a Helm chart in a Minikube Kubernetes cluster, along with the resolutions applied.

## 1. Docker CLI Path Issues

Symptoms:  
- `docker` command not found in Git Bash/MINGW32  
- `exec: "com.docker.cli.exe": executable file not found in %PATH%`  
  
Root Cause:  
- Docker Desktop paths were not added to the Windows PATH environment variable  
- Git Bash was not configured to recognize Docker commands  
  
Resolution:  
- Added Docker's CLI path manually to the shell configuration:  
 `export PATH="$PATH:/c/Program Files/Docker/Docker/resources/bin"`

## 2. Helm Installation Errors

Key Errors:  
- `INSTALLATION FAILED: "forgejo" has no deployed releases`  
- `UPGRADE FAILED: "forgejo" has no deployed releases`  
  
Causes:  
- Previous failed installations left no valid release metadata  
  
Resolution:  
- Used `helm uninstall forgejo -n code-hosting --no-hooks` to clear failed release metadata  
- Re-ran Helm install with correct values

## 3. Image Pull Failures

Critical Errors:  
- Failed to pull image `ghcr.io/forgejo/forgejo:1.20.3`: denied  
- `manifest unknown` error from container runtime  
  
Root Causes:  
- Invalid or private image tag (`1.20.3` was not publicly available)  
- Confusion between Codeberg and GHCR as image sources  
  
Resolution:  
- Updated `values.yaml` to use a valid public image:  
 `ghcr.io/forgejo/forgejo:7.0.2-1-rootless`

## 4. Namespace Management Conflicts

Errors:  
- `Error: namespaces "code-hosting" not found`  
- Invalid ownership metadata for existing namespace  
  
Triggers:  
- Manual creation of namespace without Helm annotations  
- Helm chart also attempted to create the same namespace  
  
Resolution:  
- Made namespace creation conditional in the chart  
- Added `createNamespace: false` and `namespace: code-hosting` in `values.yaml`  
- Manually created the namespace using `kubectl create namespace code-hosting`

## 5. PersistentVolumeClaim (PVC) Issues

Symptoms:  
- Pods stuck in Pending due to PVC binding issues  
  
Root Causes:  
- Mismatch in StorageClass  
- Helm chart referenced PVC before the namespace existed  
  
Resolution:  
- Ensured namespace existed before applying Helm  
- Allowed default StorageClass to bind PVC correctly

## 6. Helm Chart Configuration Issues

Problems:  
- Missing `values.yaml` file on upgrade  
- Incorrect image pull policy (should be `Always` during testing)  
- Misconfigured port settings  
  
Resolution:  
- Corrected image tag and pull policy  
- Verified file paths when running Helm commands

## 7. Minikube Networking Problems

Indicators:  
- curl and port-forward commands failed  
- Image pulls failed despite Docker Desktop being installed  
  
Resolution:  
- Verified Minikube’s container runtime and networking  
- Ensured `docker-env` integration when working with images locally

## 8. Final Verification Challenges

Post-Deployment Issues:  
- Port-forwarding failed due to pods not being in Running state  
- Resources appeared missing even though Helm reported success  
  
Resolution:  
- Verified pod status using `kubectl describe pod`  
- Manually deleted broken pods to trigger new pulls  
- Used correct and public image tag to bring up running pods

## 9. Helm Diagnostic Tools Used

- `helm list -n code-hosting`: To verify the release status  
- `helm get manifest forgejo -n code-hosting`: To debug generated manifests  
- `kubectl describe pod`: To trace deployment lifecycle and events

# Resolution Summary

- Added Docker CLI path manually to environment variables  
- Cleaned Helm release history using `helm uninstall`  
- Switched to a working public image tag: `7.0.2-1-rootless`  
- Created namespace manually and disabled namespace creation in the chart  
- Verified pod logs, events, and Helm manifests for troubleshooting

# Key Lessons Learned

- Always verify image tags from official registry sources  
- Use `--create-namespace` wisely with Helm  
- Avoid creating the same namespace manually and via Helm chart  
- Use `helm template` or `helm get manifest` for dry-run validation  
- Ensure local tools like Docker and Minikube are correctly integrated