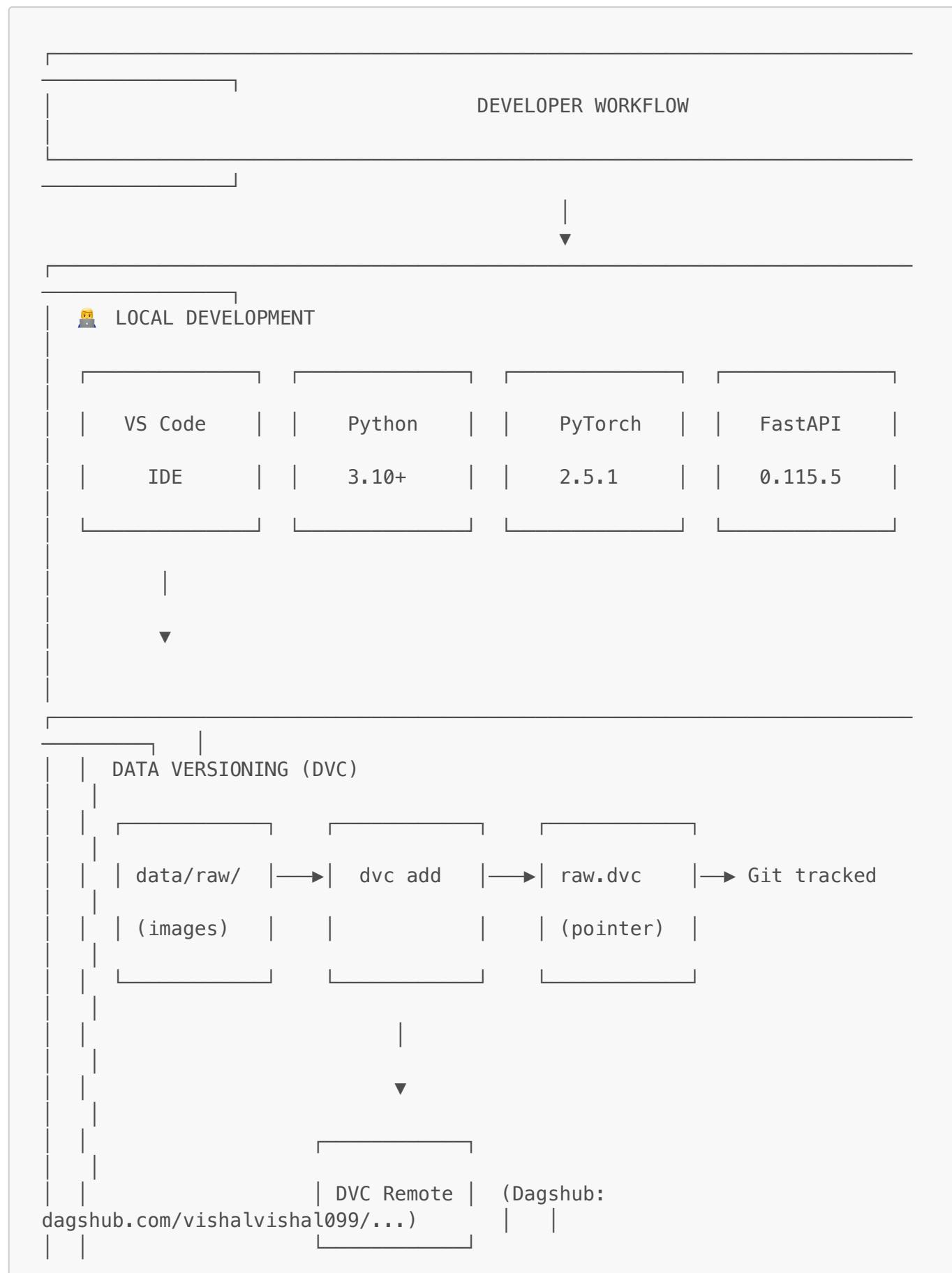
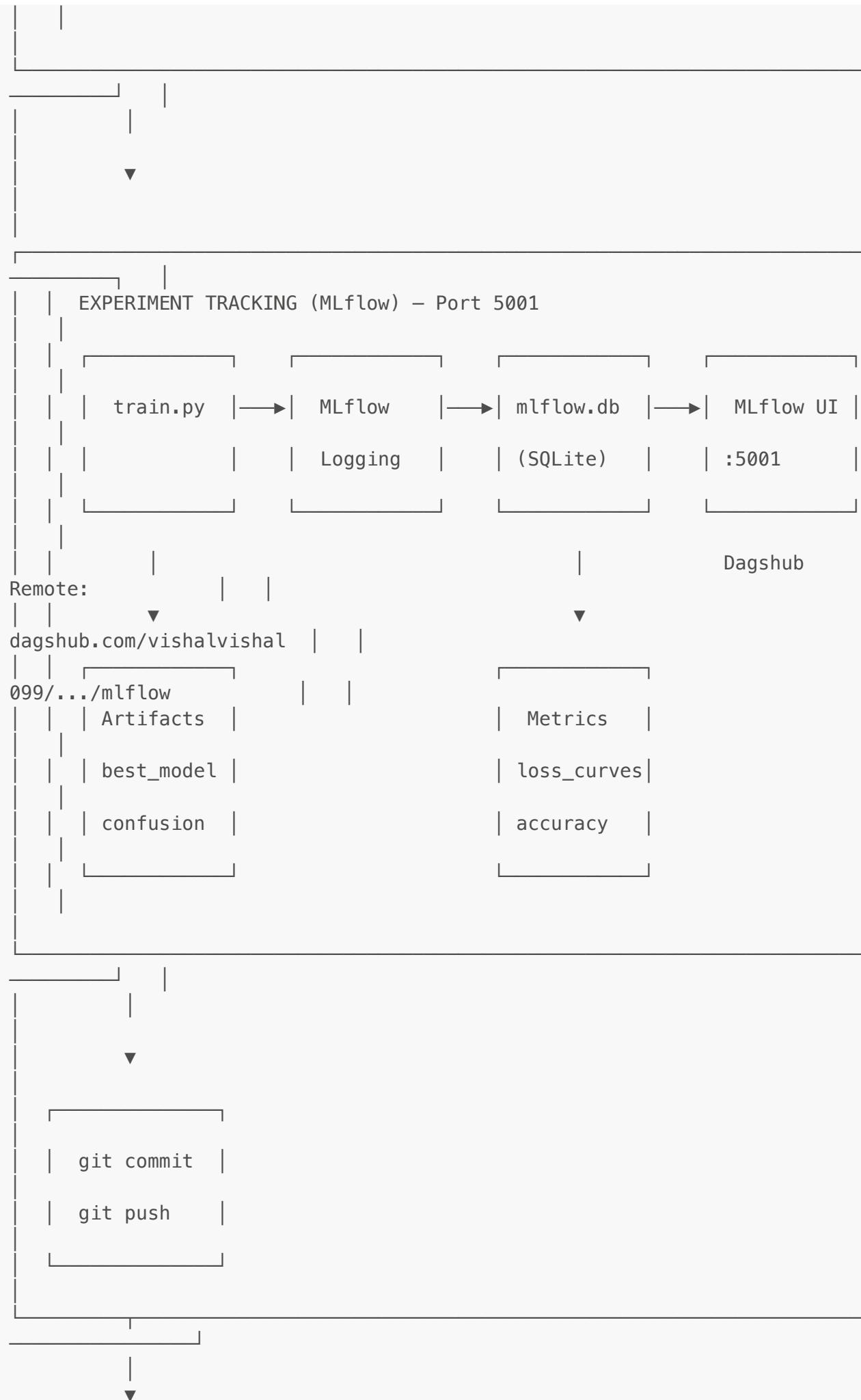
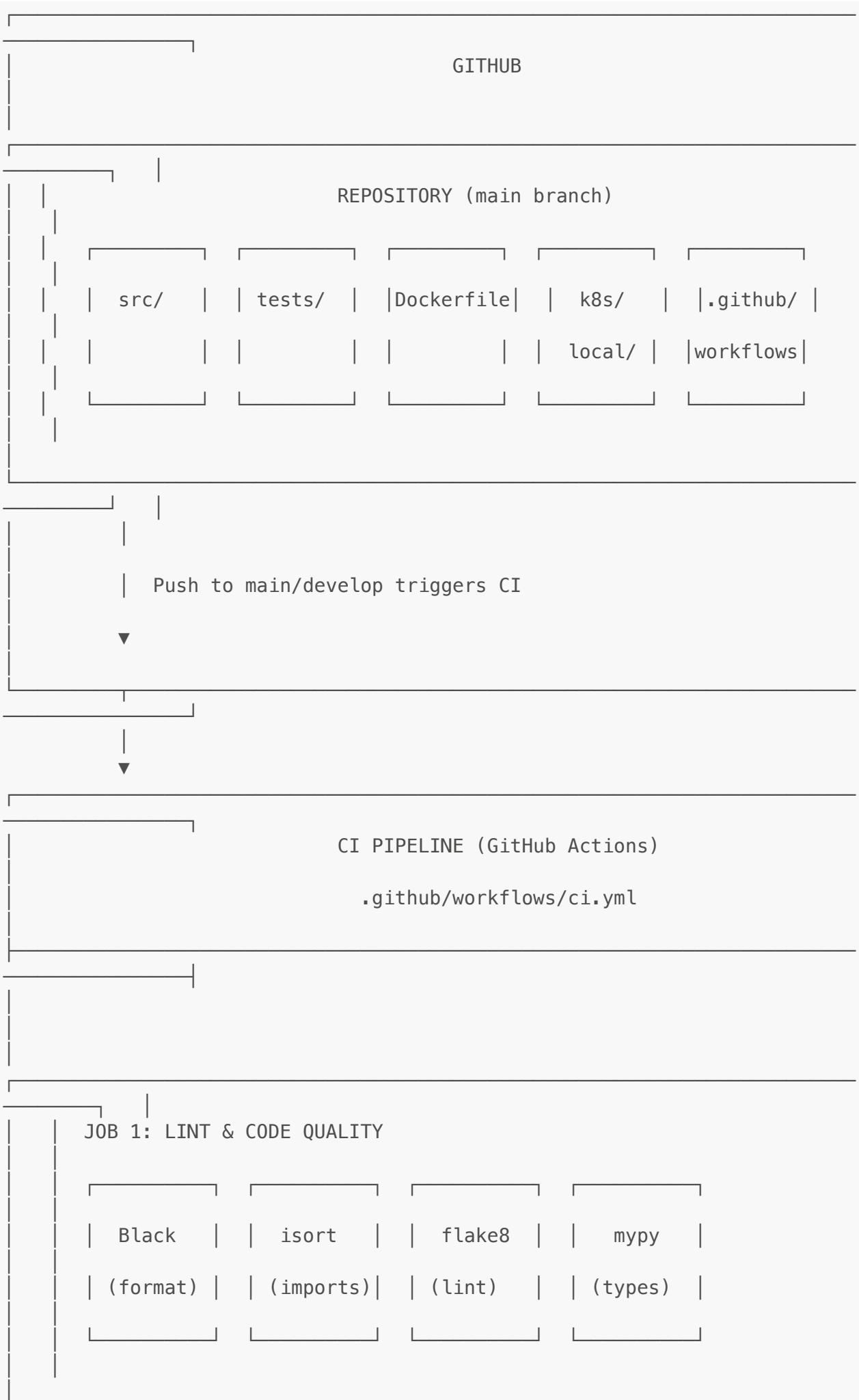


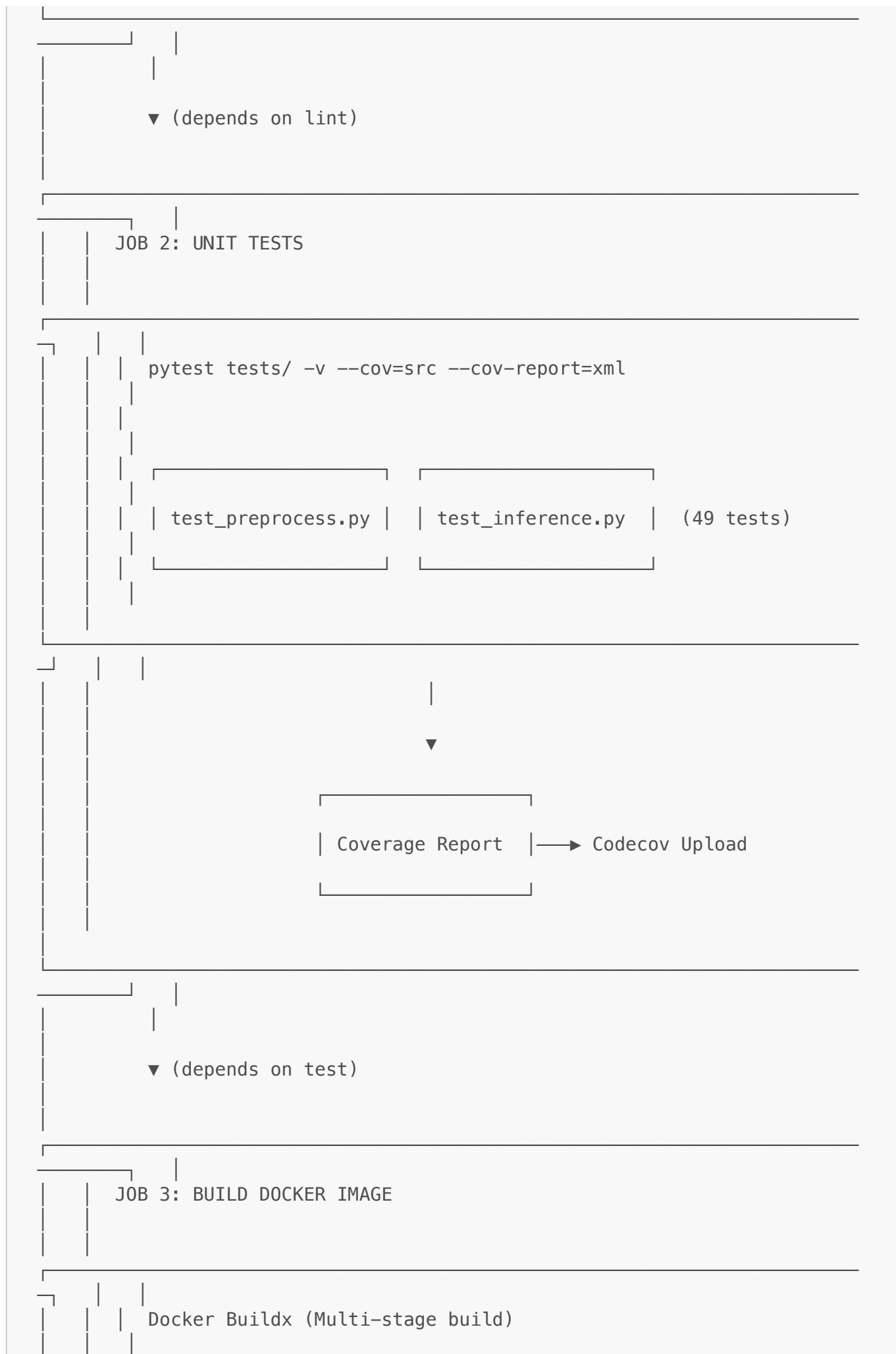
# CI/CD Architecture Diagram

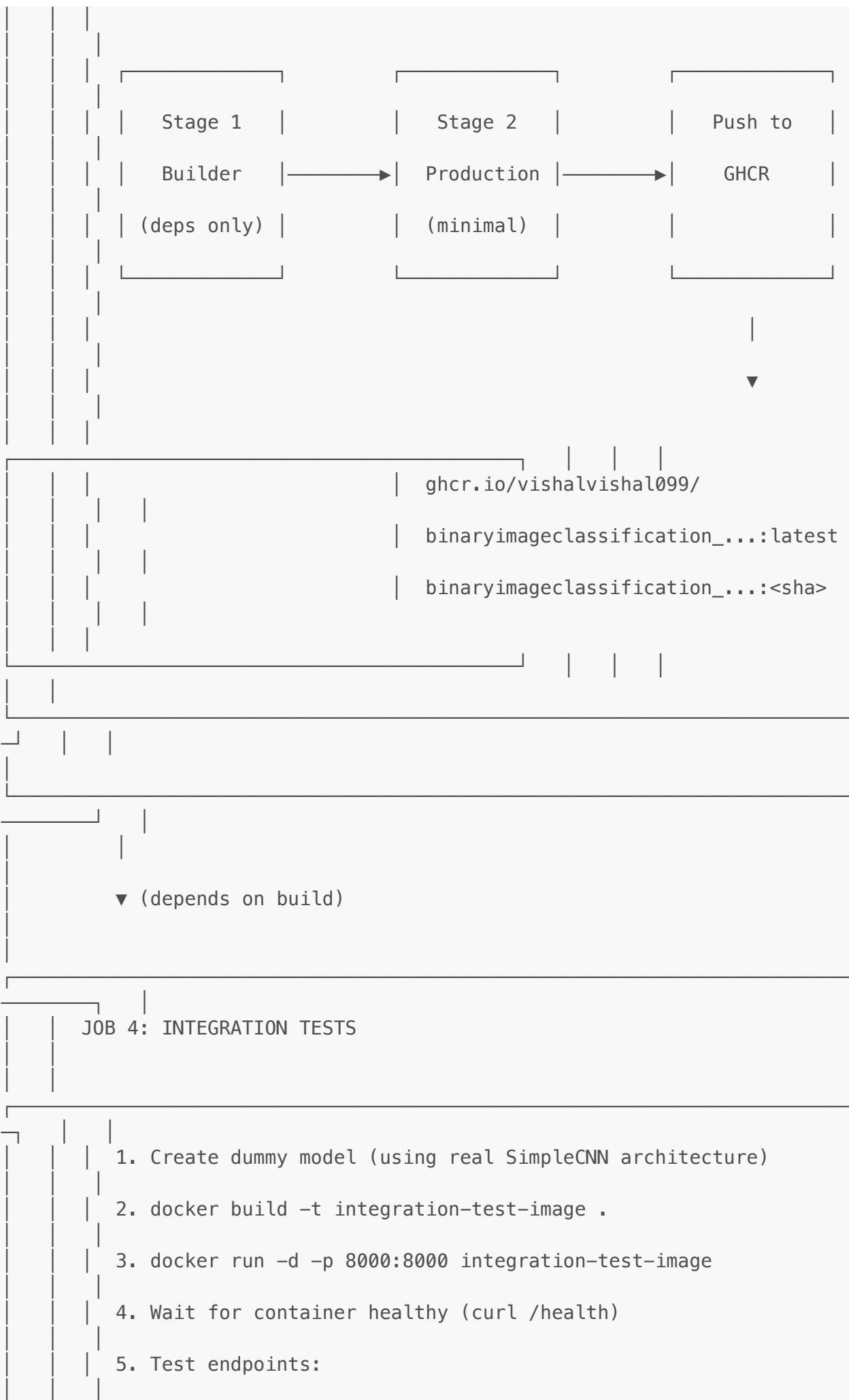
## Complete MLOps Pipeline Architecture

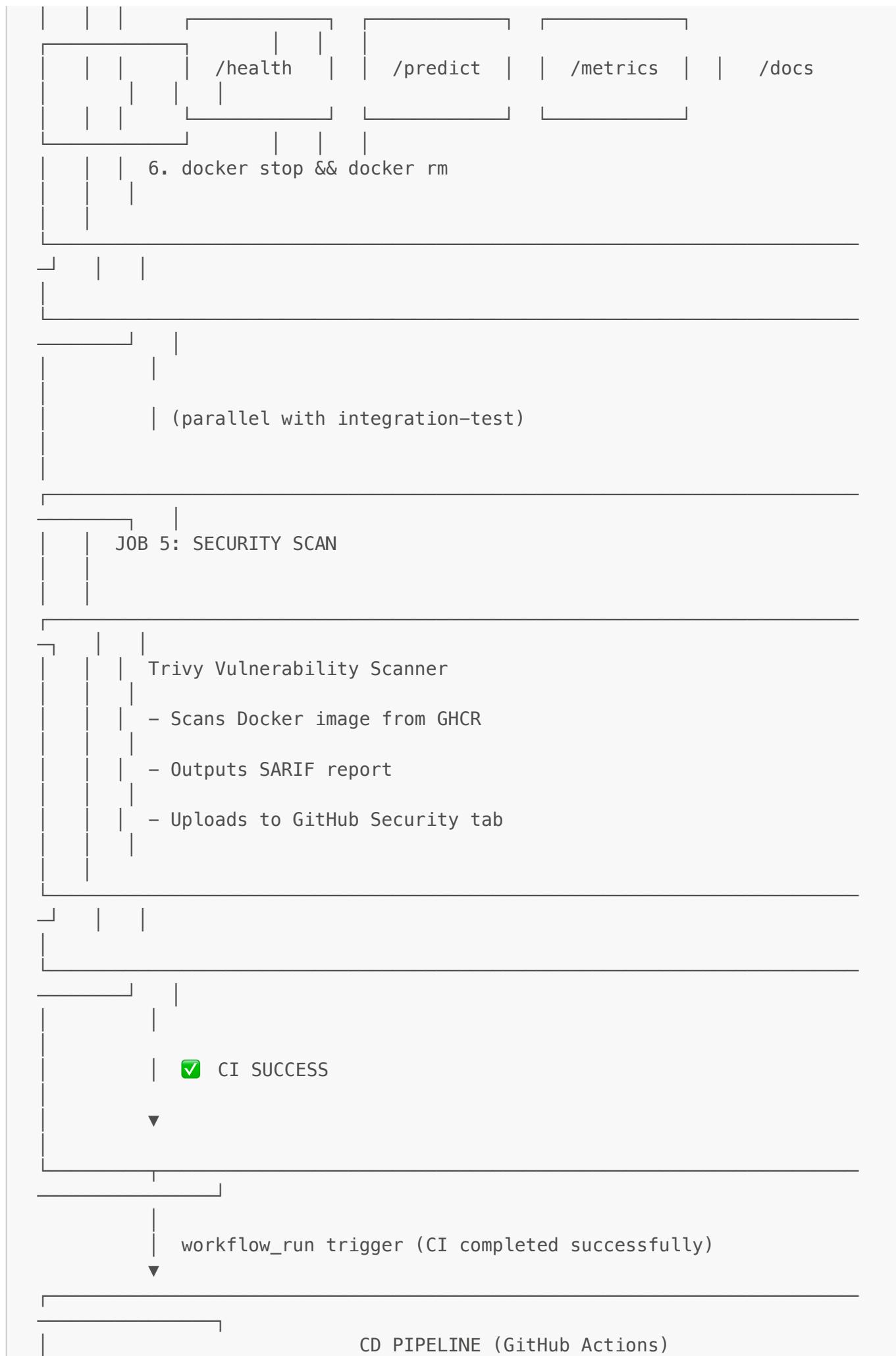


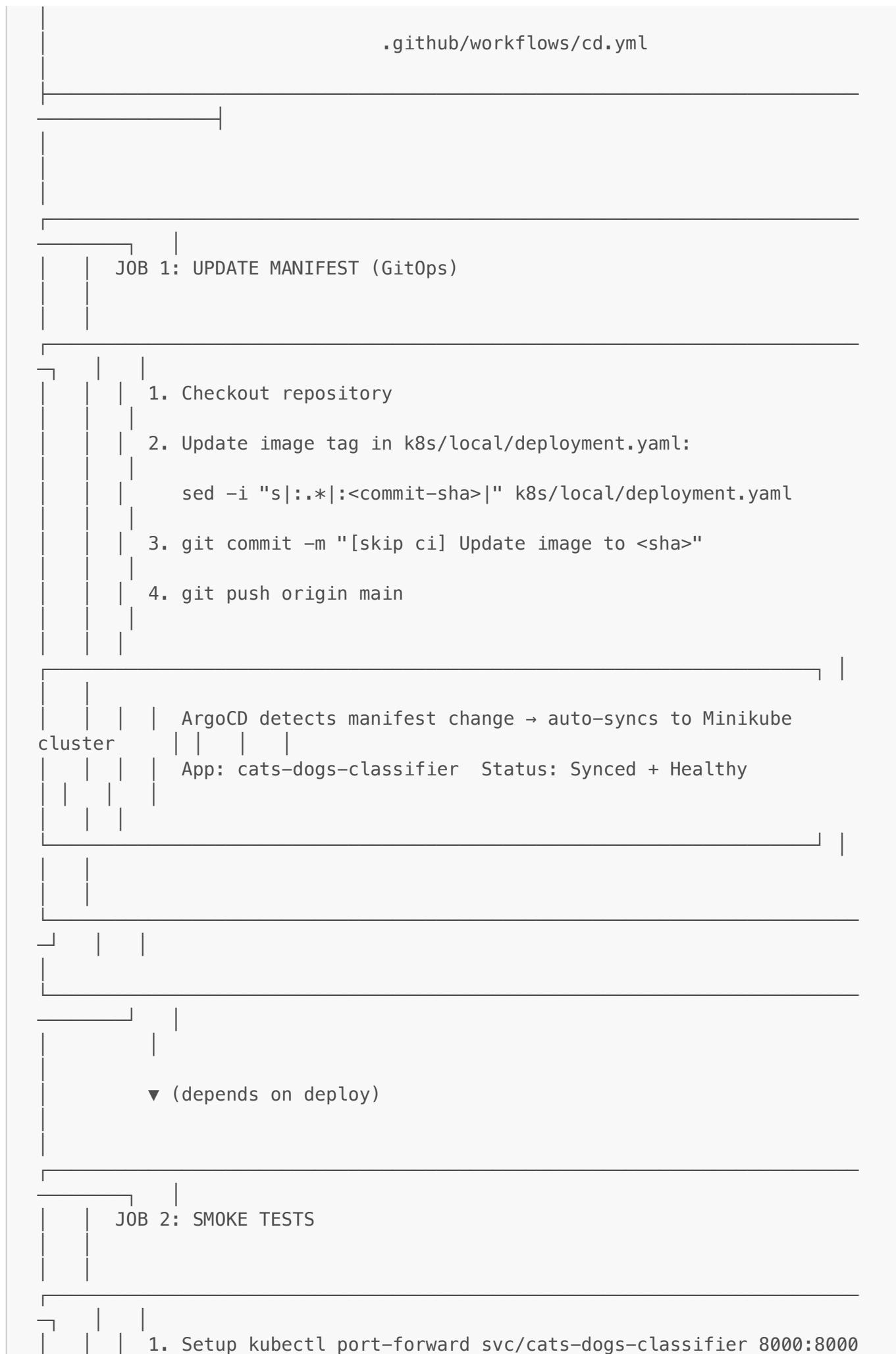


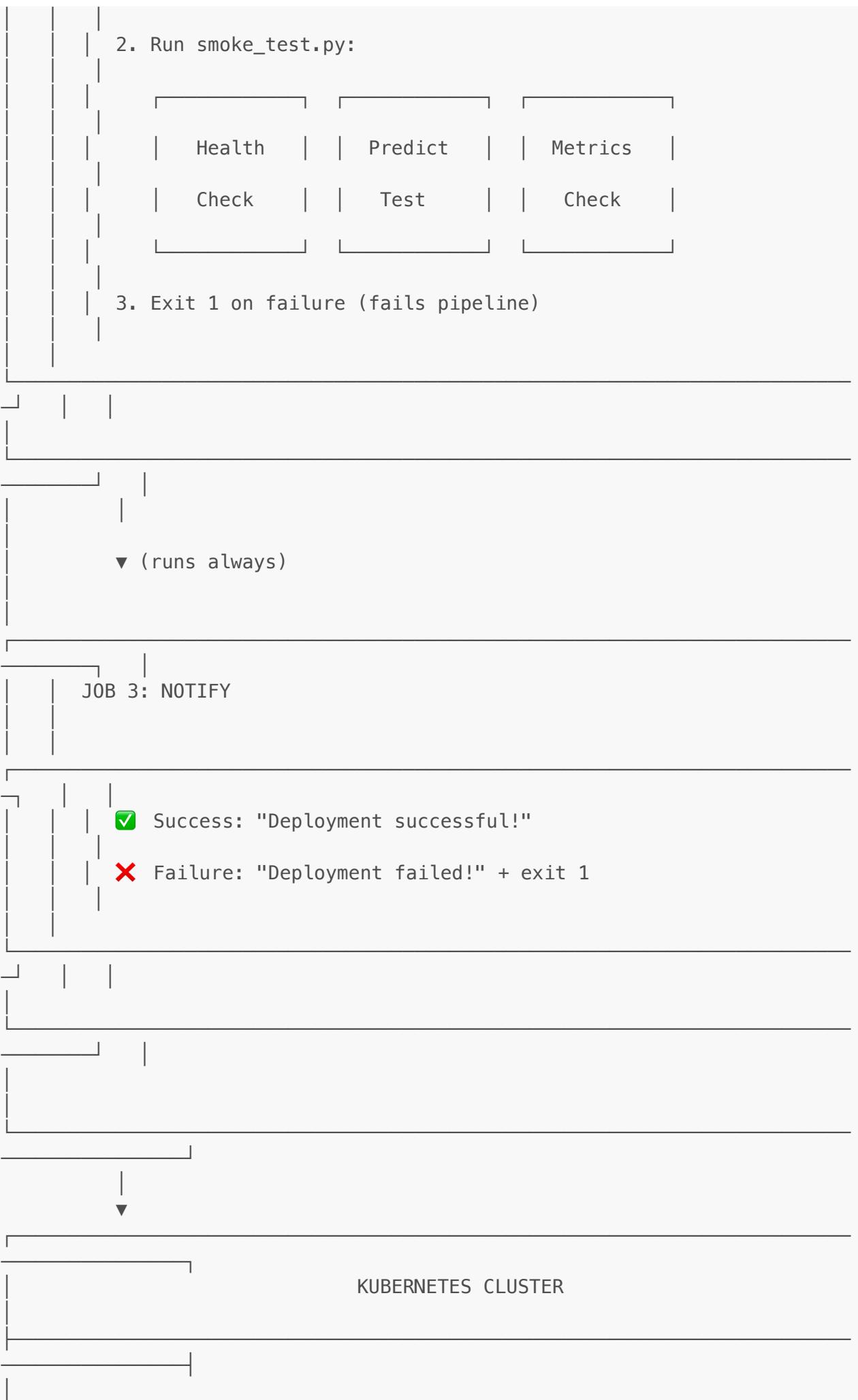


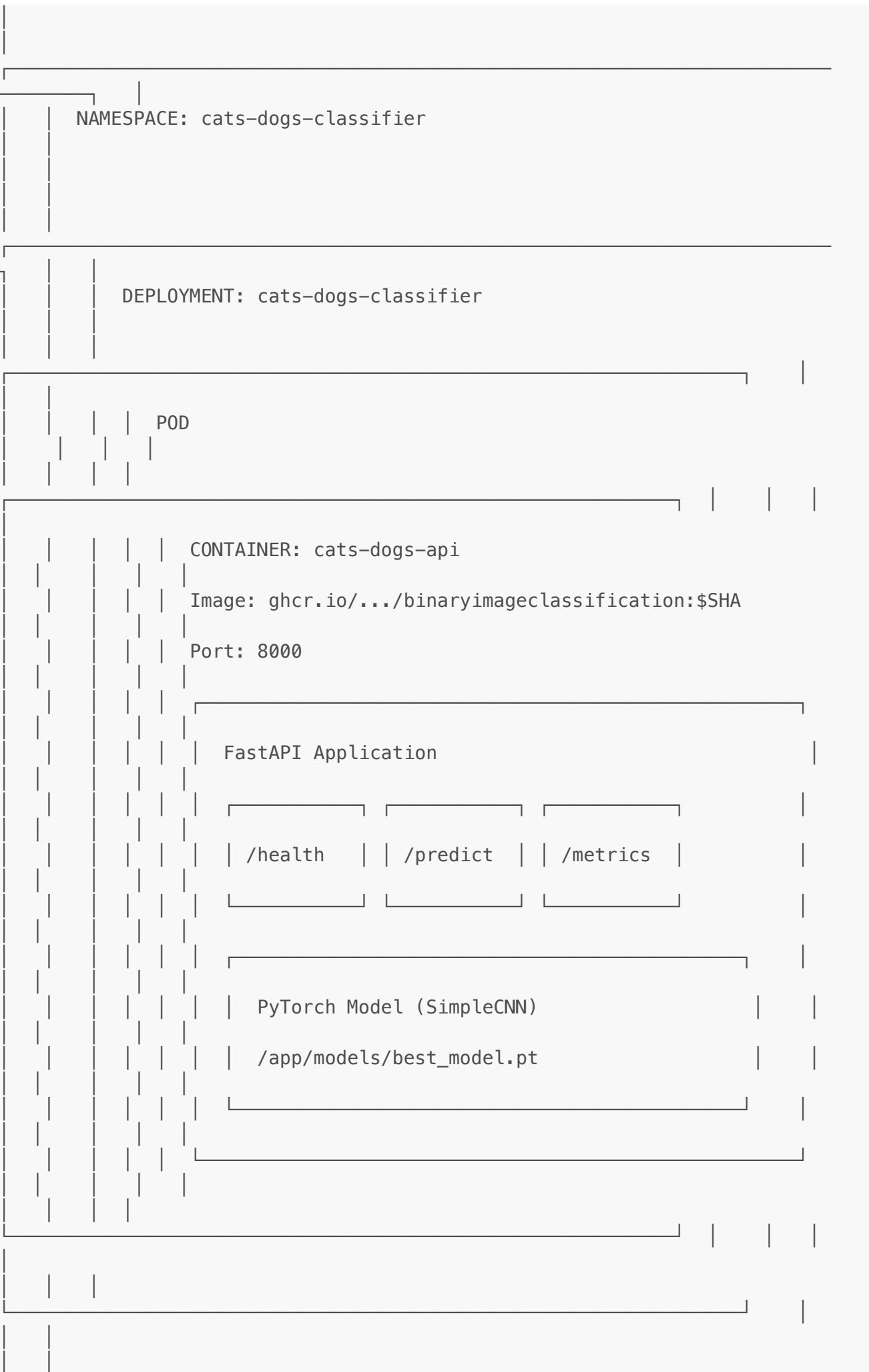


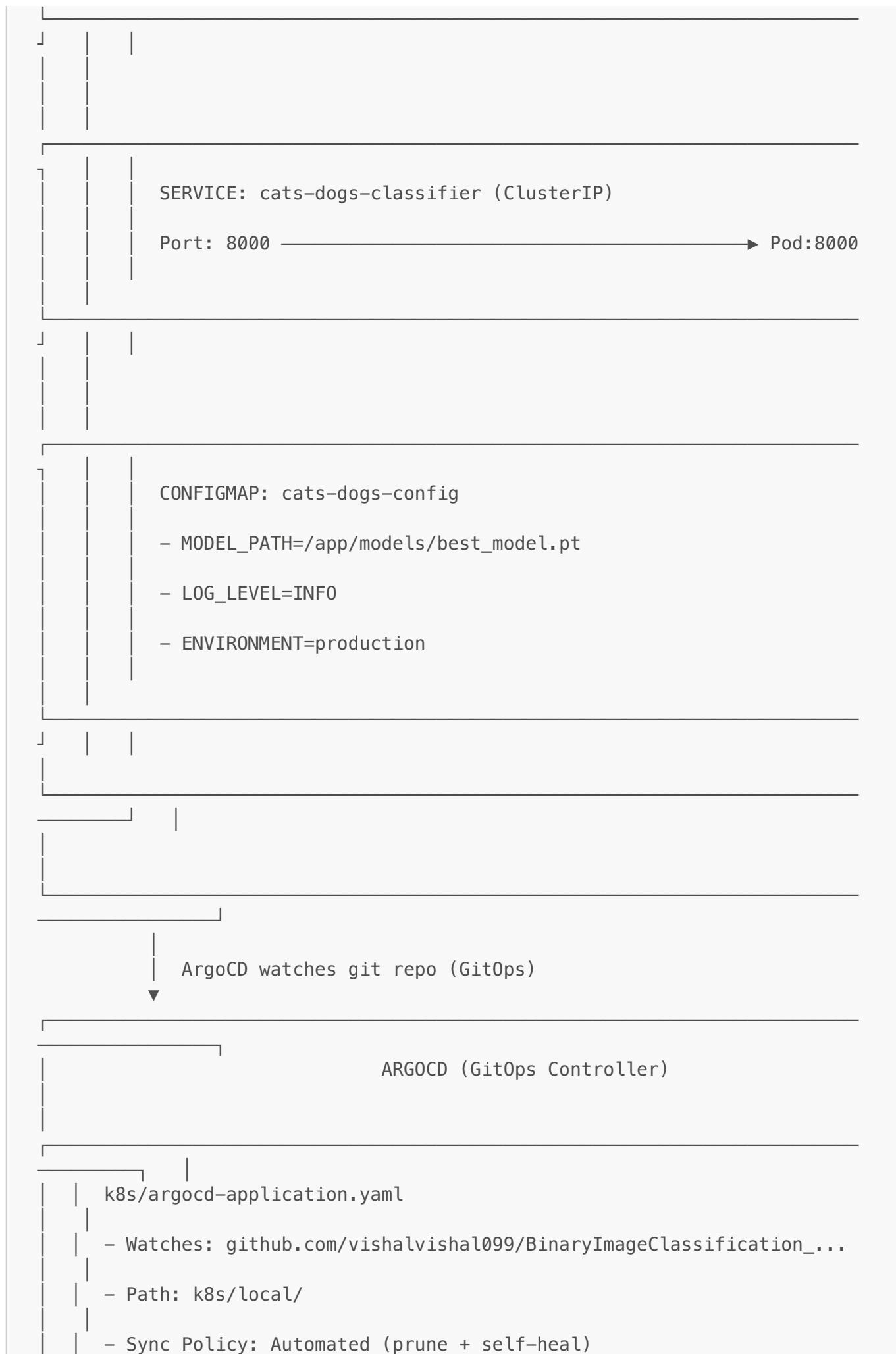


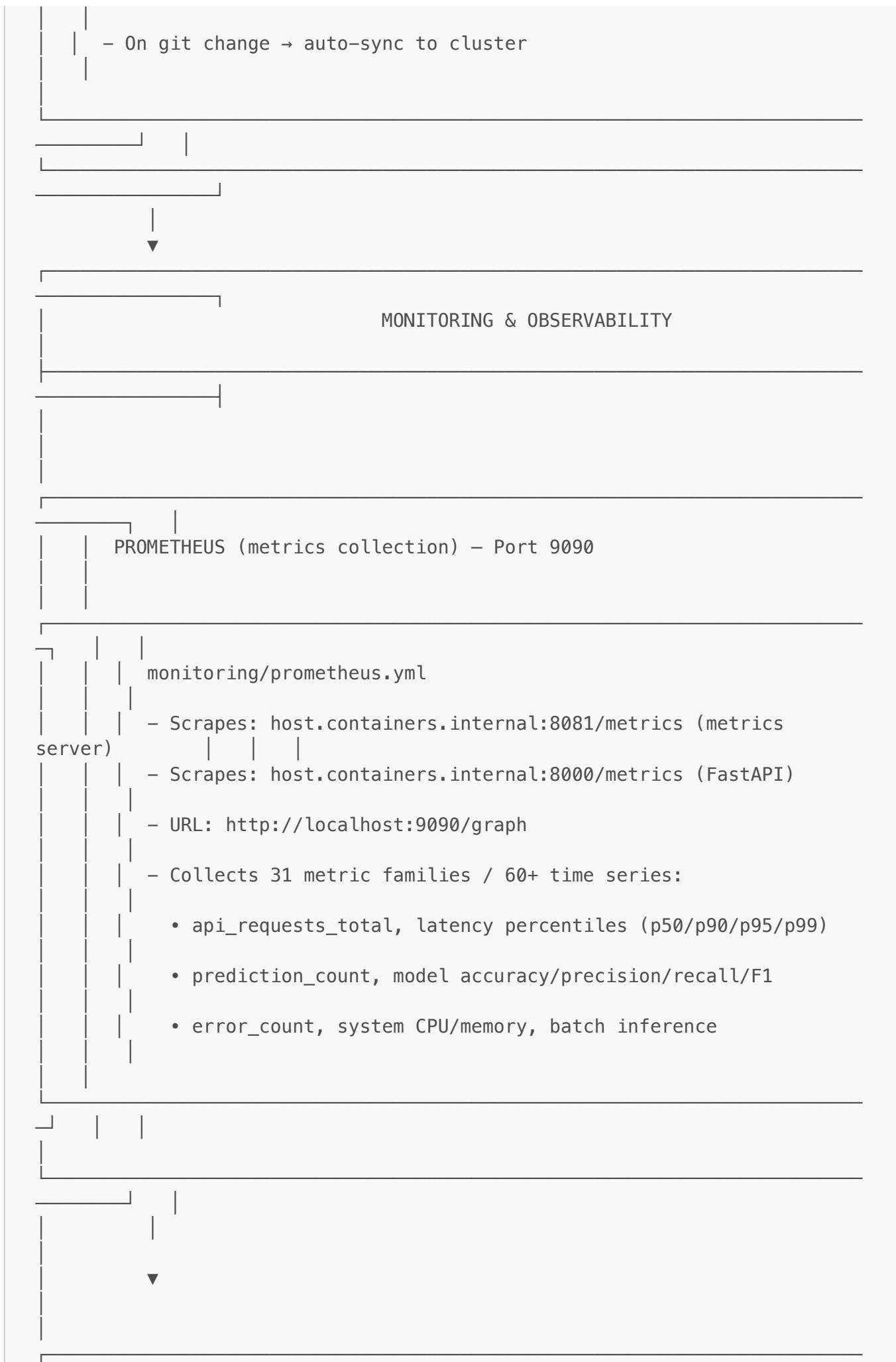








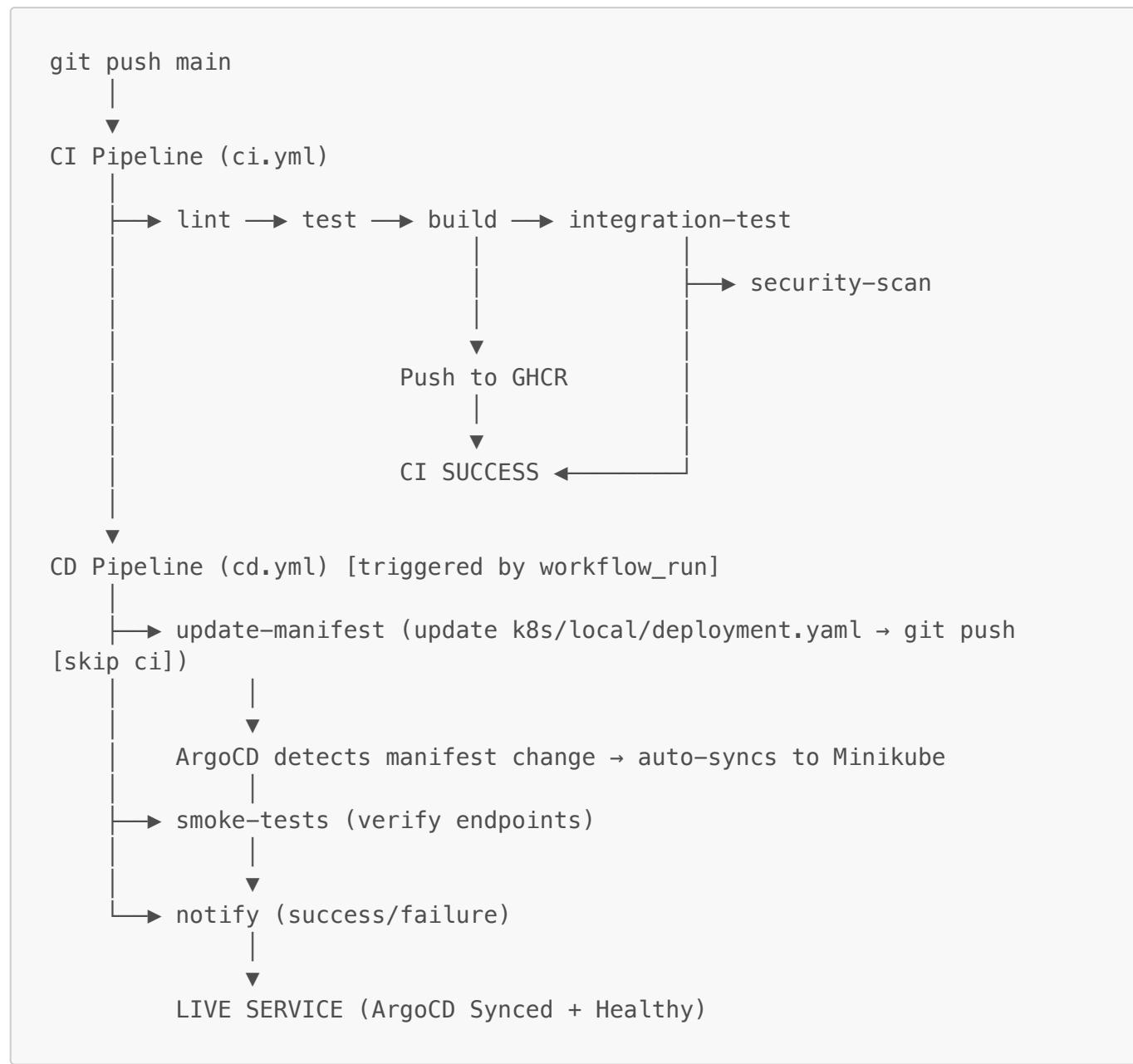






## ⌚ Flow Summary

### Trigger Chain

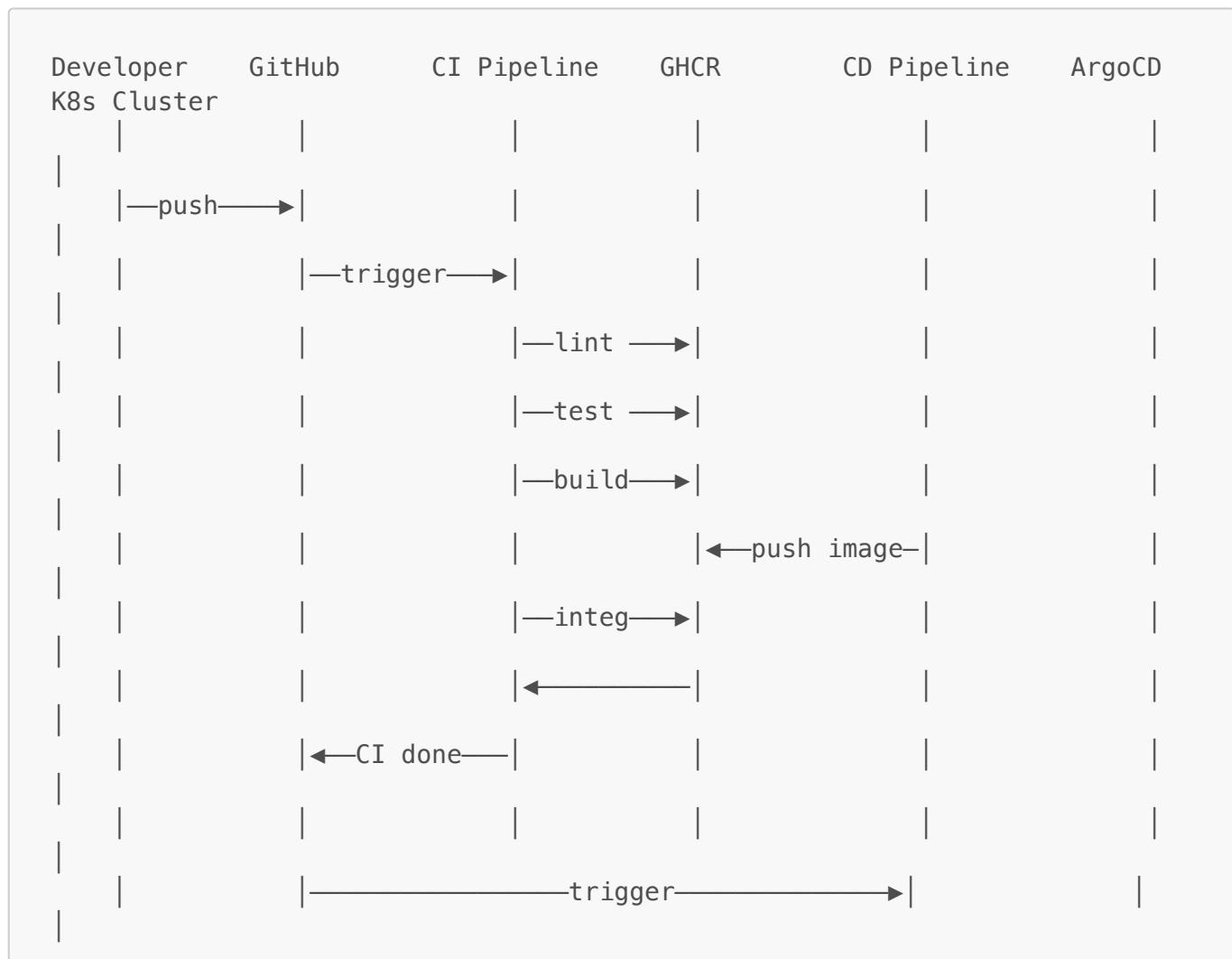


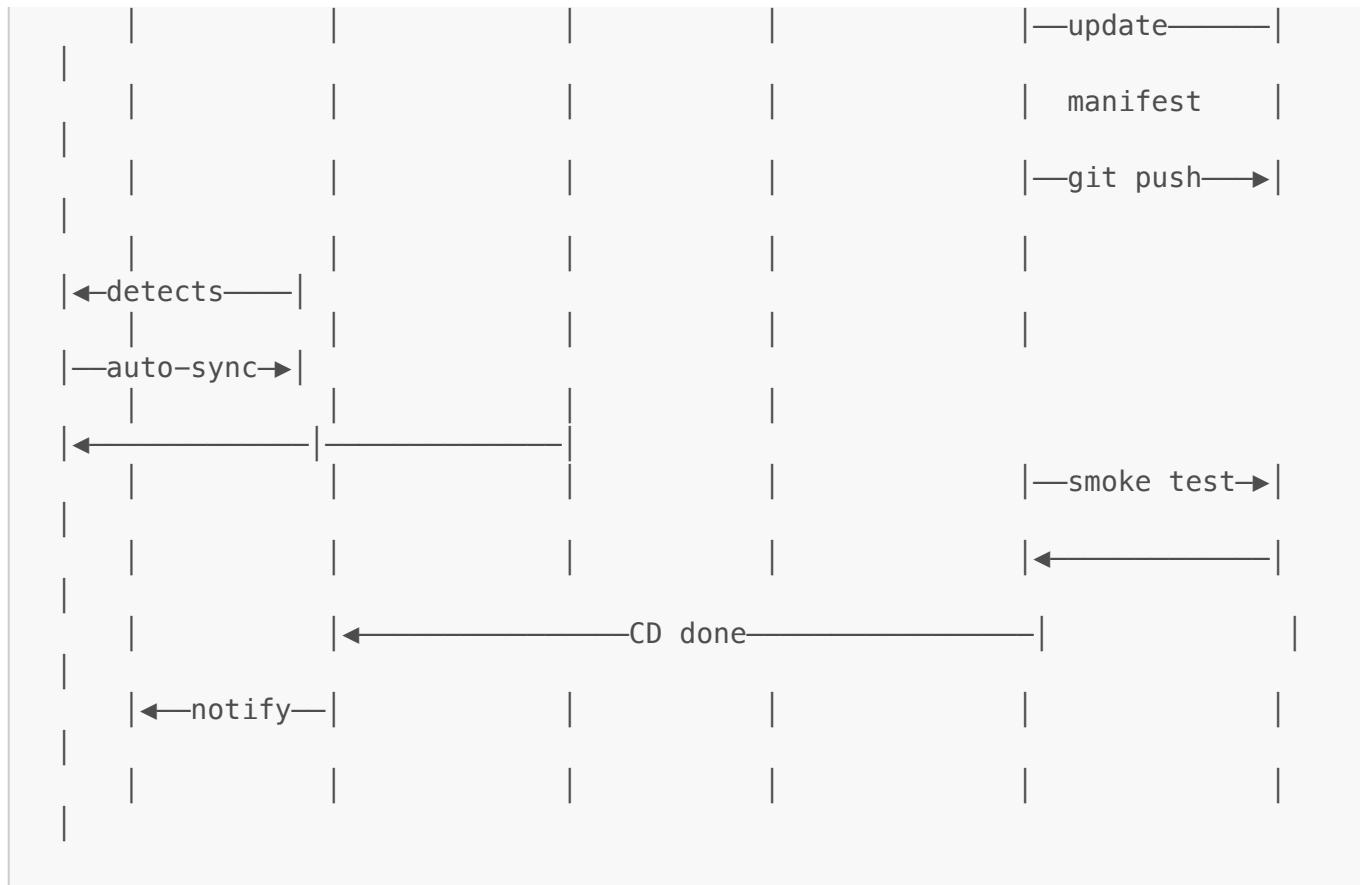
## 🛠 Tools & Technologies

Layer	Tool	Purpose
<b>Version Control</b>	Git + GitHub	Source code management
<b>Data Versioning</b>	DVC + Dagshub	Dataset versioning & cloud storage
<b>Experiment Tracking</b>	MLflow	Metrics, params, artifacts logging
<b>CI/CD</b>	GitHub Actions	Automated pipelines

Layer	Tool	Purpose
<b>Containerization</b>	Docker	Image building
<b>Registry</b>	GHCR	Container image storage
<b>Orchestration</b>	Kubernetes	Container orchestration
<b>GitOps</b>	ArgoCD	Declarative K8s deployment
<b>API Framework</b>	FastAPI	REST API service
<b>ML Framework</b>	PyTorch	Model training & inference
<b>Monitoring</b>	Prometheus	Metrics collection
<b>Visualization</b>	Grafana	Dashboards
<b>Logging</b>	structlog	Structured JSON logging
<b>Testing</b>	pytest	Unit & integration tests
<b>Security</b>	Trivy	Container vulnerability scanning
<b>Code Quality</b>	black, isort, flake8	Linting & formatting

## Sequence Diagram





## 🔐 Secrets Required

Secret Name	Purpose	Where Used
GITHUB_TOKEN	Auto-provided, GHCR auth	CI (build job)
KUBE_CONFIG	Base64-encoded kubeconfig	CD (deploy job)
SERVICE_URL	(Optional) External service URL	CD (smoke tests)

## 📁 Key Files

```

.github/workflows/
└── ci.yml          # CI Pipeline definition
└── cd.yml          # CD Pipeline definition

k8s/local/
└── namespace.yaml  # K8s namespace
└── configmap.yaml  # Environment config
└── deployment.yaml # Pod deployment
└── service.yaml    # Service exposure

k8s/
└── argocd-application.yaml # ArgoCD GitOps config

monitoring/
└── prometheus.yml    # Prometheus scrape config

```

```
└─ grafana/
    └─ provisioning/
        └─ datasources/
            └─ datasource.yml # Grafana datasource

scripts/
└─ smoke_test.py      # Post-deploy verification
└─ simulate_performance.py # Load testing
```