

# GitOps s K3s, Argo CD a GitHub Container Registry (GHCR) - Kompletná príručka

## Obsah

1. Úvod do GitOps
  2. K3s - ľahká Kubernetes distribúcia
  3. Argo CD - GitOps nástroj
  4. Kubernetes základné pojmy
  5. Secrets a ConfigMaps
  6. Struktúra Git repozitára pre GitOps
  7. Kustomize - úprava manifestov
  8. GHCR (GitHub Container Registry)
  9. GitHub Actions pre CI/CD
  10. Nasadenie pipeline: GitHub → GHCR → Argo CD → K3s
  11. Kontrola a troubleshooting
  12. Vizualizácia pipeline
- 

## 1. Úvod do GitOps

- GitOps je prístup, kde je **Git repozitár zdrojom pravdy** pre infraštruktúru a aplikácie.
- Automatické nasadenie zmien pomocou CI/CD pipeline a nástrojov ako Argo CD.
- Výhody: auditovateľnosť, reprodukovateľnosť, rollback.

## 2. K3s - ľahká Kubernetes distribúcia

- Ideálne pre lokálny vývoj a VM s nízkymi zdrojmi.
- Obsahuje všetky Kubernetes komponenty, ale je optimalizovaná.
- Inštalácia:

```
curl -sfL https://get.k3s.io | sh -
kubectl get nodes
```

## 3. Argo CD - GitOps nástroj

- Sleduje Git repozitár a automaticky synchronizuje Kubernetes cluster.
- Podporuje automatické synchronizácie, self-healing a rollbacks.
- Základné príkazy:

```
argocd login localhost:8080 --insecure
argocd app list
```

```
argocd app create -f argocd/app.yaml  
argocd app sync <app-name>
```

## 4. Kubernetes základné pojmy

- **Namespace:** logické oddelenie zdrojov.
- **Pod:** základná jednotka, obsahuje kontajnery.
- **Deployment:** spravuje repliky podov.
- **Service:** expose pody na sieti.
- **StatefulSet:** pre stavové aplikácie (DB).
- **Secret:** uchováva citlivé dátá (heslá, tokeny).
- **ConfigMap:** uchováva ne-citlivé konfigurácie.

## 5. Secrets a ConfigMaps

- Secrets sa ukladajú zakódované base64.
- Príklad získania Argo CD admin hesla:

```
kubectl -n argocd get secret argocd-initial-admin-secret -o  
jsonpath="{.data.password}" | base64 --decode
```

## 6. Struktúra Git repozitára pre GitOps

```
git_ops-K8-config/  
├── base/          # základné manifesty  
│   └── insurance-app/  
│       ├── deployment.yaml  
│       ├── service.yaml  
│       └── kustomization.yaml  
├── overlays/      # prostredie (prod, dev)  
│   └── prod/insurance-app/  
│       ├── deployment-patch.yaml  
│       ├── service-patch.yaml  
│       └── kustomization.yaml  
└── argocd/        # Argo CD Application manifesty  
    └── insurance-app.yaml  
└── .github/workflows/ # GitHub Actions  
    └── ci-ci.yaml
```

## 7. Kustomize - úprava manifestov

- Umožňuje prepísať deploymenty, images, služby bez úpravy originálov.
- Príklad prepísania image pre produkciu:

```
images:
  - name: placeholder/insuranceapp
    newName: ghcr.io/<username>/insuranceapp
    newTag: latest
```

- Patches umožňujú zmeny konkrétnych manifestov (napr. Service type LoadBalancer).

## 8. GHCR (GitHub Container Registry)

- Private/public registry pre Docker image.
- Personal access token (PAT) s právami `read:packages`, `write:packages`.
- Príkaz na login:

```
echo <GH_TOKEN> | docker login ghcr.io -u <username> --password-stdin
```

## 9. GitHub Actions pre CI/CD

- Workflow `.github/workflows/ci-cd.yaml` builduje image a pushuje do GHCR.
- Príklad:

```
jobs:
  build-and-push:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3
      - uses: docker/login-action@v2
        with:
          registry: ghcr.io
          username: ${{ github.actor }}
          password: ${{ secrets.GHCR_PAT }}
      - run: |
          docker build -t ghcr.io/${{ github.repository_owner }}/
          insuranceapp:latest ./base/insurance-app
      - run: docker push ghcr.io/${{ github.repository_owner }}/
          insuranceapp:latest
```

## 10. Nasadenie pipeline: GitHub → GHCR → Argo CD → K3s

1. GitHub Action builduje a pushuje image do GHCR.
2. Kustomize prepíše deployment image na GHCR image.
3. Argo CD sleduje Git repositár a sync-uje aplikácie.
4. K3s cluster nasadí novú verziu aplikácie.

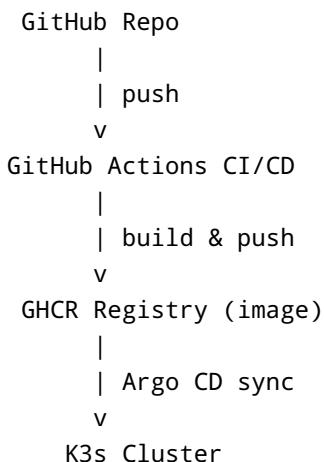
## 11. Kontrola a troubleshooting

- Lokálne:

```
kubectl get pods -n insurance-app  
kubectl describe pod <pod-name> -n insurance-app
```

- Argo CD UI:
  - URL: <https://localhost:8080/applications/<app-name>>
  - Sleduj **Sync Status** a **Health Status**
  - Chyby:
    - **InvalidSpecError** : skontroluj cestu v **path:** v Argo CD manifestoch
    - **ErrImagePull** : image ešte neexistuje alebo login do GHCR

## 12. Vizualizácia pipeline



- Všetky zmeny Git repozitára sa synchronizujú do clusteru cez Argo CD. - GHCR zabezpečuje, že image je dostupný aj pre privátne aplikácie.

---

Táto príručka obsahuje kompletný workflow a vysvetlenie všetkých krokov, ktoré sme prešli. Môžeš ju použiť ako **učebný materiál** a referenciu pre budúce projekty.