

# Tekton

- [☐☐ Tekton Cheat Sheet](#)

# ?? Tekton Cheat Sheet

Cette page regroupe les commandes les plus utiles pour administrer les pipelines **Tekton** utilisés sur la plateforme Numericare.

Tekton est utilisé principalement pour :

- Maintenance automatisée de la plateforme
- Build des images Docker
- Exécution des pipelines CI/CD Kubernetes

## ? Commandes tkn

Toutes les commandes suivantes utilisent le client **tkn**.

### PipelineRuns

| Action                                   | Commande                                                  |
|------------------------------------------|-----------------------------------------------------------|
| Liste des exécutions                     | <code>tkn pr list -n cicd</code>                          |
| Afficher les logs du dernier run         | <code>tkn pr logs -L -n cicd</code>                       |
| Suivre les logs en temps réel            | <code>tkn pr logs -f -n cicd</code>                       |
| Détails d'un PipelineRun                 | <code>tkn pr describe &lt;pipeline-run&gt; -n cicd</code> |
| Supprimer un PipelineRun                 | <code>tkn pr delete &lt;pipeline-run&gt; -n cicd</code>   |
| Conserver uniquement les 5 derniers runs | <code>tkn pr delete --keep 5 -n cicd</code>               |

# ? Lancer un Pipeline

Lancer manuellement le pipeline de maintenance.

```
tkn pipeline start maintenance \  
  -n cicd \  
  --showlog \  
  --use-param-defaults \  
  --workspace name=shared-workspace,claimName=cicd-pvc
```

## ? Pipelines de la plateforme

### maintenance

#### Objectif

- Mises à jour système
- Maintenance Linux
- Notification Telegram

#### Fréquence

Tous les dimanches à 23h00.

### docker-prune

#### Objectif

- Nettoyage Docker
- Suppression des images inutilisées
- Libération de l'espace disque

#### Fréquence

Tous les samedis à 23h30.

### build-code-server

## Objectif

- Construction de l'image Code-Server
- Publication dans le registre Docker privé

## Tâches principales

- git-clone
  - kaniko
  - docker push
- 

# ? Inspection

## Voir les pipelines disponibles

```
tkn pipeline list -n cicd
```

## Voir les tâches disponibles

```
tkn task list -n cicd
```

## Voir les PipelineRuns

```
tkn pr list -n cicd
```

---

# ? Dépannage

## Pipeline bloqué en Running

Vérifier le PVC :

```
kubectl get pvc -n cicd
```

Le volume doit être dans l'état **Bound**.

---

# Erreur Git

Host key verification failed

Mettre à jour le fichier **known\_hosts**.

```
ssh-keyscan github.com
```

# OOMKilled

Augmenter les ressources mémoire de la Task concernée.

# ? Nettoyage

Supprimer tous les anciens PipelineRuns.

```
tkn pr delete --keep 5 -n cicd
```

# ? Emplacement des Pipelines

```
infra/kubernetes/cicd/  
├─ pipelines/  
├─ tasks/  
├─ triggers/  
├─ cronjobs/  
└─ resources/
```

# ? Bonnes pratiques

- Ne jamais exécuter un pipeline directement en production sans validation.
- Consulter les logs avant de relancer un PipelineRun.
- Utiliser les CronJobs pour les tâches récurrentes.
- Nettoyer régulièrement les anciens PipelineRuns.
- Versionner les pipelines dans Git.

- Laisser ArgoCD synchroniser les ressources Tekton.
- 

## ? Documentation associée

- GitOps
- ArgoCD
- Kubernetes
- Helm
- Ansible
- Docker Registry