

# Helm

- [Installation HelmCLI](#)
- [Helm commande](#)

# Installation HelmCLI

Installation helm, regarder si vous allez bien pointé sur la dernière version d'helm par rapport à la publication de cette rubrique.

```
curl -fsSL -o get_helm.sh https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3
chmod 700 get_helm.sh
ls /usr/local/bin/helm
export PATH=$PATH:/usr/local/bin
./get_helm.sh
```

## TEST HELM INSTALL :

```
helm version
helm repo add stable https://charts.helm.sh/stable
helm repo update
```

# Helm commande

Commande helm pour tester son template values sur un cluster openshift ou kubernetes :

```
helm template onyx ./ -f values-recette.yaml --namespace onyx-recette --debug
```

## Décomposition de la commande:

- **helm template** : C'est la commande principale qui indique à Helm que nous voulons générer un template.
- **onyx** : C'est le nom du chart Helm que vous souhaitez utiliser comme base pour générer le manifest. Un chart Helm est un package qui contient les fichiers de configuration nécessaires pour déployer une application sur Kubernetes.
- **./** : Ce point indique le répertoire courant. Cela signifie que le chart Helm **onyx** se trouve dans le répertoire où vous exécutez cette commande.
- **-f values-recette.yaml** : Ce drapeau indique à Helm d'utiliser le fichier **values-recette.yaml** comme fichier de valeurs. Ce fichier contient les valeurs spécifiques à votre déploiement, comme les noms d'hôtes, les ports, etc.
- **--namespace onyx-recette** : Ce drapeau spécifie que le manifest généré sera déployé dans l'espace de noms **onyx-recette** de votre cluster Kubernetes.
- **--debug** : Ce drapeau active le mode débogage, ce qui signifie que Helm affichera plus d'informations sur le processus de génération du template.

-----  
-----  
d'autre commande arrive seront placée par la suite