

pourquoi les noms d'utilisateurs affichés par les commandes kubectl auth whoami et kubectl config get-users ne coïncident pas ?

Les noms d'utilisateurs affichés par `kubectl auth whoami` et `kubectl config get-users` ne coïncident souvent pas car ils représentent deux notions différentes dans Kubernetes :

`kubectl config get-users`

- Affiche **les utilisateurs définis dans le fichier kubeconfig local**.
- Ces « utilisateurs » correspondent à des profils de connexion (par exemple, un utilisateur OIDC, un compte de service, une identité Azure, etc.), mais **ce ne sont pas forcément les identités réelles utilisées côté serveur** ^[1].

`kubectl auth whoami`

- Affiche **l'identité réellement reconnue par le serveur Kubernetes** pour la session en cours.
- Cette identité est déterminée après l'authentification, selon le mécanisme configuré (OIDC, webhook, Azure AD, etc.) et peut être différente du nom configuré localement dans le kubeconfig ^[2] ^[3].
- Par exemple, avec Azure AD ou OIDC, le nom utilisateur retourné est souvent un identifiant unique (UPN, e-mail, ou un identifiant interne), mappé par le serveur à partir du token d'authentification.

Pourquoi cette différence ?

- Le kubeconfig définit des profils d'accès locaux, qui peuvent utiliser divers mécanismes d'authentification (certificats, tokens, plugins, etc.). Le nom dans `get-users` est **un alias local** pour le profil d'authentification, pas nécessairement l'identité effective côté API server ^[1] ^[3].
- Le serveur Kubernetes, lors de l'authentification, **interprète le token ou le certificat** fourni et attribue une identité (username) qui peut être différente, notamment avec les systèmes d'authentification dynamiques (OIDC, Azure AD, etc.) ^[2] ^[3].
- Ainsi, il est courant que le nom d'utilisateur « local » (dans kubeconfig) et l'identité réelle (côté serveur) diffèrent, surtout avec les plugins d'authentification ou les providers cloud.

Résumé

- `kubectl config get-users` = profils locaux dans kubeconfig.
- `kubectl auth whoami` = identité réelle reconnue par le serveur Kubernetes pour la session courante.
- La différence est normale et attendue, surtout avec les authentifications dynamiques (OIDC, Azure AD, etc.) ^[2] ^[3].

~

1. https://kubernetes.io/docs/reference/kubectl/generated/kubectl_config/kubectl_config_get-users/
2. https://kubernetes.io/docs/reference/kubectl/generated/kubectl_auth/kubectl_auth_whoami/
3. <https://kubernetes.io/docs/reference/access-authn-authz/authentication/>