

JENKINS: Gestion des identifiants

Table of Contents

- [JENKINS: Gestion des identifiants](#)
- [Utilisation des identifiants](#)
 - [Types d'identifiants gérés](#)
 - [Sécurité des identifiants](#)
 - [Configuration des identifiants](#)
 - [Ajout d'identifiants globaux](#)
- [Utilisation de l'API](#)
 - [Exemple d'utilisation lookupCredentials\(\)](#)
 - [Méthode getPassword](#)
 - [Boucle principale](#)

Le plug-in Credentials Binding fournit la fonctionnalité d'informations d'identification Jenkins.

Les informations d'identification stockées dans Jenkins peuvent être utilisées:

- de façon globale partout dans Jenkins,
- par un projet / élément de pipeline spécifique,
- par un utilisateur spécifique de Jenkins (comme dans le cas des projets Pipeline créés dans Blue Ocean).

Utilisation des identifiants

Types d'identifiants gérés

Jenkins peut stocker les types d'informations d'identification suivants:

- **Texte secret:** un jeton tel qu'un jeton d'API (par exemple, un jeton d'accès personnel GitHub),
- **Nom d'utilisateur et mot de passe:** qui peuvent être gérés en tant que composants séparés ou en tant que chaîne séparée par des points-virgules au format username:password (pour plus d'informations à ce sujet, reportez-vous à la section Gestion des informations d'identification),
- **Fichier secret:** qui est essentiellement le contenu secret dans un fichier,
- **Nom d'utilisateur SSH avec clé privée:** une paire de clés publique / privée SSH ,
- **Certificat:** un fichier de certificat PKCS # 12 et un mot de passe optionnel, ou
- **Informations d'identification d'authentification du certificat d'hôte Docker.**

Sécurité des identifiants

Pour maximiser la sécurité, les informations d'identification configurées dans Jenkins sont stockées

sous une forme cryptée sur l'instance principale de Jenkins (cryptée par l'ID d'instance Jenkins) et ne sont gérées que dans les projets Pipeline via leurs ID d'identification.

Cela minimise les chances d'exposer les informations d'identification réelles aux utilisateurs de Jenkins et empêche la possibilité de copier des informations d'identification fonctionnelles d'une instance de Jenkins à une autre.

Configuration des identifiants

Cette section décrit les procédures de configuration des informations d'identification dans Jenkins.

Les informations d'identification peuvent être ajoutées à Jenkins par tout utilisateur Jenkins disposant de l'autorisation Informations d'identification> Créer (défini via la sécurité basée sur Matrix). Ces autorisations peuvent être configurées par un utilisateur Jenkins avec l'autorisation Administrer . Plus d'informations à ce sujet dans la section Autorisation de la rubrique Gestion de la sécurité .

Sinon, tout utilisateur Jenkins peut ajouter et configurer des informations d'identification si les paramètres d'autorisation de la page Configurer les paramètres de sécurité globale de l'instance Jenkins sont définis sur le paramètre par défaut. Les utilisateurs connectés peuvent définir n'importe quel paramètre ou tout le monde .

Ajout d'identifiants globaux

Pour ajouter de nouvelles informations d'identification globales à une instance Jenkins:

- Si nécessaire, d'assurer d'être connecté à Jenkins (en tant qu'utilisateur disposant des informations d'identification> autorisation de création).
- Dans la page d'accueil de Jenkins (c.-à-d. Le tableau de bord de l'interface utilisateur classique de Jenkins), cliquer sur Informations **identifiants>system** à gauche.
- Sous **system** , cliquer sur le lien **Identifiants globaux (illimité)** pour accéder à ce domaine par défaut.
- Cliquer sur **Ajouter des identifiants** à gauche.
- Dans le champ Type, choisir le type d'informations d'identification à ajouter.
- Dans le champ Scope, choisir soit:
 - **Global** - si les informations d'identification à ajouter sont destinées à un projet / élément de pipeline. Le choix de cette option applique l'étendue des informations d'identification à l'"objet" de projet / élément de pipeline et à tous ses objets fils.
 - **System** - si le ou les identifiants à ajouter sont utilisés par l'instance Jenkins pour interagir avec les fonctions d'administration du système, telles que l'authentification du courrier électronique, la connexion à l'agent, etc. Cette option applique la portée du ou des identifiants à un seul objet. seulement.
- Ajouter les informations d'identification elles-mêmes dans les champs appropriés pour le type d'informations d'identification choisi:
 - **Texte secret** - copier le texte secret et le coller dans le champ Secret .
 - **Nom d'utilisateur et mot de passe** - spécifier le nom d'utilisateur et le mot de passe des informations d'identification dans leurs champs respectifs.
 - **Fichier secret** - cliquer sur le bouton Choisir un fichier en regard du champ Fichier pour sélectionner le fichier secret à télécharger vers Jenkins.

- **Nom d'utilisateur SSH avec clé privée** - spécifier les informations d'identification Nom d'utilisateur , clé privée et phrase secrète facultative dans leurs champs respectifs.
 - **Certificat** - spécifier le certificat et le mot de passe facultatif. Choisir le certificat de téléchargement PKCS#12 permet de télécharger le certificat sous forme de fichier via le bouton de certificat de téléchargement résultant.
 - **Authentification du certificat d'hôte Docker** - copier et coller les détails appropriés dans les champs Clé client, Certificat client et Certificat de l'autorité de certification du serveur.
- Dans le champ ID , indique une valeur d'identifiant d'identification jenkins-user-for-xyz-artifact-repository , par exemple jenkins-user-for-xyz-artifact-repository . On peut utiliser des lettres majuscules ou minuscules pour l'ID des informations d'identification, ainsi que tout caractère séparateur valide. Ce champ ID est facultatif. Si on ne spécifie pas de valeur, Jenkins attribue l'identificateur global unique (GUID) à l'ID des informations d'identification. Une fois un identifiant d'identification défini, il ne peut plus être modifié
 - Spécifier une description facultative pour les informations d'identification.
 - Cliquer sur OK pour enregistrer les informations d'identification.

Utilisation de l'API

Le plug-in credentials fournit une API normalisée permettant à d'autres plug-in de stocker et de récupérer différents types d'informations d'identification.

Exemple d'utilisation lookupCredentials()

L'exemple de Groovy script ci-dessous illustre comment utiliser **lookupCredentials()** de **CredentialsProvider** pour récupérer des informations d'identification.

Le but de ce script est de construire une liste de paramètre contenant les identifiants/username/mot de passe pour une utilisation dans un build.

```
import jenkins.*
import jenkins.model.*
import hudson.*
import hudson.model.*
import com.cloudbees.plugins.credentials.Credentials
import com.cloudbees.plugins.credentials.CredentialsProvider;
import
com.cloudbees.plugins.credentials.common.StandardUsernamePasswordCredentials
;
def getPassword = { userid ->
    def password = ""
    def creds =
com.cloudbees.plugins.credentials.CredentialsProvider.lookupCredentials(
    com.cloudbees.plugins.credentials.Credentials.class
)
Jenkins.instance.getAllItems(com.cloudbees.hudson.plugins.folder.Folder.clas
s).findAll{it.name.equals('reseau')}.each{ f ->
```

```

    creds =
com.cloudbees.plugins.credentials.CredentialsProvider.lookupCredentials(
    com.cloudbees.plugins.credentials.Credentials.class, f)
    def c = creds.findResult { it.id == userid ? it : null }

    if ( c ) {
        password = c.password

    } else {
        password = "n-a"
    }
}
return password
}
Set<Credentials> allCredentials = new HashSet<Credentials>();

Jenkins.instance.getAllItems(com.cloudbees.hudson.plugins.folder.Folder.class).findAll{it.name.equals('reseau')}.each{ f ->
    creds =
com.cloudbees.plugins.credentials.CredentialsProvider.lookupCredentials(
    com.cloudbees.plugins.credentials.Credentials.class, f)
    allCredentials.addAll(creds)
}

def choices=[]

for (c in allCredentials) {
    def passwd = getPassword(c.id)
    choices.add(c.id + "|" + c.username + "|" + passwd)
}
return choices

```

Méthode getPassword



Une méthode en Groovy est définie avec le mot clé `def`. Les méthodes peuvent recevoir un nombre quelconque d'arguments. Il n'est pas nécessaire que les types soient explicitement définis lors de la définition des arguments

La méthode **getPassword()** permet d'extraire un mot de passe correspondant à l'ID fournit en argument en utilisant `CredentialsProvider.lookupCredentials`.

- Pour tous les objets de l'instance appartenant au dossier 'reseau'

```

Jenkins.instance.getAllItems(com.cloudbees.hudson.plugins.folder.Folder.class).findAll{it.name.equals('reseau')}.each{ f ->

```

- Inspection des objets credentials

```
creds =  
com.cloudbees.plugins.credentials.CredentialsProvider.lookupCredentials(  
    com.cloudbees.plugins.credentials.Credentials.class, f)
```

- Récupération du mot de passe correspondant à l'ID

```
def c = creds.findResult { it.id == userid ? it : null }
```

Boucle principale

La boucle **Jenkins.instance.getAllItems** utilise `CredentialsProvider.lookupCredentials` pour traiter chaque objet `credentials` afin de construire la liste des paramètres indetifiants/username/mot de passe.

- Pour tous les objets de l'instance appartenant au dossier 'reseau'

```
Jenkins.instance.getAllItems(com.cloudbees.hudson.plugins.folder.Folder.class).findAll{it.name.equals('reseau')}.each{ f ->
```

- Inspection des objets `credentials`

```
creds =  
com.cloudbees.plugins.credentials.CredentialsProvider.lookupCredentials(  
    com.cloudbees.plugins.credentials.Credentials.class, f)
```

- Récupération des identifiants (**c.id**), username (**c.username**) et du mot de passe (**passwd**, par appel de la méthode `getPassword`] pour construire une liste d'arguments séparés par des pipes "|"

```
for (c in allCredentials) {  
    def passwd = getPassword(c.id)  
    choices.add(c.id + "|" + c.username + "|" + passwd)  
}
```

From:

<http://www.ouarte.garden/> - **dwndoc**

Permanent link:

<http://www.ouarte.garden/doku.php?id=outils:jenkins-credentials>

Last update: **2025/02/19 10:59**

