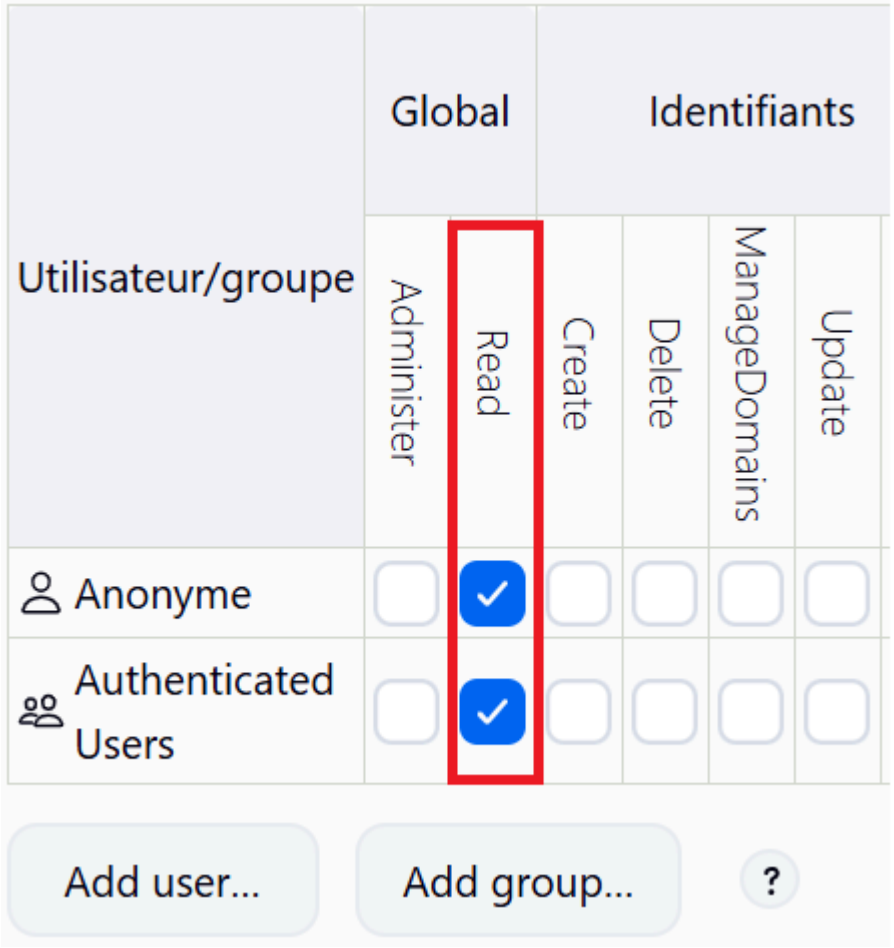
Travaux Pratiques – Jenkins : Authentification et autorisations
Public cible : Administrateurs DevOps
Durée estimée : 2 heures

Prérequis :
• Jenkins installé (version ≥ 2.400)
• Accès administrateur à l'instance Jenkins
• Accorder à minima le droit ✔️ Global: Read à un utilisateur ou un rôle pour qu'il puisse bénéficier des droits qui sont assignés:



TP 1 : Configurer l'authentification locale

Objectif
Mettre en place l'authentification native de Jenkins et créer des utilisateurs locaux.

Étapes
1. Connectez-vous à votre instance Jenkins en tant qu'administrateur.
2. Allez dans Manage Jenkins > Security.
3. Dans Security Realm, sélectionnez :
   ☑ Jenkins's own user database
   ☑ Allow users to sign up

4. Cliquez sur Save.
5. Ouvrez une interface Jenkins puis cliquez sur Sign up et créez :
   - dev1 (mot de passe : dev123)
   - admin1 (mot de passe : admin123)
6. Déconnectez-vous, puis reconnectez-vous avec dev1.

Questions de validation
• Pouvez-vous accéder à Manage Jenkins avec dev1 ?
• Où Jenkins stocke-t-il les utilisateurs locaux ? (indice : $JENKINS_HOME)

TP 2 : Mettre en place Matrix-Based Security

Objectif
Configurer des permissions fines par utilisateur.

Prérequis : TP1 terminé.

Étapes
1. Dans Manage Jenkins > Security > Authorization, choisissez Matrix-based security.
2. Pour admin1 : cochez toutes les permissions.
3. Pour dev1, cochez uniquement :
   - Global: Read
   - Job: Build, Read, Workspace
   - Agent: Delete, Update
   - Vues: Read
4. Testez les actions avec dev1.

Questions de validation
• Quelles actions sont autorisées/refusées ?
• Pourquoi ne peut-il pas créer de job ?
• Que se passe-t-il sans Global/Read ?

TP 3 : Role-Based Strategy

Objectif
Utiliser des rôles pour gérer les accès.

Étapes
1. Installez le plugin "Role-based Authorization Strategy".
2. Dans Authorization, choisissez "Role-Based Strategy" puis sauvegardez.
3. Dans Manage Jenkins > Manage Roles > Créez deux Global Roles :
   - admin (toutes permissions)
   - developer (Read, Build, etc.)
4. Dans Manage Jenkins > Manage Roles >Assign Roles:

- Assignez le rôle admin à admin1
- Assignez le rôle developer à dev1.
- Connectez-vous avec admin1 et vérifiez les permissions
- Connectez-vous avec dev1 et vérifiez les permissions

5. Créez un Item Role "frontend-dev" avec regex FRONTEND_.*
   - Attribuez le rôle frontend-dev à dev1
   - Créez un projet (i.e: un job Jenkins) avec comme nom FRONTEND_
   - Vérifiez que dev1 voit uniquement les projets dont les noms commencent par FRONTEND_

Questions de validation
• Avantage des rôles vs matrice ?
• Comment la regex sécurise-t-elle les jobs ?
• Peut-on assigner plusieurs rôles à un utilisateur Jenkins ?

TP4: Intégration de Jenkins avec LDAP

📌 Prérequis

- Docker et Docker Compose installés

📁 Fichiers fournis

`docker-compose.yml` **(serveur LDAP + phpLDAPadmin)**

`users-and-groups.ldif` **(utilisateurs et groupes)**

Étapes

1. Se connecter sur la VM

   $ vagrant.exe ssh centos

   [vagrant@jenkins ~]$ cd install_files/

2. Installer le serveur LDAP :

   ```
   mkdir -p ldap/{database,config}
   ```

```
docker compose -f ldap-docker-compose.yml up -d
```

3. Importer les utilisateurs et groupes

```
docker run --rm -it \
  --network container:openldap \
  -v "$(pwd)":/home/vagrant/install_files \
  --entrypoint ldapadd \
  osixia/openldap:1.5.0 \
  -x -D "cn=admin,dc=formation,dc=local" -w admin_ldap -f
/home/vagrant/install_files/users-and-groups.ldif
```

---

4. Configurer LDAP dans Jenkins :
    a. Allez dans Manage Jenkins → Configure Global Security
    b. Sous Security Realm, choisissez LDAP
    c. Remplissez :
        i. Server : `ldap://192.168.100.100:389`
        (Sur Linux natif, remplacez par l'IP de la machine hôte)
        ii. Root DN : `dc=formation,dc=local`
        iii. User search base : `ou=people`
        iv. User search filter : `(uid={0})`
        v. Manager DN : `cn=admin,dc=formation,dc=local`
        vi. Manager Password : `admin_ldap`
5. Tester la détection des groupes :
    a. Cliquez sur Test LDAP settings
    b. Essayez avec les logins et mots de passe suivants:
        i. `alice / alice123` → doit afficher le groupe `admins`
        ii. `bob / bob123` → doit afficher le groupe `developers`
        iii. `charlie / charlie123` → doit afficher le groupe `testers`
6. Enregistrer

🧪 TP 5 : Gestion des groupes et autorisations fines

Objectif

Attribuer des droits fins différents selon l'appartenance à un groupe LDAP (`admins`, `developers`, `testers`).

Étapes

1. Vérifiez que les groupes sont bien créés :
    - Accédez à phpLDAPadmin : `http://192.168.100.100:8081`
    - Connectez-vous avec :
        - DN : `cn=admin,dc=formation,dc=local`
        - Mot de passe : `admin_ldap`
    - Vérifiez la présence de `ou=groups` et des 3 groupes.
    - Vérifiez la présence de `ou=people` et des 3 utilisateurs.
2. Configurer les autorisations par groupe :
    - Dans Authorization, choisissez Matrix-based security
    - Cliquez sur Add group et saisissez :
        - `admins (LDAP)` → cochez toutes les cases
        - `developers (LDAP)`→ cochez uniquement
            - ✔️ Global: Read
        - `testers (LDAP)`→ ne cochez aucune case
3. Se déconnecter de l'utilisateur admin
4. Se connecter avec admin et expliquer le résultat
5. Se connecter avec bob et expliquer le résultat
6. Se connecter avec charlie et expliquer le résultat
7. Se connecter avec alice et expliquer le résultat
8. Configurer les autorisations par groupe :
    - Dans Authorization, choisissez Matrix-based security
    - Cliquez sur Add group et saisissez :
        - `developers (LDAP)` → cochez :
            - ✔️ Global: Read
            - ✔️ Job: Build, Cancel, Read, Workspace
            - ✔️ SCM: Tag
        - `testers (LDAP)` → cochez :
            - ✔️ Global: Read
            - ✔️ Job: Read, Build
    - Enregistrez
9. Valider :
    - Reconnectez-vous successivement avec `alice`, `bob`, et `charlie`
    - Vérifiez que les droits correspondent aux rôles attendus

---

📝 Résumé des rôles

alice	admins	Accès total (administrateur)
bob	developers	Créer, lancer, lire les jobs
charlie	testers	Lire et lancer des builds (pas de config)

Questions de validation
• Pourquoi utiliser LDAP en entreprise ?
• Avantages vs authentification locale ?
• Que se passe-t-il si LDAP est hors service ?
• Est-il possible d'utiliser la stratégie d'autorisation "Role-based Authorization Strategy" avec une authentification LDAP (si oui, c'est un TP potentiellement à faire par les plus motivés) ?