

Cloud-init et Profils Incus : Note synthétique

Concepts essentiels

- *Profils Incus* : Modèles de configuration Incus (réseau, stockage, CPU, mémoire, etc.) qui s'appliquent aux instances. C'est la configuration infrastructure.
- *Cloud-init* : Outil qui configure le système d'exploitation au premier démarrage (utilisateurs, paquets, services). C'est la configuration applicative.

Différence clé

- Profil Incus = "Comment créer la machine" (2 CPU, 4GB RAM, réseau bridgé)
- Cloud-init = "Comment configurer le système" (installer nginx, créer utilisateur)

Relation entre les deux

Un profil Incus peut contenir de la configuration cloud-init, mais cloud-init peut aussi être ajouté directement sur une instance.

Règles fondamentales

- vendor-data → dans les profils (configuration générale)
- user-data → sur les instances (configuration spécifique)
- network-config → configuration réseau (profils ou instances)
- Format YAML obligatoire avec pipe |
- Images /cloud uniquement (ex: images:ubuntu/24.04/cloud)
- Chaque type de config doit être spécifié séparément (vendor-data, user-data, network-config)

Commandes profils de base

```
incus profile list           # Lister les profils
incus profile show default  # Voir la config du profil
incus profile create mon-profil # Créer un profil
incus profile edit mon-profil  # Éditer un profil
incus profile copy default nouveau # Copier un profil
```

Structure d'un profil Incus

```
config:
  # Configuration infrastructure Incus
  limits.cpu: "2"           # Limiter à 2 CPU
  limits.memory: "4GB"      # Limiter à 4GB RAM

  # Configuration cloud-init (chaque type séparément)
  cloud-init.vendor-data: |  # Configuration système générale
    #cloud-config
    package_update: true
    packages:
      - nginx

  cloud-init.network-config: | # Configuration réseau (séparée)
    version: 2
    ethernet:
      eth0:
        dhcp4: true
```

```

description: "Profil serveur web"
devices:
  eth0:                                # Configuration réseau Incus
    name: eth0
    network: incusbr0
    type: nic
  root:                                # Configuration stockage Incus
    path: /
    pool: default
    type: disk

```

Configuration réseau cloud-init

Réseau statique

```

cloud-init.network-config: |
  version: 2
  ethernets:
    eth0:
      dhcp4: false
      addresses: [192.168.1.100/24]
      gateway4: 192.168.1.1
      nameservers:
        addresses: [8.8.8.8]

```

Réseau DHCP

```

cloud-init.network-config: |
  version: 2
  ethernets:
    eth0:
      dhcp4: true

```

Utilisation pratique

Créer instance avec profils

```

# Méthode recommandée pour cloud-init
incus init images:ubuntu/24.04/cloud test --profile default --profile mon-profil
incus start test

```

```

# Vérifier l'application
incus config show test --expanded

```

Configuration directe sur instance

```

# Méthode 1 : Configuration directe en ligne
incus config set test cloud-init.user-data - << EOF
#cloud-config
hostname: serveur-test
packages:
  - htop
EOF

```

```

# Méthode 2 : Depuis un fichier YAML
incus config set test cloud-init.user-data "$(cat user-data.yaml)"
incus config set test cloud-init.network-config "$(cat network-config.yaml)"

```

```

# Méthode 3 : Application d'un profil depuis fichier
incus profile edit mon-profil < profil-config.yaml

```

Vérification et dépannage

```
# Dans l'instance
incus exec test -- bash
cloud-init status          # État de cloud-init
cloud-init status --wait   # Attendre la fin
cloud-init status --long   # Détails complets

# Logs de débogage
cat /var/log/cloud-init.log
cat /var/log/cloud-init-output.log
```

Exemples

Exemple 1 : Profil infrastructure uniquement

```
incus profile create serveur-2cpu
incus profile edit serveur-2cpu
config:
  limits.cpu: "2"          # Configuration Incus
  limits.memory: "4GB"     # Configuration Incus
devices:
  eth0:
    network: incusbr0      # Configuration réseau Incus
    type: nic
# Pas de cloud-init ici
```

Exemple 2 : Profil avec cloud-init inclus

```
incus profile create web-complet
config:
  limits.cpu: "2"          # Infrastructure Incus
  cloud-init.vendor-data: | # Configuration système
    #cloud-config
  packages:
    - apache2
```

Exemple 3 : Instance avec cloud-init direct

```
# Utilise juste le profil infrastructure
incus init images:ubuntu/24.04/cloud dev --profile default

# Ajoute cloud-init directement sur l'instance
incus config set dev cloud-init.user-data - << EOF
#cloud-config
users:
  - name: developer
    groups: sudo
EOF
```

Exemple 4 : Utilisation de fichiers YAML

```
# Créer les fichiers de configuration
cat > user-data.yaml << EOF
#cloud-config
hostname: serveur-web
users:
  - name: webadmin
    groups: sudo
    shell: /bin/bash
packages:
  - nginx
  - certbot
runcmd:
```

```
- systemctl enable nginx
- systemctl start nginx
EOF
```

```
cat > network-config.yaml << EOF
version: 2
ethernets:
  eth0:
    dhcp4: false
    addresses: [192.168.1.50/24]
    gateway4: 192.168.1.1
    nameservers:
      addresses: [8.8.8.8, 1.1.1.1]
EOF
```

```
cat > profil-web.yaml << EOF
config:
  limits.cpu: "2"
  limits.memory: "4GB"
  cloud-init.vendor-data: |
    #cloud-config
    package_update: true
    package_upgrade: true
    timezone: Europe/Paris
description: "Profil serveur web avec limites"
devices:
  eth0:
    name: eth0
    network: incusbr0
    type: nic
  root:
    path: /
    pool: default
    type: disk
EOF
```

```
# Appliquer les configurations
incus profile create web-server
incus profile edit web-server < profil-web.yaml
```

```
incus init images:ubuntu/24.04/cloud monsite --profile default --profile web-server
```

```
# Deux syntaxes possibles pour les fichiers :
incus config set monsite cloud-init.user-data "$(cat user-data.yaml)"
# Substitution
```

```
incus config set monsite cloud-init.network-config - < network-config.yaml
# Redirection
```

```
incus start monsite
```

Points d'attention

✅ À faire :

- Utiliser incus init puis incus start pour cloud-init
- Respecter l'indentation YAML
- Tester avec des instances temporaires
- Utiliser des fichiers YAML pour configurations complexes

- Valider les fichiers : `cloud-init schema --config-file fichier.yaml`

✗ À éviter :

- Oublier `#cloud-config` en première ligne
- Configurer cloud-init après le premier boot
- Mélanger `vendor-data` et `user-data` avec mêmes clés
- Utiliser des caractères spéciaux dans les noms de fichiers