📘Guide de Déploiement — Platform Auth (Backend Only)

Objectif : Déployer une plateforme d'authentification basée sur **SuperTokens**, **PostgreSQL** et **Backend API** sous Kubernetes.
Particularité : Le **Frontend est désactivé**, l'Ingress redirige directement vers le Backend.
Date : 08 Décembre 2025

---

1. Installation des Outils (Serveur vierge)

Avant de gérer un cluster Kubernetes, vous devez installer **kubectl** et **Helm**.

🔧Étapes d'installation

1.1 Installer les dépendances système

```
sudo apt-get update && sudo apt-get install -y apt-transport-https ca-
certificates curl
```

1.2 Installer kubectl (Kubernetes CLI)

```
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.29/deb/Release.key | sudo
gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://
pkgs.k8s.io/core:/stable:/v1.29/deb/ /' | sudo tee /etc/apt/sources.list.d/
kubernetes.list
sudo apt-get update && sudo apt-get install -y kubectl
```

1.3 Installer Helm

```
curl -fsSL -o get_helm.sh https://raw.githubusercontent.com/helm/helm/main/
scripts/get-helm-3
chmod 700 get_helm.sh
./get_helm.sh
```

1.4 Configurer l'accès au cluster K3s

```
mkdir -p ~/.kube
sudo cp /etc/rancher/k3s/k3s.yaml ~/.kube/config
sudo chown deploy:deploy ~/.kube/config
chmod 600 ~/.kube/config
```

Le fichier ~/.kube/config donne à kubectl & Helm l'accès au cluster.

---

2. Configuration des Secrets (values-prod.yaml)

Les mots de passe, tokens API et configurations critiques sont définis dans le fichier Helm **values-prod.yaml**.

📌Fichier : `/opt/auth_server_tenancy/infrastructure/helm/platform/values-prod.yaml`

🔧Exemple complet (Backend Only)

```
masterDatabase:
  auth:
    password: "PlzOmdEusiPpYHlzoUJitOJi"  # Mot de passe PostgreSQL

supertokens:
  enabled: true
  apiKey: "816MAwinMx42aZdwRz2oLRDvCCAQoEMB"  # API key SuperTokens

backend:
  enabled: true
  replicas: 1
  port: 3001
  env:
    API_DOMAIN: "http://auth.annestay.com"
    # Ajouter ici toutes les variables backend nécessaires

frontend:
  enabled: false   # FRONTEND désactivé

ingress:
  enabled: true
  host: "auth.annestay.com"
  tls:
    enabled: false
```

🕗Important : Le frontend étant désactivé, l'Ingress doit absolument router **/** vers le backend.

---

3. Correction du Template Ingress (Fix du 503)

Par défaut, Helm tente de router `/` vers le Frontend. S'il est désactivé → **Erreur 503**.

🔧Correction à appliquer

📌Fichier : `templates/ingress.yaml`

```
rules:
  - host: {{ .Values.ingress.host }}
    http:
      paths:
        - path: /
          pathType: Prefix
          backend:
            service:
              name: {{ include "platform.backend.serviceName" . }}
              port:
                number: {{ .Values.backend.port }}  # Doit pointer sur le
Backend (3001)
```

> Cette modification force Kubernetes à renvoyer toutes les requêtes root `/` vers l'API Backend Express.

---

4. Déploiement avec Helm

Placez-vous dans le dossier Helm avant de lancer l'installation.

```
cd /opt/auth_server_tenancy/infrastructure/helm/platform
```

4.1 Supprimer l'ancien Ingress (si bug ou conflit)

```
kubectl delete ingress platform-prod-ingress -n prod
```

4.2 Installer / Mettre à jour la plateforme

```
sudo helm upgrade --install prod . -f values-prod.yaml -n prod --kubeconfig /
home/deploy/.kube/config
```

> Helm créera le namespace s'il n'existe pas, générera les services, deployments, secrets, etc.

---

5. Patch PostgreSQL (Fix Auth MD5 / SCRAM)

Par défaut PostgreSQL dans les charts utilise SCRAM-SHA-256, mais SuperTokens tente parfois MD5 → échec.

🔧Étapes du correctif

5.1 Ajouter l'autorisation MD5 dans `pg_hba.conf`

```
kubectl exec -it platform-prod-postgres-0 -n prod -- sh -c "echo 'host all
platform 10.244.0.0/16 md5' >> /var/lib/postgresql/data/pg_hba.conf"
```

5.2 Redémarrer le Pod Postgres

```
kubectl delete pod platform-prod-postgres-0 -n prod
```

5.3 Redémarrer SuperTokens

```
kubectl delete pod $(kubectl get pods -n prod | grep supertokens | awk
'{print $1}') -n prod
```

Cela force SuperTokens à se reconnecter après recharge de la conf PostgreSQL.

---

6. Vérifications Finales

6.1 État des pods

```
kubectl get pods -n prod
```

Résultat attendu

- PostgreSQL → **Running**
- SuperTokens → **Running**
- Backend → **Running**
- Aucun Frontend (normal)

6.2 Vérifier l'accès API

```
curl -v http://auth.annestay.com/
```

Réponse attendue

```
HTTP/1.1 404 Not Found
X-Powered-By: Express
Cannot GET /
```

🎯Cela signifie que **Nginx → Ingress → Backend** fonctionne correctement.

📌Conclusion

Votre stack Platform Auth (Backend Only) est maintenant : - 🔧Déployée sous Kubernetes - 🔧Sécurisée (secrets isolés, ingress corrigé) - 🔧Fonctionnelle sans frontend - 🔧Compatible avec SuperTokens & PostgreSQL

Si tu veux : - Ajouter TLS / HTTPS - Automatiser avec ArgoCD - Créer un monitoring (Grafana / Loki / Prometheus)

Je peux te faire la suite complète 🔐