JEDY STARWARSKUBERNETES

Conteneurisation & Orchestration

Profile Tech

TP DevOps 2025

Contexte & Objectifs Cloud-Native

🚀 Mission

Simuler le cycle de vie complet (code source à production) d'une application moderne en utilisant des principes Cloud-Native pour l'automatisation, la résilience et la scalabilité.

🎓 Compétences Clés

- **Conteneurisation :** Builds optimisés et sécurisés.
- **Orchestration :** Maîtrise de Kubernetes et du Gateway API.
- **Automatisation :** Pipeline CI/CD (GitHub Actions).
- **Backend :** Développement Python avec FastAPI et Gemini AI.

Architecture Microservices

Frontend (Client)

- **Stack :** Astro, Node.js, Tailwind, Three.js.
- **Mode :** SSR (Server-Side Rendering) pour la performance.
- **API Consommée :** Backend API
- **Rôle** : Interface utilisateur interactive et affichage SWAPI/IA.

Backend (API)

- **Stack Moderne** : Python 3.14, FastAPI.
- **Sécurité**: Authentification avec token de session (JWT).
- **Fonctionnalité** : Proxy vers SWAPI et utilisation de Google Gemini pour l'IA.
- **ORM** : SQLModel pour la gestion des utilisateurs.

Réseau & Données

- **Base de données** : PostgreSQL.
- **Conteneurisation** : Docker Multi-stage + séparation DEV & PROD
- **Point d'entrée** : Envoy Gateway (Kubernetes Gateway API).
- **Routing** : Gestion du routage via HTTPRoute.

Orchestration & Réseau K8s

Gateway API (Envoy)

Point d'entrée unique. Routage intelligent des requêtes (`/` vers Front, `/api` vers Back).

Deployments & H.A.

Gestion des Pods pour la haute disponibilité et les Rolling Updates sans interruption.

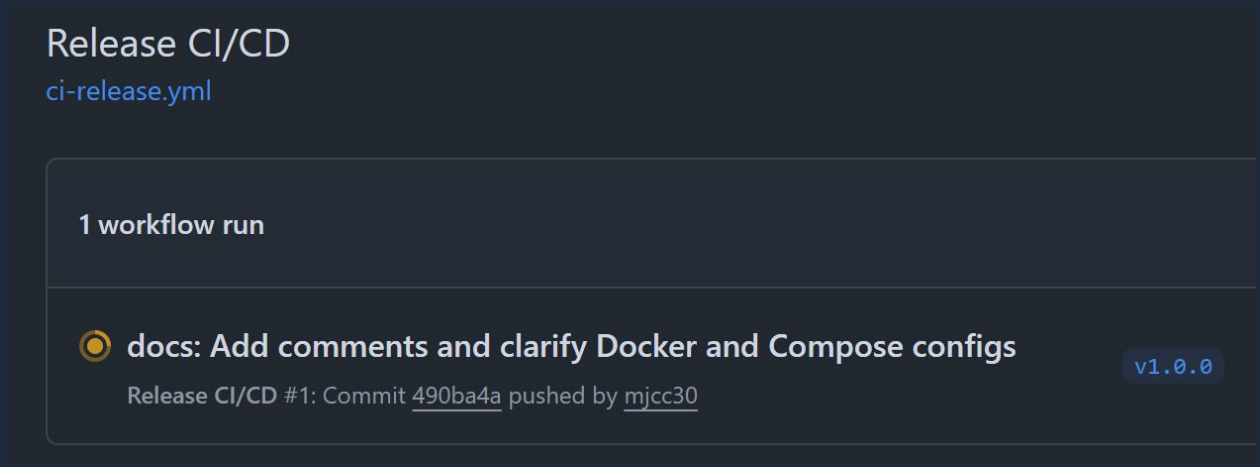
Persistance des Données

Utilisation de PersistentVolumeClaims (PVC) pour sécuriser les données PostgreSQL.

Pipeline CI/CD (GitHub Actions)

- **Build & Lint** : Construction des images Docker et vérification de la qualité.
- **Tests Unitaires** : Exécution des tests `pytest` (avec SQLite in-memory).
- **Push** : Envoi des images conteneurisées vers le registre.
- **Deploy** : Mise à jour automatique des manifestes Kubernetes.

L'automatisation du cycle de vie est gérée par **GitHub Actions**, assurant un déploiement continu et fiable.



DevOps : Sécurité & Efficacité

Bonnes Pratiques de Conteneurisation

- **Multi-stage** : Sépare la construction (dependencies) de l'exécution (runtime slim).
- **Rootless** : Le conteneur final s'exécute sous l'utilisateur 'appuser' (défini dans Dockerfile.prod du backend).
- **Outil Modern** : Utilisation de 'uv' pour une gestion des dépendances Python ultra-rapide et fiable.

```dockerfile
# Build stage
FROM python:3.14-slim AS builder
WORKDIR /app
# Install uv directly from the official image (faster/cleaner than pip)
COPY --from=ghcr.io/astral-sh/uv:latest /uv /uvx /bin/
# Set env vars for uv
ENV UV_COMPILE_BYTECODE=1
ENV UV_LINK_MODE=copy
# Install dependencies into a virtual environment
COPY pyproject.toml .
RUN uv venv /opt/venv && \
    uv pip install --no-cache -r pyproject.toml --python /opt/venv
# Runtime stage
FROM python:3.14-slim AS runner
WORKDIR /app
# Create a non-root user for security
RUN groupadd -r appuser && useradd -r -g appuser appuser
# Copy the virtual environment from the builder stage
COPY --from=builder /opt/venv /opt/venv
# Activate the virtual environment by setting PATH
ENV PATH="/opt/venv/bin:$PATH"
# Copy application code
COPY . .
# Change ownership of the application code to the non-root user
RUN chown -R appuser:appuser /app
# Switch to non-root user
USER appuser
# Expose the port
EXPOSE 4000
# Command to run the application      You, 3 seconds ago • Uncommitted
CMD ["uvicorn", "app.main:app", "--host", "0.0.0.0", "--port", "4000"]
```

Opérations : Auto-Healing & Scaling

Auto-Healing

Le ReplicaSet K8s garantit qu'un Pod défaillant est instantanément remplacé. Démonstration :

```
NAME                                READY   STATUS    RESTARTS        AGE
back-deployment-86d5747c84-r5h6m    1/1     Running   0               20s
front-deployment-7cf9cdbd77-gtfdv   1/1     Running   1 (6m16s ago)   6h3m
front-deployment-7cf9cdbd77-v4vtt   1/1     Running   1 (6m16s ago)   6h3m
front-deployment-7cf9cdbd77-z7bxl   1/1     Running   1 (6m16s ago)   6h3m
postgres-statefulset-0              1/1     Running   1 (6m16s ago)   6h5m
back-deployment-86d5747c84-r5h6m    1/1     Terminating   0             50s
back-deployment-86d5747c84-kkrsq    0/1     Pending       0             0s
back-deployment-86d5747c84-r5h6m    1/1     Terminating   0             50s
back-deployment-86d5747c84-kkrsq    0/1     Pending       0             0s
back-deployment-86d5747c84-kkrsq    0/1     Init:0/1      0             0s
back-deployment-86d5747c84-kkrsq    0/1     PodInitializing   0         1s
back-deployment-86d5747c84-kkrsq    0/1     Running           0         3s
back-deployment-86d5747c84-r5h6m    0/1     Completed         0         53s
back-deployment-86d5747c84-r5h6m    0/1     Completed         0         54s
back-deployment-86d5747c84-r5h6m    0/1     Completed         0         54s
back-deployment-86d5747c84-kkrsq    1/1     Running           0         16s
```

Scaling et Déploiement

Contrôle précis du nombre de réplicas et des mises à jour sans downtime :

```
deployment.apps/back-deployment scaled
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
back-deployment     1/5     5            1           6h9m
back-deployment     2/5     5            2           6h9m
back-deployment     3/5     5            3           6h9m
back-deployment     4/5     5            4           6h9m
back-deployment     5/5     5            5           6h9m
```

Axes d'Amélioration

Pour durcir l'infrastructure et tendre vers une solution de niveau professionnel :

- **Observabilité** : Intégration de **Prometheus** et **Grafana** pour le monitoring des métriques d'application.
- **Sécurité (DevSecOps)** : Scan automatique des images Docker (avec **Trivy** ou autre) dans le pipeline CI/CD.
- **GitOps** : Migration de l'approche CI/CD vers **ArgoCD** pour une gestion du déploiement déclarative.
- **Service Mesh** : Implémentation d'**Istio** pour la sécurité mTLS, le circuit breaking et l'observabilité.

Questions ?

Merci de votre attention.

https://github.com/mjcc30/Jedy-StarWarsKubernetes