

Introduction (Slides 1-5) : Le “Pourquoi”

“Bonjour à tous. Aujourd’hui, je vais vous présenter Orion-Dynamic, un projet d’orchestrateur de ressources à inventaire dynamique.

Le point de départ est un constat simple : dans le monde du cloud actuel, nous avons besoin d’infrastructures jetables, capables de gérer un gros volume de ressources hétérogènes. La problématique était de créer une plateforme capable de gérer automatiquement ces ressources sans intervention humaine constante.“

Le Concept (Slides 15-18) : La Solution

“Orion-Dynamic simule une plateforme PaaS (Platform as a Service). Le principe est celui d’un workflow locatif : un utilisateur demande une ressource, le système la lui provisionne pour une durée déterminée (un bail), puis la nettoie automatiquement à l’expiration. C’est ce qu’on appelle l’automatisation NoOps.”

Architecture & Philosophie (Slides 19-22) : Le “Cœur”

“Techniquement, j’ai opté pour une séparation stricte entre le Control Plane (l’intelligence) et le Data Plane (les ressources).

La grande particularité ici est le mode ‘Push’. Contrairement aux outils classiques qui scannent le réseau, ici, chaque ressource possède un agent léger qui vient s’enregistrer de lui-même auprès de l’API au démarrage. Cela permet de traverser les Firewalls sans configuration complexe et d’ajouter de la puissance de calcul instantanément.“

Choix Techniques Clés (Slides 23-26) : La Maîtrise

“Pour garantir la fiabilité, j’ai fait trois choix structurants :

MariaDB pour la cohérence : Dans un système de location, on ne peut pas se permettre de louer deux fois le même serveur. Les transactions ACID garantissent cette intégrité.

SKIP LOCKED pour la performance : Le Scheduler utilise ce mécanisme SQL pour que plusieurs instances puissent travailler en parallèle sans se bloquer mutuellement, assurant ainsi la scalabilité du système.

Ansible comme moteur de sécurité : Ansible ne me sert pas à déployer, mais à isoler les clients en créant des utilisateurs UNIX éphémères. À la fin du bail, tout est supprimé : aucune donnée résiduelle ne subsiste.“

Résilience & Démo (Slides 27-31) : Le “Self-Healing”

“Le système est conçu pour être autonome. Le Scheduler effectue des Health Checks constants via SSH. Si un worker tombe en panne, le système déclenche une migration à chaud : l’utilisateur est déplacé vers un nœud sain de manière transparente.

Dans ma démonstration (full_demo.sh), je prouve ce concept en simulant une panne réelle, où l’on voit le système réagir et sécuriser l’infrastructure immédiatement.“

Bilan & Ouverture (Slides 32-39) : L’Avenir

“Pour conclure, Orion-Dynamic est un projet qui atteint une couverture de tests de 82%, garantissant une grande qualité logicielle.

C’est une base solide que j’aimerais faire évoluer vers du vrai Infrastructure as Code avec Terraform et en ajoutant une couche d’observabilité complète pour surveiller la santé du cluster en temps réel.“