



# Serveur WoW via Docker – Partie 1/2

# Objectifs Partie 1

- Installer Docker & docker-compose sur Debian/Ubuntu
- Déployer un serveur AzerothCore via container
- Configurer le port 8085 pour connexion client

# Installation des prérequis Docker

- `sudo apt install docker.io docker-compose`
- `sudo usermod -aG docker $USER`
- `sudo systemctl enable docker`

# Récupération du projet AzerothCore

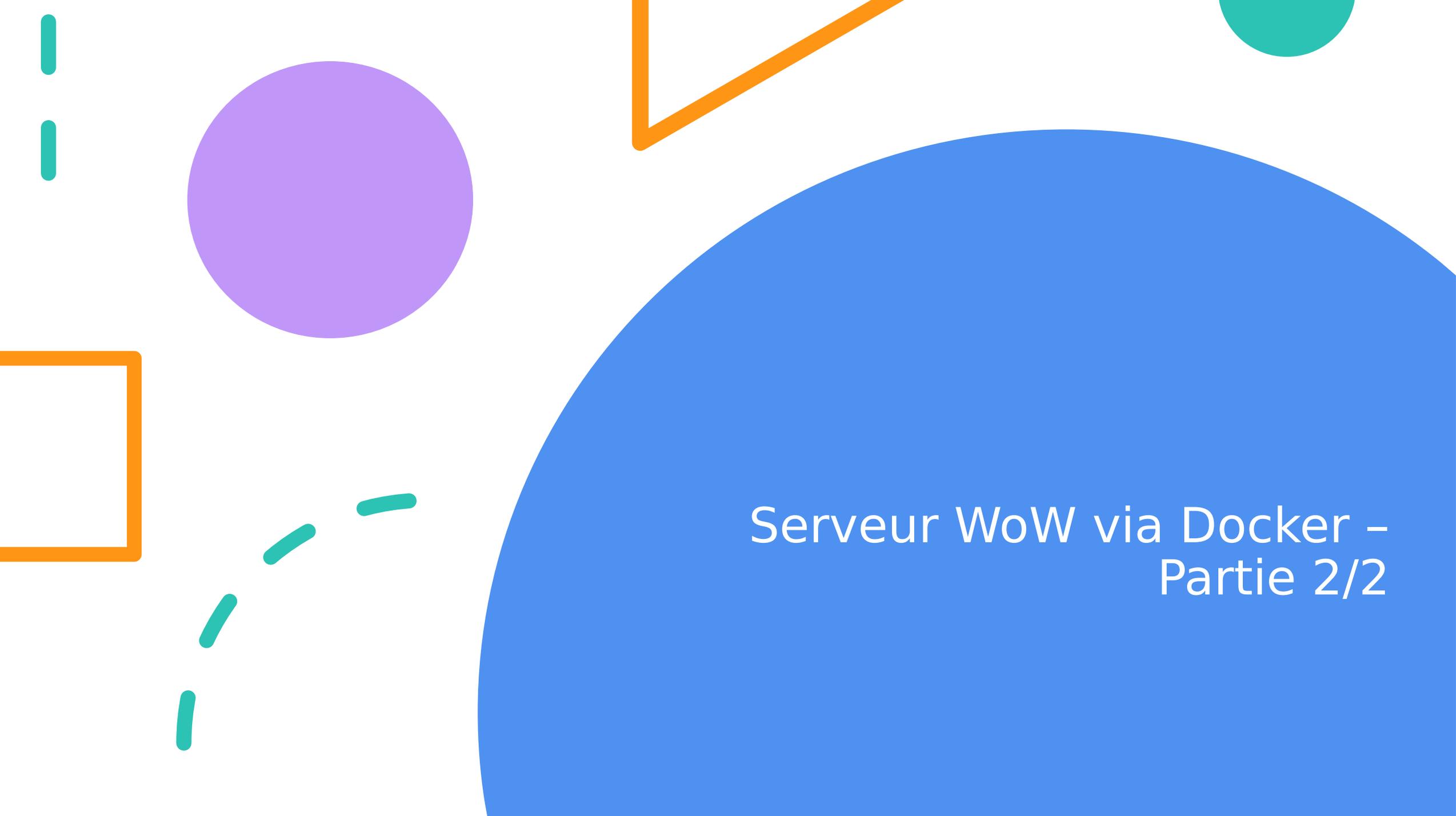
- `git clone https://github.com/azerothcore/docker.git`
- `cd docker`
- Modifier `.env` si besoin (ports, DB, etc.)

# Lancement du serveur WoW

- `docker-compose build`
- `docker-compose up -d`
- `docker ps` pour vérifier que tout tourne

# TP 1 - Déploiement du serveur AzerothCore

- Installer Docker, cloner le projet, lancer le serveur
- Vérifier le mapping des ports (8085/3306)
- Connecter un client WoW en local



# Serveur WoW via Docker – Partie 2/2

# Objectifs Partie 2

- Créer un compte joueur dans la base de données
- Configurer realmlist & compte GM
- Gérer la persistance via volumes Docker

# Création de compte WoW privé

- `docker exec -it authserver acore account create nom mdp`
- `acore account set gmlevel nom 3 -1`
- `realmist = adresse IP du serveur`

# TP 2 - Gestion des comptes et realmlist

- Créer plusieurs comptes (joueur / GM)
- Configurer realmlist sur le client WoW
- Connexion avec plusieurs clients test

# Persistance & sauvegarde

- Volumes Docker : `mysql:/var/lib/mysql`
- Sauvegarder les fichiers `.conf` et les DB
- Restaurer un container WoW complet

# Résumé du cours WoW + Docker

- docker-compose pour démarrer le serveur AzerothCore
- Création de comptes via CLI Docker
- Connexion client local via port 8085