FR

Guide Pratique

Docker sur Ubuntu 2026

Installation complete et bonnes pratiques

Mars 2026

Sommaire

1. Installation Docker Engine

```
# Supprimer anciennes versions
sudo apt remove -y docker docker-engine docker.io 2>/dev/null || true

# Prerequis
sudo apt update
sudo apt install -y ca-certificates curl gnupg

# Cle GPG Docker
sudo install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg \
  | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg

# Depot officiel
echo "deb [arch=$(dpkg --print-architecture) \
  signed-by=/etc/apt/keyrings/docker.gpg] \
  https://download.docker.com/linux/ubuntu \
  $(. /etc/os-release && echo $VERSION_CODENAME) stable" \
  | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

# Installation
sudo apt update
sudo apt install -y docker-ce docker-ce-cli containerd.io \
                    docker-buildx-plugin docker-compose-plugin
docker --version && docker compose version
```

2. Post-installation

```
# Groupe docker (eviter sudo)
sudo usermod -aG docker $USER
newgrp docker

# Activer au demarrage
sudo systemctl enable docker
sudo systemctl start docker
docker ps   # Test sans sudo
```

3. Commandes essentielles

```
docker ps                          # Conteneurs actifs
docker ps -a                       # Tous les conteneurs
docker run -d -p 8080:80 nginx     # Demarrer nginx
docker stop / start / restart NOM  # Gestion
docker rm NOM                      # Supprimer
docker exec -it NOM bash           # Shell interactif
docker logs -f NOM                 # Logs en direct
docker stats                       # CPU/RAM temps reel
docker images                      # Images locales
docker pull IMAGE:TAG              # Telecharger image
docker image prune -f              # Nettoyer
```

4. docker-compose.yml

```
version: '3.8'

services:
  webapp:
    image: nginx:alpine
    container_name: mon-site
    restart: unless-stopped
    ports:
      - "8080:80"
    volumes:
      - ./html:/usr/share/nginx/html:ro
    networks:
      - app-network

  database:
    image: postgres:16-alpine
    environment:
      POSTGRES_PASSWORD: ${DB_PASSWORD}
    volumes:
      - db-data:/var/lib/postgresql/data

volumes:
  db-data:
networks:
  app-network:
```

5. Variables .env

```
# Fichier .env (NE PAS commiter dans git)
DB_PASSWORD=MonMotDePasseSecurise
APP_KEY=cle-aleatoire

# Ajouter .env au .gitignore
echo ".env" >> .gitignore
```

6. Volumes

```
# Bind mount (dossier hote)
- ./data:/var/data

# Named volume (gere par Docker)
volumes:
  mon-volume:

# Backup d'un volume
docker run --rm \
  -v mon-volume:/data \
  -v $(pwd):/backup \
  alpine tar czf /backup/backup-$(date +%Y%m%d).tar.gz /data
```

7. Networks

```
# Les services du meme compose se voient
# par leur nom de service (ex: http://database:5432)

docker network ls
docker network create mon-reseau
docker network inspect mon-reseau
```

8. Mise a jour

```
cd /chemin/vers/app
docker compose pull        # Nouvelles images
docker compose up -d       # Redeployer
docker compose ps          # Verifier
docker image prune -f      # Nettoyer
```

Article complet : botum.ca/installer-docker-ubuntu/

Site : www.botum.ca -

- Canada