Installation



Fichier docker-compose

Dans cette procédure, nous allons utiliser **Docker**. Une documentation préalable à ce sujet est disponible ici.
Commençons par créer un fichier `docker-compose.yaml`, adapté à nos besoins :

```
services:
  # Service hbbs : serveur de rendez-vous (ID Server)
  hbbs:
    container_name: hbbs  # Nom explicite du conteneur
    image: rustdesk/rustdesk-server:latest  # Image officielle du serveur RustDesk
    command: hbbs  # Lance le composant "hbbs" (Heartbeat Broker Server)
    volumes:
      - ./data:/root  # Monte le dossier local "data" dans /root du conteneur pour conserver les clés et paramètres
    network_mode: "host"  # Utilise directement le réseau de l'hôte (nécessaire pour les ports UDP/TCP fixes)
```

```
    depends_on:
      - hbbr  # Démarre uniquement si le serveur relais (hbbr) est déjà en cours d'exécution
    restart: unless-stopped  # Redémarrage automatique sauf si le conteneur est arrêté manuellement

  # Service hbbr : serveur de relais (Relay Server)
  hbbr:
    container_name: hbbr  # Nom du conteneur
    image: rustdesk/rustdesk-server:latest  # Utilise la même image que hbbs
    command: hbbr  # Lance le composant "hbbr" (Relay Server)
    volumes:
      - ./data:/root  # Partage le même dossier que hbbs (utile pour les certificats et clés)
    network_mode: "host"  # Nécessaire pour exposer les ports réseau sans restriction
    restart: unless-stopped  # Redémarrage automatique si le conteneur s'arrête de manière imprévue
```

Dans ce fichier, nous n'avons spécifié **aucun port explicitement**, car nous utilisons le mode réseau `host`. En revanche, il est important de connaître les ports utilisés par RustDesk, notamment si un **pare-feu** est en place ou si le serveur doit être **accessible depuis l'extérieur**.

Ports utilisés par RustDesk

Port	Protocole	Description
21115	TCP	Test du type de NAT
21116	UDP	Enregistrement de l'ID et service Heartbeat
21116	TCP	Connexions entrantes
21117	TCP	Service de relais
21118	TCP	Client Web (non obligatoire)
21119	TCP	Client Web (non obligatoire)

Démarrage

Une fois le fichier `docker-compose.yaml` créé, nous pouvons **lancer RustDesk** avec la commande suivante :

```
docker compose up -d
```

Récupération de la clé publique

Il reste une dernière étape essentielle : récupérer la **clé publique** générée par le serveur. Celle-ci est nécessaire pour que les clients puissent établir une connexion chiffrée avec le serveur :

```
docker cp <container-id>:/id_ed25519.pub <destination>
```

> Remarque : le chemin ou le nom du conteneur peut varier si vous avez modifié les paramètres de base du fichier docker-compose.

Revision #1
Created 2025-10-29 12:46:25 UTC
Updated 2025-10-29 12:46:25 UTC