



## Guide d'installation d'un serveur RustDesk Remote Desktop sur AlmaLinux 9

Équipe de rédaction et d'approbation		
<b>NGANSOP NJANOU ULRICH SOSTAIRE</b>	Junior Network Administrator <b>EVOLV IZSOFTWARES GROUP Ltd</b>	Date: 14 janvier 2025
<b>GILDAS FOTSO TABAFO</b>	Junior Network Administrator <b>EVOLV IZSOFTWARES GROUP Ltd</b>	Date: 14 janvier 2025
<b>Beryl Ngonga</b>	Encadreur professionnelle - <b>EVOLV IZSOFTWARES GROUP Ltd</b>	Date de fin : _
Sommaire des révisions		
Historique de Révision	Description générale	Date approuvée
<b>0.0.3</b>	Document, version 3	

### But

Le but de cette procédure est de décrire la stratégie de base de l'installation et de la configuration techniques pratiques pour installer et configurer un serveur RustDesk Remote Desktop Server sécurisé et privé.

### **Application**

Cette procédure est destinée au département des technologies de l'information (TI) d'**IZSoftware**. La date de mise en place de cette procédure est la date d'approbation.

## Installation d'un Serveur RustDesk Remote Desktop Server sur AlmaLinux 9

RustDesk est un outil open-source d'accès à distance qui permet de contrôler et d'accéder en toute sécurité à des machines, même à distance. Contrairement à de nombreuses solutions cloud, RustDesk offre la possibilité d'un

hébergement autonome, garantissant ainsi un contrôle accru des données et une sécurité renforcée.

En plus de sa gratuité, RustDesk est compatible avec plusieurs plateformes, notamment Windows, macOS, Linux, iOS et Android. Il combine un contrôle à distance performant avec un chiffrement de bout en bout, le tout dans une interface simple à configurer. Ses fonctionnalités comme le transfert de fichiers et son ergonomie le rendent idéal aussi bien pour un usage personnel que professionnel.

L'objectif de ce rapport est de fournir un guide détaillé pour installer un serveur RustDesk sur une infrastructure dédiée, afin de garantir une connexion stable et sécurisée entre clients.

## **1- Prérequis nécessaires**

### **a) Matériels (Minimum) :**

- Un serveur physique ou virtuel avec :
  - Processeur : Dual-core minimum.
  - RAM : 4 Go ou plus.
  - Disque : 20 Go d'espace libre.

### **b) Logiciels :**

- Système d'exploitation : Linux ( AlmaLinux 9 ou plus récent recommandé).
- Docker et Docker Compose installés.
- Un domaine ou une IP publique configurée pour accéder au serveur.
- Accès root ou sudo à la machine.

### **c) Étapes d'installation du Serveur**

- Avant de commencer l'installation, prenez le temps de vérifier que votre serveur est à jour.

```
dnf update -y
```

```
dnf install -y dnf-utils nano
```

- Ensuite, assurez-vous que Docker est bien installé (Docker est nécessaire pour déployer rapidement le serveur RustDesk). Si ce n'est pas encore le cas, vous pouvez facilement l'installer en suivant les commandes ci-dessous.

```
yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
```

```
dnf install -y docker-ce docker-ce-cli containerd.io
```

```
systemctl start docker
```

```
systemctl enable docker
```

- Par la suite, procédez à l'installation de Docker Compose.

```
dnf install curl -y
```

```
curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

```
chmod +x /usr/local/bin/docker-compose
```

```
ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose
```

```
docker-compose --version
```

## **2- Procédez à l'installation de RustDesk**

### **a) Créez un répertoire dédié à RustDesk**

Créez un répertoire afin d'y stocker les fichiers Docker de RustDesk. Ce dossier servira à conserver les fichiers de configuration ainsi que toutes les données essentielles pour le serveur RustDesk.

```
mkdir -p /opt/rustdesk
```

Entrez dans le répertoire que vous venez de créer.

```
cd /opt/rustdesk
```

### b) Créer un fichier de configuration Docker Compose

Dans le répertoire, créez un fichier Docker Compose nommé **rustdesk.yml**. Ce fichier servira à définir les services indispensables au bon fonctionnement du serveur RustDesk, et sera situé à l'adresse `/izsoftware/rustdesk`.

```
nano rustdesk.yml
```

Insérez les configurations suivantes dans le fichier **rustdesk.yml**. Elles permettent de mettre en place deux services essentiels pour le fonctionnement du serveur RustDesk : **hbbs** et **hbbr**.

```
1 version: '3'
2
3 networks:
4
5   rustdesk-net:
6
7     external: false
8
9 services:
10
11   hbbs:
12
13     container_name: hbbs
14
15     ports:
16
17       - 21115:21115
18
19       - 21116:21116
20
21       - 21116:21116/udp
22
23       - 21118:21118
24
25     image: rustdesk/rustdesk-server:latest
26
27     command: hbbs -r 127.0.0.1:21117 -k _
28
29     volumes:
30
31       - ./hbbs:/root
32
33     networks:
34
35       - rustdesk-net
36
37     depends_on:
38
39       - hbbr
40
41     restart: unless-stopped
42
```

```

43   hbbr:
44
45     container_name: hbbr
46
47     ports:
48
49       - 21117:21117
50
51       - 21119:21119
52
53     image: rustdesk/rustdesk-server:latest
54
55     command: hbbr -k _
56
57     volumes:
58
59       - ./hbbr:/root
60
61     networks:
62
63       - rustdesk-net
64
65     restart: unless-stopped

```

Sauvegardez les modifications et fermez le fichier. (Ctrl+x )

### c) Lancez les conteneurs du serveur RustDesk

Utilisez Docker Compose pour lancer les conteneurs du serveur RustDesk.

**L'option -d** permet de les exécuter en mode détaché, ce qui signifie qu'ils fonctionneront en arrière-plan.

```
docker-compose -f rustdesk.yml up -d
```

Vérifiez que les conteneurs RustDesk **hbbs** et **hbbr** sont bien en fonctionnement. La commande suivante affiche la liste de tous les conteneurs Docker actuellement en cours d'exécution.

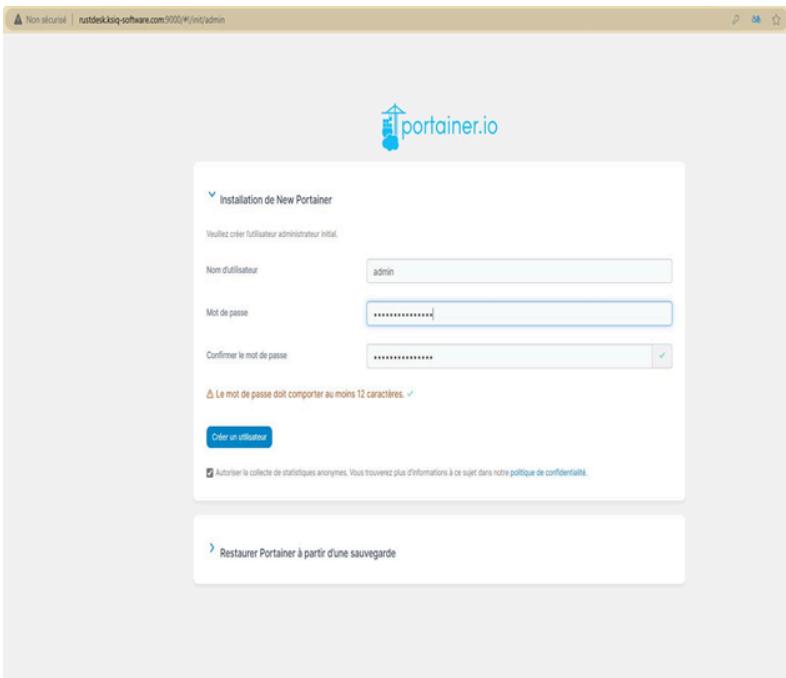
```
docker ps
```

**Pour utiliser l'interface graphique de docker , installer l'utilitaire suivant (Portainer) :**

```
docker volume create portainer_data && docker run -d -p 9000:9000 -p 8000:8000 --name=portainer --restart=always -v /var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data portainer/portainer-ce
```

**i** Remarque : Pour accéder, utilisez l'URL suivante : <http://remote.ksiq-software.com:9000>

**i** Remarque : remplacez le [remote.ksiq-software.com](http://remote.ksiq-software.com) par votre nom de domaine réel



Première connexion à l'interface graphique de docker ; Création du l'utilisateur admin et du mot de passe

Name	State	Quick Actions	Stack	Image	Created	IP Address	Published
hbr	running			rustdesk/rustdesk-server:latest	2024-12-29 11:53:19	172.18.0.3	171.17.2.21
hobs	running			rustdesk/rustdesk-server:latest	2024-12-29 11:53:20	172.18.0.2	171.18.2.11
portainer	running			portainer/portainer-ce	2024-12-29 12:36:30	172.17.0.2	172.17.0.2 8000:8000

L'interface graphique d'administration

### 3- Protégez le serveur en utilisant Nginx ☺

#### a) Installez Nginx

Si Nginx n'est pas encore installé sur votre serveur, vous pouvez l'installer en utilisant la commande suivante. Nginx servira de proxy inverse pour rediriger le trafic vers le serveur RustDesk.

```
dnf install -y nginx
```

#### b) Modifier le fichier nano /etc/nginx/nginx.conf

```

GNU nano 5.6.1          /etc/nginx/nginx.conf
keepalive_timeout 65;
types_hash_max_size 4096;

include /etc/nginx/mime.types;
default_type application/octet-stream;

# Load modular configuration files from the /etc/nginx/conf.d directory.
# See http://nginx.org/en/docs/ngx_core_module.html#include
# for more information.
include /etc/nginx/conf.d/*.conf;

server {
    listen      8080;
    listen      [::]:8080;
    server_name remote.ksiq-software.com;
    root        /usr/share/nginx/html;

    # Load configuration files for the default server block.
    include /etc/nginx/default.d/*.conf;
}

```

^G Help ^C Write Out ^W Where Is ^K Cut ^T Execute ^C Location  
^X Exit ^R Read File ^M Replace ^U Paste ^J Justify ^L Go To Line

Dans le fichier /etc/nginx/nginx.conf modifier la section Serveur

**i Remarque : remplacez le `remote.ksiq-software.com` par votre nom de domaine réel ↗**

### c) Configurez le pare-feu (Firewall)

- Désactiver selinux

`nano /etc/selinux/config`

```

SELINUX=permissive
# SELINUXTYPE= can take one of these three values:
#   targeted - Targeted processes are protected,
#   minimum - Modification of targeted policy. Only selected processes are protected
#   mls - Multi Level Security protection

```

Remplacer *enforcing* par *permissive*

- Vérifier les zones associé à l'interface réseau

`firewall-cmd --get-active-zones`

- Associer l'interface vitre interface à la zone DMZ

`sudo firewall-cmd --zone=dmz --add-interface=ens160 --permanent`

- Ouverture des ports nécessaire aux services Docker et RustDesk.

`sudo firewall-cmd --permanent --zone=dmz --add-port=8080/tcp`

`sudo firewall-cmd --permanent --zone=dmz --add-port=80/tcp`

`sudo firewall-cmd --permanent --zone=dmz --add-port=8443/tcp`

`sudo firewall-cmd --permanent --zone=dmz --add-port=443/tcp`

`sudo firewall-cmd --permanent --zone=dmz --add-port=21115/tcp`

`sudo firewall-cmd --permanent --zone=dmz --add-port=21116/tcp`

`sudo firewall-cmd --permanent --zone=dmz --add-port=21116/udp`

`sudo firewall-cmd --permanent --zone=dmz --add-port=21117/tcp`

`sudo firewall-cmd --permanent --zone=dmz --add-port=8000/tcp`

`sudo firewall-cmd --permanent --zone=dmz --add-port=9000/tcp`

`firewall-cmd --reload`

### d) Créez un fichier de configuration Nginx pour RustDesk

Créez un nouveau fichier de configuration Nginx dédié à RustDesk. Ce fichier permettra à Nginx de rediriger les requêtes de votre sous-domaine vers le serveur RustDesk.

```
nano /etc/nginx/conf.d/remote.ksiq-software.com
```

Ajoutez la configuration suivante dans le fichier. Remplacez remote.ksiq-software.com par votre véritable sous-domaine.

```
1 server {  
2  
3     listen 8080;  
4  
5     listen [::]:8080;  
6  
7     server_name remote.ksiq-software.com;  
8  
9     location / {  
10  
11         proxy_pass http://192.168.2.6:21117;  
12  
13         proxy_set_header Host $host;  
14  
15         proxy_set_header X-Real-IP $remote_addr;  
16  
17     }  
18  
19 }
```

**i Remarque :** remplacez le `remote.ksiq-software.com` par votre nom de domaine réel / Spécifier l'adresse ip de votre serveur Rustdesk

**Enregistrez et fermez le fichier.**

## 4- Affichez les clés de chiffrement du serveur RustDesk ↗

```
cat /opt/rustdesk/hbbs/id_ed25519.pub
```

**i** Le résultat devrait ressembler à ceci :

```
13toHGQLmzIPonp7lMphnhbAui9VfLYLCi7iD03ZZak=
```

## 5- Sécuriser Nginx avec SSL a l'aide de Certbot ↗

Pour garantir une communication sécurisée avec votre serveur RustDesk, vous devez sécuriser votre proxy inverse Nginx en activant SSL. Cela peut être facilement réalisé à l'aide de Certbot, qui délivre des certificats SSL gratuits via Let's Encrypt.

### a) Activez le référentiel EPEL

Commencez par activer le référentiel EPEL afin de pouvoir installer Certbot.

```
dnf install -y epel-release
```

### b) Installez Certbot ainsi que le plugin Nginx. ↗

```
dnf install -y certbot python3-certbot-nginx
```

### c) Obtenez et installez un certificat SSL. ↗

Utilisez Certbot pour obtenir et installer un certificat SSL pour votre domaine.

- i** Remplacez **remote.ksiq-software.com** par votre véritable sous-domaine.

```
certbot --nginx -d remote.ksiq-software.com
```

- i** Suivez les étapes pour compléter l'installation. Vous serez invité à fournir une adresse e-mail pour recevoir les notifications importantes de renouvellement et de sécurité, puis à accepter les conditions d'utilisation.

#### d) Vérifier la configuration SSL ↗

Une fois l'installation de Certbot terminée, vérifiez que Nginx est configuré correctement et que la connexion SSL fonctionne comme prévu.

```
nginx -t
```

```
systemctl reload nginx
```

- i** Un certificat SSL est crucial pour sécuriser les connexions entre les clients et le serveur RustDesk. Il chiffre les données échangées, protégeant ainsi les informations sensibles contre les attaques. Cela assure la confidentialité et la sécurité des utilisateurs, en particulier lors de l'accès à des systèmes distants via des réseaux non sécurisés.

Nous recommandons un certificat SSL payant pour RustDesk, car il offre des garanties supplémentaires, telles qu'une validation d'identité renforcée et un support technique dédié. Cela assure une sécurité maximale et renforce la confiance des utilisateurs, ce qui est essentiel pour un service de bureau à distance comme RustDesk.

Voici quelques fournisseurs populaires de certificats SSL payants, offrant des options adaptées à différents besoins de sécurité pour des services comme RustDesk :

Fournisseurs de certificats SSL ↗	Option de certificats ↗	Site Web ↗
<b>GoDaddy ↗</b>	<b>Fournisseur bien connu pour ses certificats SSL abordables et une bonne assistance technique. Offre des certificats SSL DV, OV et EV. ↗</b>	<a href="https://www.godaddy.com/en-ca">https://www.godaddy.com/en-ca ↗</a>
<b>RapidSSL ↗</b>	<b>certificats SSL ↗</b>	<a href="https://www.rapidssl.com/">https://www.rapidssl.com/ ↗</a>
<b>GlobalSign ↗</b>	<b>Offre une large gamme de produits SSL, y compris des certificats EV et des certificats Wildcard. ↗</b>	<a href="https://www.globalsign.com/en">https://www.globalsign.com/en ↗</a>

#### **i** Remarque 1 : ↗

Une fois la configuration du serveur RustDesk terminée, il est essentiel de garantir que les connexions entrantes puissent atteindre correctement votre serveur, en particulier si celui-ci est derrière un routeur domestique ou un pare-feu d'entreprise. Pour cela, il est nécessaire de rediriger certains ports spécifiques vers votre serveur Rustdesk . L'ouverture des ports permettra aux clients RustDesk de se connecter à votre serveur à distance, tout en maintenant la sécurité de votre infrastructure.

## **❶ Remarque 1 :Ouverture des Ports pour un serveur RustDesk Privé:** ☀

Ports à ouvrir :

1. **Port TCP 21115** : Utilisé par le service **hbbs** pour la communication principale.
2. **Port TCP 21116** : Utilisé par **hbbs** pour la gestion des connexions avec les clients.
3. **Port UDP 21116** : Transport des données via le protocole UDP pour une communication plus rapide (mais moins fiable que TCP).
4. **Port TCP 21118** : Utilisé pour la gestion supplémentaire des connexions.
5. **Port TCP 21117** : Utilisé par **hbbr** pour la communication avec **hbbs**.
6. **Port TCP 21119** : Permet au **hbbr** d'assurer la connectivité des clients via le relais.

Pour garantir une communication fluide et sécurisée avec votre serveur RustDesk privé, vous devez ouvrir les ports suivants sur votre routeur ou pare-feu :

- **TCP 21115, 21116, 21118, 21117, 21119**
- **UDP 21116** (si vous utilisez le transport de données via UDP).

Cela permettra à votre serveur privé de fonctionner correctement et d'assurer une expérience utilisateur optimale pour les connexions client-serveur.

Redirection de port							
Dans certains cas, il est nécessaire d'ouvrir les ports pour permettre la transmission du trafic dans le réseau local.							
Nom	État	Protocole	Port interne	Port externe	Adresse IP locale / Nom de l'appareil	Créé par	Action
rustdesk-21115	<input checked="" type="checkbox"/>	Les deux	21115	21115	Appareil inconnu - 54	Utilisateur	
rustdesk-21116	<input checked="" type="checkbox"/>	Les deux	21116	21116	Appareil inconnu - 54	Utilisateur	
rustdesk-21117	<input checked="" type="checkbox"/>	Les deux	21117	21117	Appareil inconnu - 54	Utilisateur	
rustdesk-8080	<input checked="" type="checkbox"/>	Les deux	80	80	Appareil inconnu - 54	Utilisateur	
portail docker	<input checked="" type="checkbox"/>	Les deux	9000	9000	Appareil inconnu - 54	Utilisateur	
rustdesk-443	<input checked="" type="checkbox"/>	Les deux	443	443	Appareil inconnu - 54	Utilisateur	

Redirection de port vers mon serveur RustDesk

## **6- Configuration du client RustDesk (à l'attention des utilisateurs finaux)** ☀

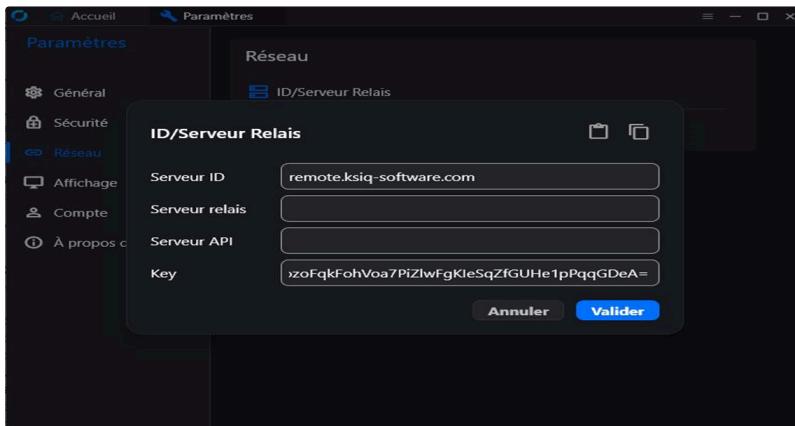
1. Téléchargez et installez le client RustDesk sur votre ordinateur.

2. Ouvrez et configurez le client RustDesk :

- Allez dans les paramètres, puis sélectionnez « Réseau » et cliquez sur « Déverrouiller les configurations réseau ».
- Ensuite, dans la section « **ID/Relay Server** » de l'application, dans le champ « ID Serveur », entrez le nom de domaine de votre serveur RustDesk (par exemple : *remote.ksiq-software.com*).

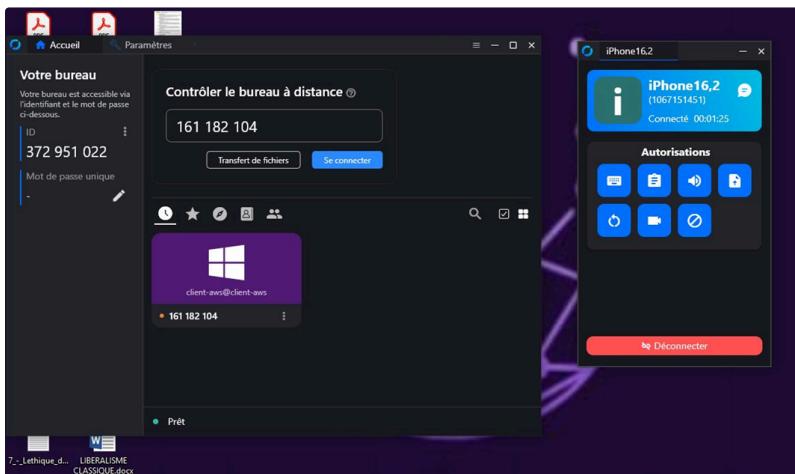
- Dans le champ « **Clé** », saisissez la clé publique plus haut fournie par votre serveur RustDesk (par exemple : `13toHGQLmzIPonp7IMphnhbAui9VflYLCi7iDO3ZZak=`)

3. Appliquez les paramètres : Cliquez sur « **Valider** » pour sauvegarder votre configuration.



4. Établissez une connexion :

- Dans le champ « **Entrer l'ID de l'appareil distant** », entrez l'ID de la machine distante.
- Cliquez sur « **Connacter** » pour établir une connexion sécurisée et stable via votre serveur RustDesk privé.



Contrôle à distance établi depuis un téléphone via notre serveur RustDesk

## 7- Option de Sécurité : Protégez-vous efficacement (à l'attention des utilisateurs finaux) ☺

### A- OPTION ACTIVE

Il existe plusieurs options de sécurité dédiées aux utilisateurs finaux pour garantir une utilisation sécurisée de RustDesk. Il est important de noter que ces options doivent être choisies en fonction des besoins spécifiques de chaque utilisateur et peuvent être combinées si nécessaire.

**a) . Sécurité par mot de passe :** C'est l'une des protections de base. Vous devez définir un mot de passe fort et unique pour protéger l'accès à votre session.

- Pourquoi ?** Si quelqu'un tente d'accéder à votre session sans votre autorisation, il devra fournir le mot de passe que vous avez défini.
- Conseil :** Utilisez un mot de passe comportant au moins 8 caractères, avec des lettres majuscules, minuscules, chiffres et symboles (ex. : P@ssw0rd123).

**b) Authentification à deux facteurs (2FA)** : Cette option ajoute une couche supplémentaire de sécurité. Même si quelqu'un connaît votre mot de passe, il devra aussi entrer un code unique généré sur votre téléphone ou envoyé par e-mail.

- **Pourquoi** ? Cela rend l'accès à votre session presque impossible sans votre téléphone ou e-mail.
- **Conseil** : Activez 2FA et utilisez une application de génération de codes comme **Google Authenticator ou Authy**.

**c). Sécurité "Autoriser l'accès direct par IP"** : Vous pouvez restreindre l'accès à votre session uniquement à des adresses IP spécifiques.

- **Pourquoi** ? Cela empêche quiconque, en dehors des IP autorisées (par exemple, celles de votre bureau ou maison), d'accéder à votre session.
- **Conseil** : Configurez cette option si vous connaissez les adresses IP fiables qui doivent accéder à votre session.

**d) Utiliser une liste blanche IP** : Avec cette option, vous créez une liste des adresses IP autorisées à se connecter à votre session. Si une adresse IP n'est pas sur la liste, elle sera bloquée automatiquement.

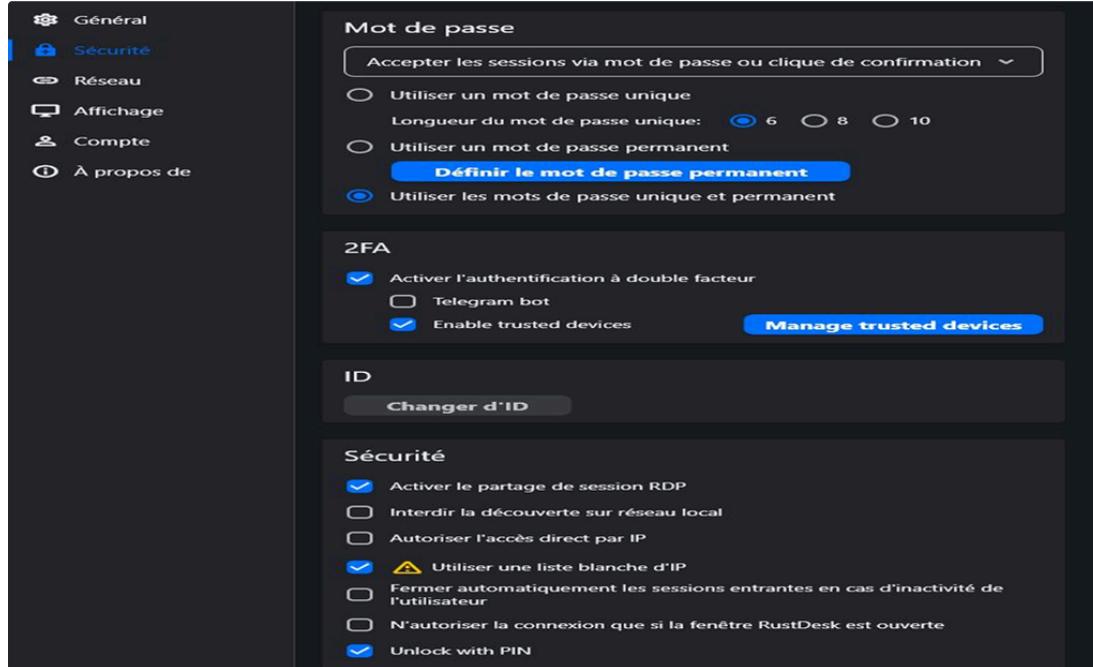
- **Pourquoi** ? Cela ajoute un filtre strict, empêchant les connexions indésirables.
- **Conseil** : Mettez uniquement les adresses IP que vous connaissez et considérez sûres.

**e) Déverrouillage avec un PIN** : Vous pouvez configurer un code PIN court pour déverrouiller rapidement l'accès à votre session RustDesk.

**Pourquoi** ? C'est une mesure pratique pour éviter d'avoir à saisir votre mot de passe complet à chaque fois, tout en gardant un accès sécurisé.

**Conseil** : Utilisez un code PIN facile à mémoriser mais difficile à deviner, comme 1934 au lieu de 1234.

**f) Fermeture automatique des sessions en cas d'inactivité** : Cette fonction déconnecte automatiquement une session si l'utilisateur ne l'utilise pas pendant un certain temps



## B- OPTION PASSIVE

### Les Autorisations dans RustDesk : Gérer et Limiter les Actions des Utilisateurs Distant

RustDesk propose des options d'autorisation qui permettent de contrôler précisément ce que l'utilisateur distant peut faire sur votre appareil. Ces autorisations sont essentielles pour garantir la sécurité et éviter tout abus. Voici un aperçu des principales autorisations et leur rôle :

## **Autorisation pour la prise de contrôle de l'écran**

Cette autorisation permet à l'utilisateur distant d'avoir le contrôle complet de votre écran.

- Pourquoi ? Elle est utile pour des sessions d'assistance où une intervention directe est nécessaire.
- Comment gérer ? Activez cette autorisation uniquement si vous faites confiance à l'utilisateur distant. Sinon, limitez leur accès à un simple affichage de l'écran.

## **Autorisation pour le transfert de fichiers**

Avec cette autorisation, l'utilisateur distant peut envoyer ou récupérer des fichiers de votre appareil.

- Pourquoi ? Cela peut être pratique pour échanger des documents rapidement, mais cela représente aussi un risque si une personne mal intentionnée télécharge des fichiers sensibles.
- Comment gérer ? Désactivez cette autorisation si elle n'est pas nécessaire pendant la session.

## **Autorisation pour le clavier et la souris**

Cette autorisation donne à l'utilisateur distant la possibilité de contrôler votre clavier et votre souris.

- Pourquoi ? Elle est nécessaire pour que l'utilisateur distant puisse effectuer des réglages ou résoudre un problème à votre place.
- Comment gérer ? Activez-la uniquement pendant la session d'assistance et désactivez-la une fois le problème résolu.

## **Autorisation pour l'enregistrement de l'écran**

Avec cette autorisation, l'utilisateur distant peut enregistrer ce qui se passe sur votre écran.

- Pourquoi ? Cela peut être utile pour documenter une session ou conserver une preuve d'intervention. Toutefois, cela peut poser des problèmes de confidentialité si des informations sensibles sont enregistrées.
- Comment gérer ? Activez cette autorisation uniquement si vous êtes d'accord avec l'enregistrement.

## **Autorisation pour l'accès aux paramètres système**

Cette option permet à l'utilisateur distant de modifier des paramètres système sur votre appareil.

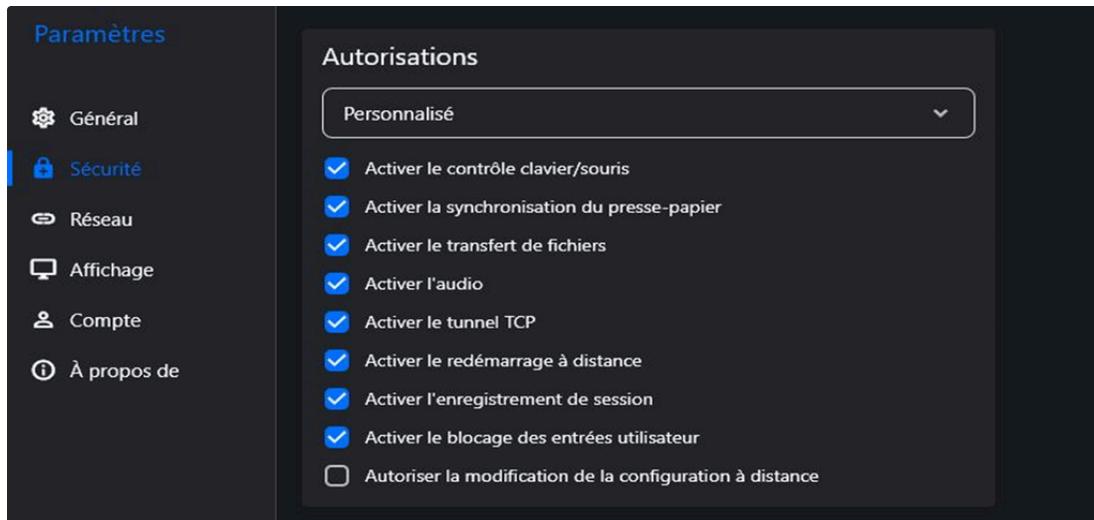
- Pourquoi ? Elle peut être nécessaire pour résoudre des problèmes techniques qui demandent des droits administrateurs.
- Comment gérer ? Ne donnez cette autorisation qu'à une personne de confiance. Vérifiez les modifications après la session.

## **Autorisation pour le chat intégré**

RustDesk offre un chat intégré permettant à l'utilisateur distant de communiquer avec vous. Cette autorisation est activée par défaut et ne pose pas de risques majeurs.

- Pourquoi ? Elle facilite la communication pendant la session pour expliquer les étapes ou poser des questions.
- Comment gérer ? Vous pouvez désactiver le chat si vous préférez communiquer par un autre moyen.

En combinant plusieurs de ces options, vous pouvez sécuriser efficacement votre utilisation de RustDesk tout en maintenant une bonne expérience utilisateur.



## 8 - Conclusion ☺

Ce guide vous a permis de configurer un serveur RustDesk sécurisé et performant sur votre serveur dédié. En suivant les étapes détaillées, vous avez pu mettre en place un système qui offre un accès à distance fiable et efficace. Ce processus est essentiel pour garantir que les utilisateurs puissent se connecter à leurs machines à distance sans compromettre la sécurité de leurs données. Grâce à un chiffrement de bout en bout, toutes les communications entre les appareils sont protégées, assurant ainsi la confidentialité et l'intégrité des informations échangées. Ce niveau de sécurité est primordial dans un monde où les menaces numériques sont de plus en plus fréquentes, et il permet aux utilisateurs de travailler en toute tranquillité, sachant que leurs données sont en sécurité.

### 💡 Suggestions pour la sécurité : ☺

- Pare-feu : Configurez un pare-feu pour restreindre les ports accessibles.
- Certificat SSL : Installez un certificat SSL payant pour un chiffrement HTTPS.
- Sauvegardes : Effectuez des sauvegardes régulières des fichiers de configuration.

Pour toute question ou difficulté, veuillez contacter les auteurs de ce document ou consulter les articles d'aide et les forums Graylog.

#### Editorial Team:

**Ulrich Sostaire Ngansop Njanou** (Junior Network Administrator)

**Gildas Fotso Tabafo** (Junior Network Administrator)