RustDesk Self-Hosted on AWS EC2 (Ubuntu LTS) Step-by-Step Beginner Guide

Generated: 2025-09-16 15:20 UTC

This guide walks you through launching an Ubuntu EC2 instance, opening the correct ports, installing and running the RustDesk ID server (hbbs) and relay server (hbbr) with Docker Compose, and configuring Windows clients to connect using your server's public IPv4 address. Throughout the guide, replace <EC2_PUBLIC_IP> with your instance's public IPv4. Example used in this document: 13.200.13.23

Prerequisites

- An AWS account with access to the EC2 console.
- A downloaded SSH key pair (.pem) created at instance launch.
- Your public IPv4 address of the EC2 instance (ideally an Elastic IP).
- Two Windows PCs with the RustDesk client installed for testing.

Part A — Launch the EC2 instance

1) Region: Choose the region closest to you (e.g., Asia Pacific (Mumbai) ap-south-1).

2) Launch instances →

- Name: rustdesk-prod (or any name you like)
- AMI: Ubuntu Server 24.04 LTS (HVM), 64-bit (x86)
- Instance type: t3.micro (okay for light use; you can resize later)
- Key pair: Create new → type RSA → download the .pem and keep it safe.
- Network settings (Security Group inbound rules):

```
    - SSH: TCP 22 → Source: My IP (recommended). Update if your IP changes.
    - Custom TCP 21115 → Source: 0.0.0.0/0
    - Custom TCP 21116 → Source: 0.0.0.0/0
    - Custom UDP 21116 → Source: 0.0.0.0/0
    - Custom TCP 21117 → Source: 0.0.0.0/0
    - (Optional for web client) Custom TCP 21118 and 21119 → Source: 0.0.0.0/0
  Leave outbound = All traffic.
```

- Storage: 16–20 GB gp3 is fine.

3) Elastic IP (strongly recommended): EC2 → Elastic IPs → Allocate → Associate with your instance.

Part B — Connect to the instance via SSH (Windows)

PowerShell (pick the correct path for your .pem):

```
ssh -i "$env:USERPROFILE/.ssh/rustdesk-key.pem" ubuntu@<EC2_PUBLIC_IP>
# or, if the file is in Downloads:
ssh -i "$env:USERPROFILE/Downloads/rustdesk-key.pem" ubuntu@<EC2_PUBLIC_IP>
```

Tips:

- Username is ubuntu for Ubuntu AMIs.
- If your public IP changes, update the SSH rule (My IP) in the security group.

Part C — Prepare Ubuntu (updates + firewall)

Become root and update the system, then mirror the security-group ports in UFW:

```
sudo -s
apt-get update && apt-get -y upgrade
apt-get -y install ufw curl
ufw allow 22/tcp
ufw allow 21115:21119/tcp
ufw allow 21116/udp
ufw --force enable
```

Part D — Install Docker & Compose plugin

```
curl -fsSL https://get.docker.com | sh
apt-get -y install docker-compose-plugin
usermod -aG docker ubuntu
newgrp docker     # make the docker group active for this shell
```

Part E — Deploy RustDesk server (hbbs + hbbr) with Docker Compose

1) Create a working directory and write compose.yml:

```
mkdir -p /opt/rustdesk-server
cd /opt/rustdesk-server

cat > compose.yml <<'YAML'
services:
  hbbr:
    image: rustdesk/rustdesk-server:latest
    command: hbbr
    volumes:
      - ./data:/root
    network_mode: host
    restart: unless-stopped

  hbbs:
    image: rustdesk/rustdesk-server:latest
    command: hbbs
    volumes:
      - ./data:/root
    network_mode: host
    depends_on:
      - hbbr
    restart: unless-stopped
YAML
```

(Why this layout? Both services mount ./data:/root so they share one keypair; host networking exposes the default ports directly.)

2) Start services and check:

```
docker compose up -d
docker compose ps
```

3) Confirm listeners (21115/21116/21117 and optionally 21118/21119):

```
ss -lntup | egrep '2111[5-9]' || true
```

4) Get the server public key (paste this into clients):

```
for i in {1..20}; do [ -f /opt/rustdesk-server/data/id_ed25519.pub ] && break || sleep 1; done
```

```
cat /opt/rustdesk-server/data/id_ed25519.pub
```

Copy the entire single line. Do NOT edit or convert it. This is not your SSH key; it’s RustDesk’s server key for clients.

Part F — Configure Windows clients

On EACH Windows PC:

1) Open RustDesk → Settings → Network

2) Set:

- ID Server: <EC2_PUBLIC_IP> (example: 13.200.13.23)
- Relay Server: <EC2_PUBLIC_IP> (example: 13.200.13.23)
- Key: paste the exact line from /opt/rustdesk-server/data/id_ed25519.pub

3) Click OK/Save. Exit RustDesk from the tray icon and open it again.

4) From one PC (Controller), enter the ID of the other PC (Target) and click Connect.

On the Target, approve the connection, or configure unattended access with a strong password.

Part G — Force relay (optional, for testing)

Edit /opt/rustdesk-server/compose.yml and add under hbbs (indented correctly):

```
    environment:
      - ALWAYS_USE_RELAY=Y
```

Apply:

```
    cd /opt/rustdesk-server
    docker compose up -d
```

Remove the environment block later if you prefer P2P when possible.

Part H — Quick troubleshooting

Server checks:

```
docker compose ps
docker compose logs --tail=80 hbbs
docker compose logs --tail=80 hbbr
ss -lntup | egrep '2111[5-9]'
```

Windows reachability:

```
Test-NetConnection <EC2_PUBLIC_IP> -Port 21116
Test-NetConnection <EC2_PUBLIC_IP> -Port 21117
```

Common issues:

• Key mismatch: Re-copy exactly from /opt/rustdesk-server/data/id_ed25519.pub and restart the client app.

To regenerate keys:

```
cd /opt/rustdesk-server
docker compose down
rm -f data/id_ed25519*
docker compose up -d
cat data/id_ed25519.pub
```

• Timeouts/relay errors: Ensure Security Group AND UFW allow TCP 21115/21116/21117 and UDP 21116.

• Windows firewall prompts: Allow RustDesk on Private/Public networks.

• Public IP changed after stop/start: Attach an Elastic IP or update clients.

Part I — Maintenance & upgrades

Update server images without losing keys (they live in ./data):

```
cd /opt/rustdesk-server
docker pull rustdesk/rustdesk-server:latest
docker compose down
docker compose up -d
```

Useful commands:

```
docker compose restart hbbs hbbr
docker compose logs -f hbbs
docker compose logs -f hbbr
```

Part J — One-shot installer (copy/paste)

Run after SSH'ing to the instance; it installs everything, starts services, and prints your server public key.

```
sudo -s <<'EOS'
set -euo pipefail

apt-get update && apt-get -y upgrade
apt-get -y install ufw curl
ufw allow 22/tcp
ufw allow 21115:21119/tcp
ufw allow 21116/udp
ufw --force enable

curl -fsSL https://get.docker.com | sh
apt-get -y install docker-compose-plugin || true
usermod -aG docker ubuntu || true

mkdir -p /opt/rustdesk-server
cd /opt/rustdesk-server

cat > compose.yml <<'YAML'
services:
  hbbr:
    image: rustdesk/rustdesk-server:latest
    command: hbbr
    volumes:
      - ./data:/root
    network_mode: host
    restart: unless-stopped
  hbbs:
    image: rustdesk/rustdesk-server:latest
    command: hbbs
    volumes:
      - ./data:/root
    network_mode: host
    depends_on:
      - hbbr
    restart: unless-stopped
```

```
    YAML

    docker compose up -d

    for i in {1..20}; do [ -f /opt/rustdesk-server/data/id_ed25519.pub ] && break || sleep 1; done

    echo "==== RustDesk SERVER PUBLIC KEY (paste into each client Settings > Network > Key) ===="

    cat /opt/rustdesk-server/data/id_ed25519.pub || echo "Key not ready yet; run: docker compose
logs hbbs"

    EOS
```