Connecting with SSH

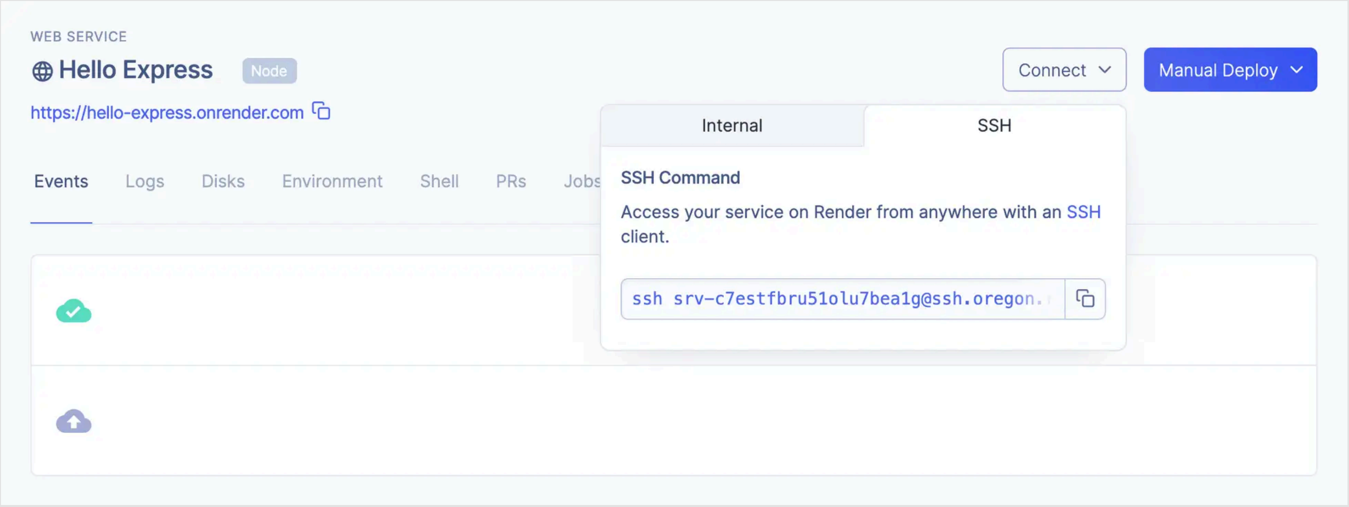
You can connect to your services on Render using SSH, in addition to using the shell in the Render Dashboard.

How to connect to a service

1. If this is your first time using SSH with a Render service, add an SSH key to your Render account.

2. Find the service you want to connect to in the dashboard.

3. Click the **Connect** button, and then click the **SSH** tab.



SSH Command

Access your service on Render from anywhere with an SSH client.

```
ssh srv-c7estfbru51olu7bea1g@ssh.oregon.
```

4. Copy the SSH command to your clipboard.

5. In a terminal, paste the command you copied earlier.

SHELL

```
$ ssh YOUR_SERVICE@ssh.YOUR_REGION.render.com
```

> If your service has multiple instances, you will connect to one instance at random.

6. You may see a warning like this:

TEXT

```
The authenticity of host 'render.com (IP_ADDRESS)' can't be establi
ED25519 key fingerprint is (SSH_KEY_FINGERPRINT)
Are you sure you want to continue connecting (yes/no)?
```

Verify that the fingerprint in the message you see matches Render's public key fingerprint. If it does, then type `yes` .

If you receive a "permission denied" message, see Troubleshooting SSH.

Render's SSH key fingerprints

Public key fingerprints can be used to validate a connection to a remote server.

Render's public SSH key fingerprints are as follows:

REGION	FINGERPRINT
Oregon	`SHA256:KkZPgnApmttFYSkdJsCi7B01sgZPMI6kY53MDbbanGM`
Ohio	`SHA256:kRDsLlrHqOyqso58sEKyO6ZFMPj7p24zfNxYJ42yXGI`
Virginia	`SHA256:NCpSwboPnqL/Nvyy2Qc8Kgzpc3P/f3w5wDphhc+UZO0`

REGION	FINGERPRINT
Frankfurt	SHA256:dBRrCEA0tBkvaYLzzDw/mzaANw6nUJO961Zx806spZs
Singapore	SHA256:CUlRyv4TZ0vmHwmhsJkII/pz2cO4IgvR+ykqnRsOQFs

You can also directly add Render's public keys to your `$SSH_DIR/known_hosts` file. Render's full set of entries is as follows:

BASH

```bash
# RENDER PUBLIC KEYS
# ------------------

# Oregon
ssh.oregon.render.com ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIFON8eay2FgHD

# Ohio
ssh.ohio.render.com ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAINMjC1BfZQ3CYot

# Virginia
ssh.virginia.render.com ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIJ6uO0jKQX9

# Frankfurt
ssh.frankfurt.render.com ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAILg6kMvQOG

# Singapore
ssh.singapore.render.com ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIGVcVcsy7F
```

Service instance memory usage

When connecting to a service instance via SSH, memory consumed is part of the service plan's memory limits. Using SSH requires 2 MB of memory plus ~3 MB for each active session, not including the memory of the programs run by each session.

For example, if SSHing into one service instance from two different computers to run bash, memory usage would be:

- 8 MB for SSH
  - 1.8 MB for SSH access
  - 2x3 MB for the two SSH sessions
- 7 MB for bash
  - 2x3.5 MB for the two bash processes

Total memory usage would be 15 MB.

Connection lifetime

SSH connections are kept open for as long as possible. When infrastructure upgrades are required we give existing connections 1 hour before automatically closing them. SSH connections are also closed when your service is redeployed either manually or with auto-deploy.

If you are running a long-running command consider using Jobs or running the command with nohup to minimize the likelihood of interruptions.

Use with a Docker container

Our Native runtimes provide a working out-of-the-box experience with SSH connections.

For Docker services, openSSH (`openssh-server`) needs to be installed in the container, and if the Dockerfile specifies a non-root user, that user must have shell access.

If your Dockerfile references a parent image, you will need to perform these steps in a Dockerfile that you control, making use of the `USER` instruction to change back to a root user and `usermod` (or equivalent) to modify the non-root user.

Troubleshooting

Follow steps in SSH Troubleshooting if you are unable to connect.

Limitations

- Services that haven't been deployed since `January 11, 2022` need to be redeployed to enable SSH.
- SSH is not supported for free plan services, cron jobs, or static sites.
- You can only SSH into a service that's owned by a workspace you belong to.
- Some Docker service configurations are not supported.
  - Dockerfiles that use the root account cannot lock the account. Use `usermod --unlock root` or `passwd -u root` to unlock the account.
  - Dockerfiles that specify a non-root user with the USER instruction must have user accounts set up. To SSH into these services, either create a user account with a tool like `useradd` or use the root (UID 0) account.
  - Dockerfiles must have a `~/.ssh` directory.
  - The `~` and `~/.ssh` directories must be owned by the running user and have a permission that only gives the owner write access. For example, use `chmod 0700 ~/.ssh` to change the permission.
  - A disk may not be mounted to the `$HOME` directory of the running user.