Only you can see this message

This story's distribution setting is on. You're in the Partner Program, so this story is eligible to earn money. Learn more

SSH Tunneling, a Guide

Craig Godden-Payne
Jul 1 · 3 min read ★

SSH tunneling is something I have to do all the time, yet its something that never seems to stick in my brain.

Photo by Jordan Harrison on Unsplash

I have created a guide (with examples) to quickly and easily demystify SSH tunneling and get you going.

. . .

Before we start — SSH Config

Use ssh config to save your connections, to save having to go through your bash history to find the command you used last week that worked.

I am a serial bash history searcher, but there are easier ways to setup your ssh tunnels and connections in general using SSH config.

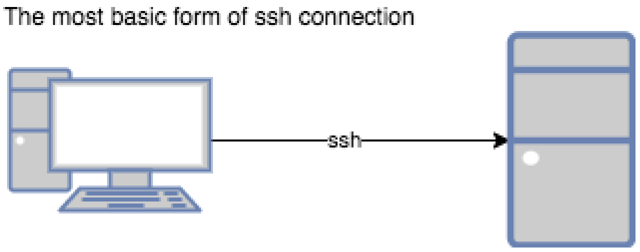
SSH config is usually located at: ~/.ssh/config

If it isn't you should just be able to create that file.

You can add blocks into this config, so that you can in the future use an alias.

. . .

Scenario one — Connect to one server



In its most basic form, you can connect from your machine to a server. This is pretty straightforward and can be achieved as follows:

```
Host my-prod-public-bastion
  HostName 1.2.3.4
  User ec2-user
  IdentityFile ~/.ssh/my-prod-public-bastion
```

In the case above, I can now connect using the alias as follows:

```
ssh my-prod-public-bastion
```

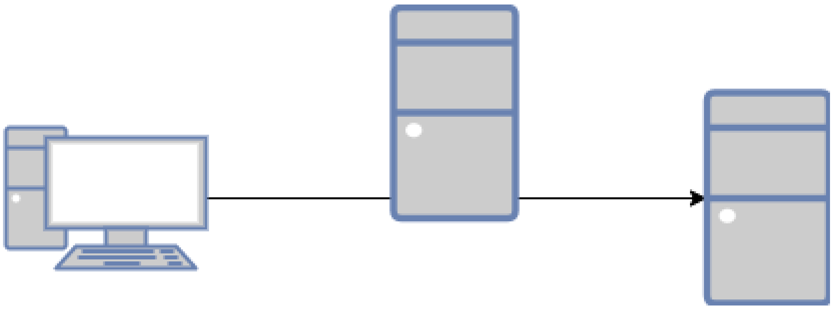
. . .

Creating a Simple, Low-Cost Twitter Bot, utilising Serverless Technologies.

People have a love-hate relationship with twitter bots.

medium.com

Scenario two — Jumping across bastions to get where you want



It is possible to ssh from your machine to another machine which will forward the traffic to another server.

You can use SSH config to achieve this as follows:

```
Host my-prod-public-bastion
  HostName 1.2.3.4
  User ec2-user
  IdentityFile ~/.ssh/my-prod-public-bastion

Host my-prod-private-bastion
  HostName 10.0.0.53
  User ec2-user
  IdentityFile ~/.ssh/my-prod-private-bastion
  ProxyCommand ssh -W %h:%p my-prod-public-bastion
```

You should then be able to ssh using the alias, and it will appear as though you are connected to server 2 when actually the traffic is being routed from server 1 to server 2 behind the scenes.

```
ssh my-prod-private-bastion
```

The Only Step You Need to Progress in Your Career as a Developer or Engineer.

I don't usually like to write articles about myself, as it feels a bit self-indulgent, but I thought it would be useful…

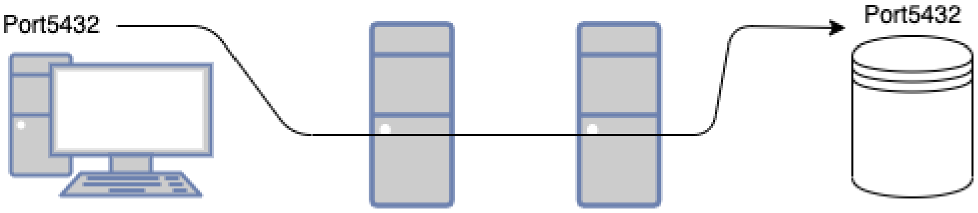
medium.com

A few notes about the above config though…

> *The `-W` option is built into versions of OpenSSH 5.4+ unless your distro back-ported this feature. The feature connects stdio on the client to a single port forward on the server. This allows, for example, using ssh as a ProxyCommand to route connections via intermediate servers.*

. . .

Scenario three — Portforwarding a local port to a remote port



It is possible to map a local port on your machine, to a remote port over SSH.

This is so powerful.

You can achieve this as follows

```
ssh -L 5432:my-database-server:5432 my-prod-private-bastion
```

In this example, I have an instance of MySQL, which is only accessible from my-prod-private-bastion, making it very difficult to access.

Using SSH in this way means I can now connect to localhost:5432, and run queries against the database, just like I am sat on the server `my-prod-private-bastion`

Ssh Networking DevOps

About Help Legal