Some useful ssh config option

Link: https://taozhi.medium.com/some-useful-ssh-config-option-7858a58c5e7b

When managing multiple Linux servers, we use SSH for logging in and performing tasks. Understanding how to configure SSH properly is essential for efficient server management.

Basic Config

```
Host my_jump
    identityfile "~/.ssh/my_jump"
    hostname 47.254.197.212
    hostkeyalias my_jump
    user root
    port 22
```

In the above config, “my_jump” is the hostname supporting wildcards to match multiple servers simultaneously.

The identityfile specifies the authorized private keys, hostname is the server’s IP address, and hostkeyalias is useful for connecting to the server when its IP address changes without needing to update known_hosts. The user and port specify the SSH login credentials.

Reuse the sock

Upon relogging into the server, how can we bypass entering the password and reuse the previous session to quickly reconnect? We should the control setting in ssh config.

```
Host *
    serveraliveinterval 60
    keepalive yes
    controlmaster auto
    controlpath ~/.ssh/socks/%h-%k-%p-%r
    controlpersist yes
```

By using the above configuration, we set the controlpath for all servers using the ‘*’ symbol in the Host field. The controlpath specifies the socket path.

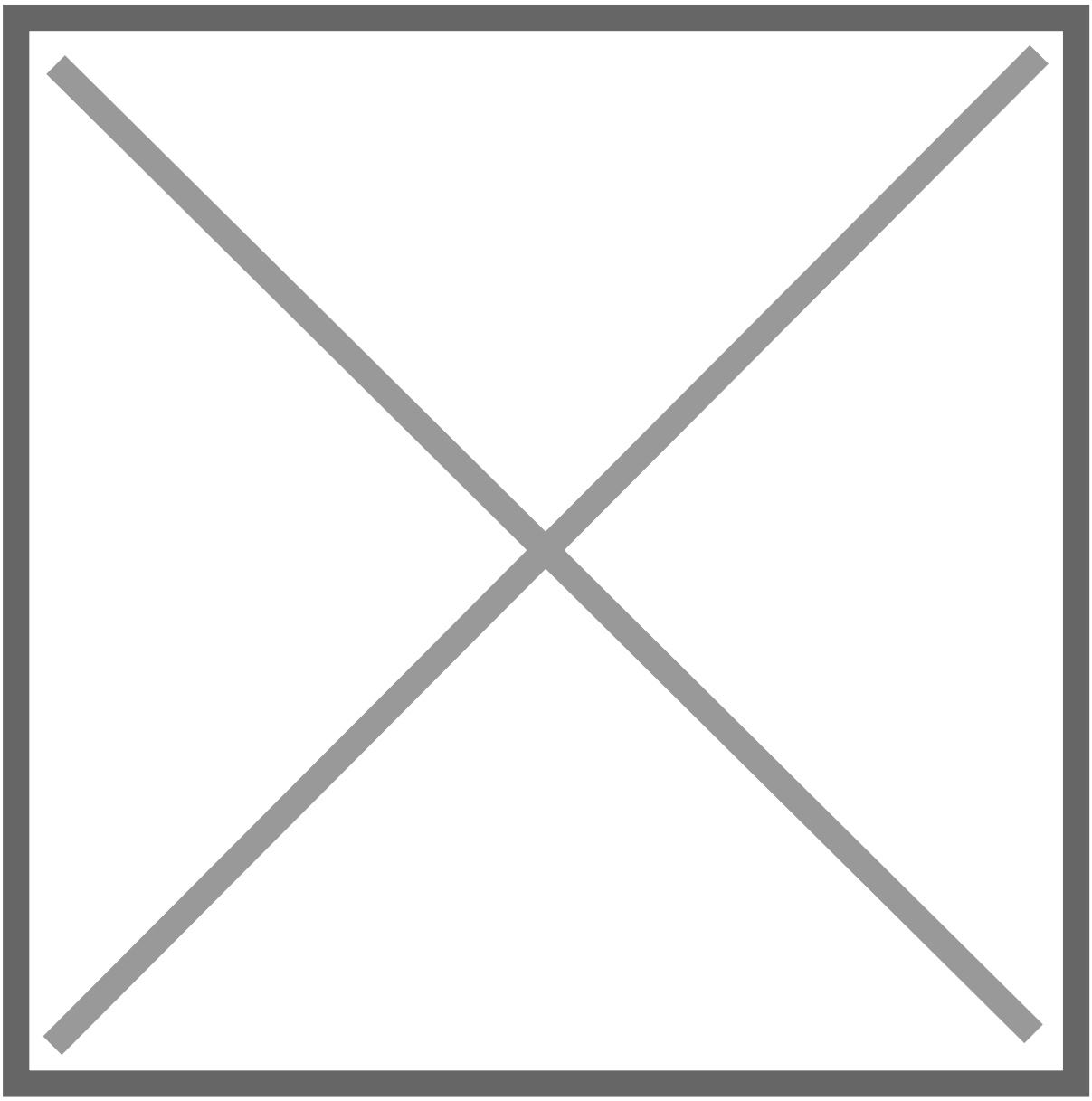
%h represents the host IP.

%k represents the hostname.

%p represents the port.

%r represents the username.

When you connect to a server using ssh, you should see a socket file present. `~/.ssh/socks`.



Set Jump Server

To secure production servers inaccessible for direct login, we can first connect to a jump server, then use SSH through the jump server to access the production server. Automating this process is possible by configuring ProxyCommand or ProxyJump in the SSH settings.

Config the jump server a and b first.

```
Host jump-server-a
 HostKeyAlias jump-server-a
 Hostname 100.97.200.66

Host jump-server-b
 HostKeyAlias jump-server-b
 Hostname 100.97.200.67

Host jump-server-*
 HashKnownHosts no
 ServerAliveInterval 60
 Port 22
 User root
 PreferredAuthentications publickey
 IdentityFile ~/.ssh/id_taozhi
 Controlpath ~/.ssh/socks/%h-%k-%p-%r
 ControlMaster auto
 ControlPersist 5m
 setenv LC_ALL=C.UTF-8
```

Config the production servers

```
Host production-server-a
 ProxyJump jump-server-a

Host production-server-b
 ProxyJump jump-server-b

Host production-server-c
 ProxyCommand ssh -W %h:%p jump-server-b

Host production-server-*
 LogLevel ERROR
 UserKnownHostsFile /dev/null
 StrictHostKeyChecking no
 Port 22
 User root
 IdentityFile ~/.ssh/id_taozhi
 controlmaster no
 setenv LC_ALL=C.UTF-8
```

Following configuration, we can login to the production server locally.

```
ssh -o Hostname=172.16.28.19 production-server-a
```

You can log in to the production server with one command now.

Conclusions

SSH is a versatile command with numerous configuration options. More options can be found for reading here. If you have any useful ssh config you are using, please comment it, let using it together.

Revision #1
Created 2025-01-05 11:50:27 UTC by Administrador
Updated 2025-05-16 11:49:54 UTC by Administrador