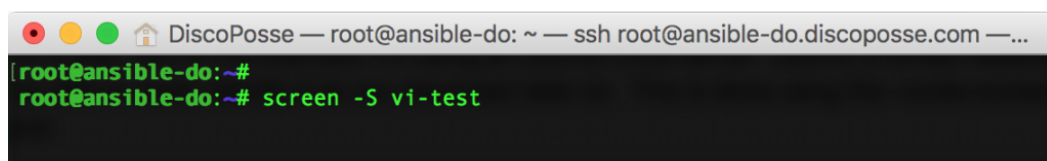


A Quick Intro Screen on Linux

For fans of RDP on Windows, one of the great utilities that you have available on Linux is one called Screen. Using Screen lets you start up a terminal session within your Linux host that can be left persistently in the background for you to re-enter at another time from another location.

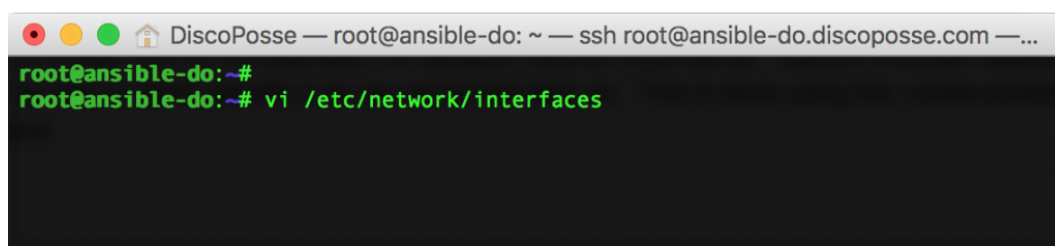
Screen Basics

This is even available on a Mac, but for my example, I'm using an Ubuntu Linux server. Launch a Screen session and give it a name so that you will be able to easily identify it as you reconnect later on. This is done using the `screen -S session name` command:



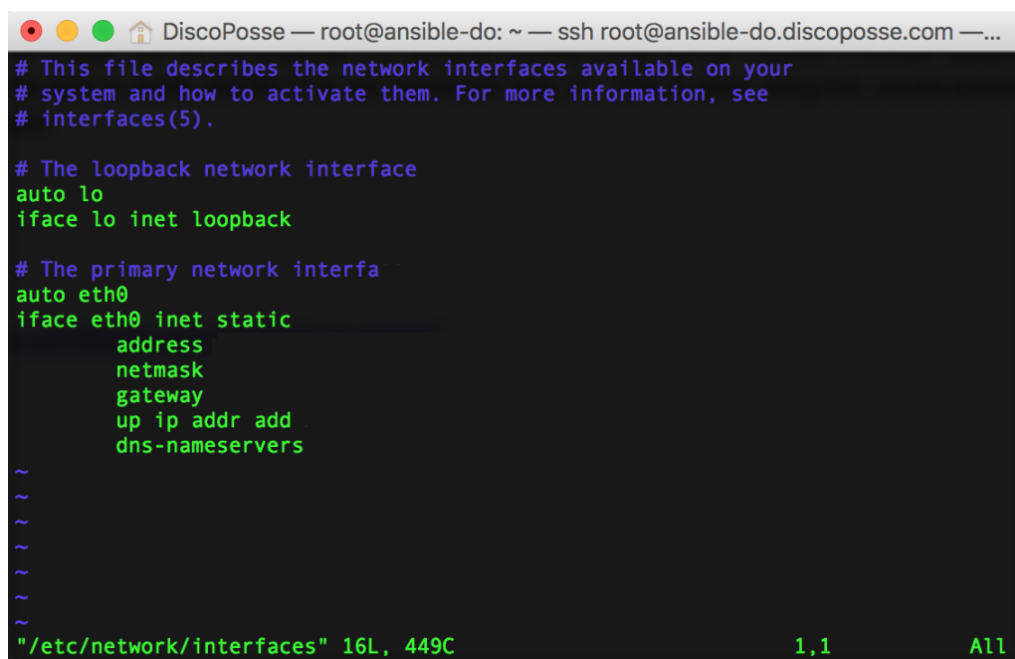
```
DiscoPosse — root@ansible-do: ~ — ssh root@ansible-do.discoposse.com —...
[root@ansible-do:~#
root@ansible-do:~# screen -S vi-test
```

Let's launch `vi` to open a file. I'll use my `/network/hosts/interfaces` file in this example.



```
DiscoPosse — root@ansible-do: ~ — ssh root@ansible-do.discoposse.com —...
root@ansible-do:~#
root@ansible-do:~# vi /etc/network/interfaces
```

We can see the file is open now on our screen. This opens up a window that you know you'll have to exit in order to return to a terminal session. It's a great use of Screen because we can jump out to the system as needed:



```
DiscoPosse — root@ansible-do: ~ — ssh root@ansible-do.discoposse.com —...
# This file describes the network interfaces available on your
# system and how to activate them. For more information, see
# interfaces(5).

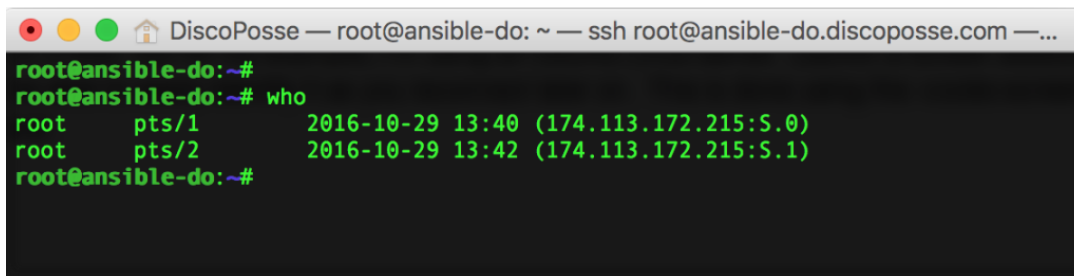
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interfa
auto eth0
iface eth0 inet static
    address
    netmask
    gateway
    up ip addr add
    dns-nameservers

~
~
~
~
~
~
"/etc/network/interfaces" 16L, 449C          1,1          All
```

Use the `Ctrl-A C` sequence. That means to use the `Ctrl-A` followed by the `C` key. This will bring you back to an interactive shell. If we run a `who` command, you can see there are two active

sessions:

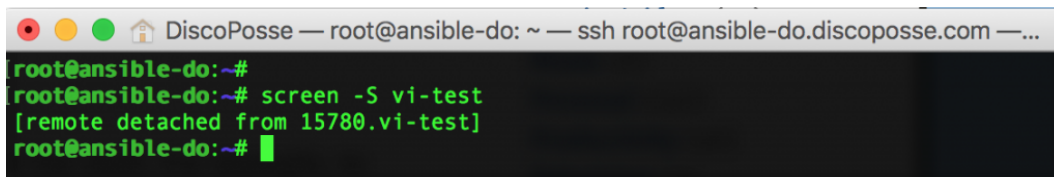


```
DiscoPosse — root@ansible-do: ~ — ssh root@ansible-do.discoposse.com —...
root@ansible-do:~#
root@ansible-do:~# who
root pts/1 2016-10-29 13:40 (174.113.172.215:S.0)
root pts/2 2016-10-29 13:42 (174.113.172.215:S.1)
root@ansible-do:~#
```

In order to switch back and forth between the different Screen sessions, use the `Ctrl-A` sequence. Sometimes you will find that you have to tap the key sequence more than once to get the sessions to switch. Let's try it a couple of times to see that you can go back and forth from your `vi` window to your interactive bash shell.

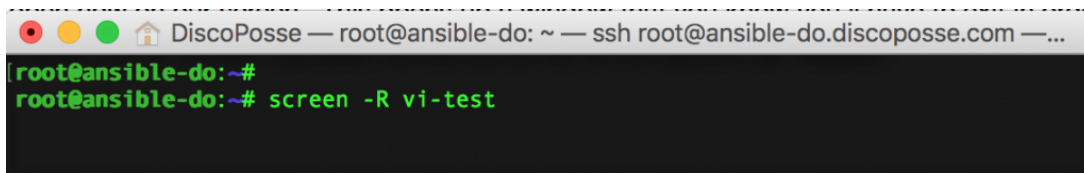
Pretty cool, right!

Now let's test out the ultimate process. We are going to detach the Screen session. Use the `screen -d` command which will detach us altogether from Screen. You will see the detach message when it works:



```
DiscoPosse — root@ansible-do: ~ — ssh root@ansible-do.discoposse.com —...
root@ansible-do:~#
root@ansible-do:~# screen -S vi-test
[remote detached from 15780.vi-test]
root@ansible-do:~#
```

Exit from your terminal session altogether to prove out our concept. Re-enter a remote session into the same server and then we can re-attach to the screen session using the `screen -R vi-test` command:



```
DiscoPosse — root@ansible-do: ~ — ssh root@ansible-do.discoposse.com —...
root@ansible-do:~#
root@ansible-do:~# screen -R vi-test
```

This is why we choose meaningful names for our Screen sessions. It makes it much easier to know which sessions we are using as we re-attach later on. Now that re-attached, you will see a familiar looking session. Yay!

```
DiscoPosse — root@ansible-do: ~ — ssh root@ansible-do.discoposse.com —...
# This file describes the network interfaces available on your
# system and how to activate them. For more information, see
# interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interfa
auto eth0
iface eth0 inet static
    address
    netmask
    gateway
    up ip addr add
    dns-nameservers

~
~
~
~
~
~
~
"/etc/network/interfaces" 16L, 449C          1,1          All
```

To close out each session, make sure to return to the shell and use the `exit` command. That ensures that you free up the unused session when you are done with it.

Screen sessions do not survive reboots. Just like a Windows RDP session, they will be usable as long as the system is in the current boot session.

You can use the `man screen` command to find out all about the Screen utility. This is hopefully a good start to enjoying Screen for folks who haven't known about it in the past.