Installing & Configuring

Domino 10.0.1

on

CentOS 7

Enterprise Linux

Devin S. Olson

Table of Contents

Introduction

In this document I will walk you through the necessary steps to perform a ground-up installation of 64-bit CentOS 7, the minimal configuration of the Operating System, configuration specific for IBM/HCL Domino installation, and finally the installation of 64 bit IBM/HCL Domino 10.0.1.

This document and (any future accompanying videos) are aimed at administrators who have some working knowledge / understanding of IBM/HCL Domino, but who may be unfamiliar or uncomfortable with Linux. The goal of this document is to demonstrate the very minimal requirements for Domino on Linux, point out some potential pitfalls, and ultimately show that installing and running Domino on a Linux machine is much easier than you think.

COPY / PASTE

If you are sitting at a server console with this document in electronic format, then do yourself a favor whenever you see a console command such as:

```
>#  cat /etc/selinux/config
```

Copy / paste it directly into the console. It will save you a bunch of time. Just make sure to change any specific-to-your-installation values before hitting that enter key.

Note on Fonts and Spacing

I am deliberately using larger fonts because they are **easier for me** to read. I realize they may appear gigantic on lower resolution displays, and for that I'm sorry for you.

Typographic Conventions

Throughout this document there are several different “types” of text.

- There is explanatory text, which appears in Arial.
- `Example file content or lists of information appear in Consolas.`
- `Courier New is used for command-line content (stuff you should type).`
- Console commands are **`orange bold Courier New`**.
- I use colors, *italics*, and **bold** fonts in various areas to help *stuff* stand **out**. You are a smart person, you will figure it out.

I am a bit of an old-school type of person, in that I prefer paper documents to digital. I find being able to scrawl hand-written notes onto a document comforting. This document is intended to be printed out and used / referenced many times. To that end, I have made sure to include lots of blank white space on this document for you to take your own notes. If that doesn’t suit you then just scroll on by.

"Linux" is a registered trademark of Linus Torvalds. “CentOS” is a registered trademark of the CentOS Project. “IBM”, “Domino”, and “Domino 10.0.1” are registered trademarks of International Business Machines, Incorporated. All other trademarks are property of their respective owners. As of this writing, the HCL purchase of the IBM Collaboration software portfolio has not been finalized, which is why I refer to it as “IBM” Domino and not “HCL” Domino; although I am totally stoked about the purchase. I think HCL will be able to improve certain aspects of the portfolio that, for whatever reasons, IBM was unable to do. All other content is Copyright @ 2019 by Devin S. Olson unless otherwise assigned. This document and (any future accompanying videos) are released under the Apache License, Version 2.0. If you are still reading this, you are a complete nerd and should probably move on to the more important stuff contained in the rest of this document. Seriously, there is nothing else important enough in this paragraph worth reading.

CentOS Installation

Depending upon your installation, you might be setting up a hosted server with one of the various hosting providers (such as prominic*), a Virtual Machine, or directly as the main Operating System on a physical server.

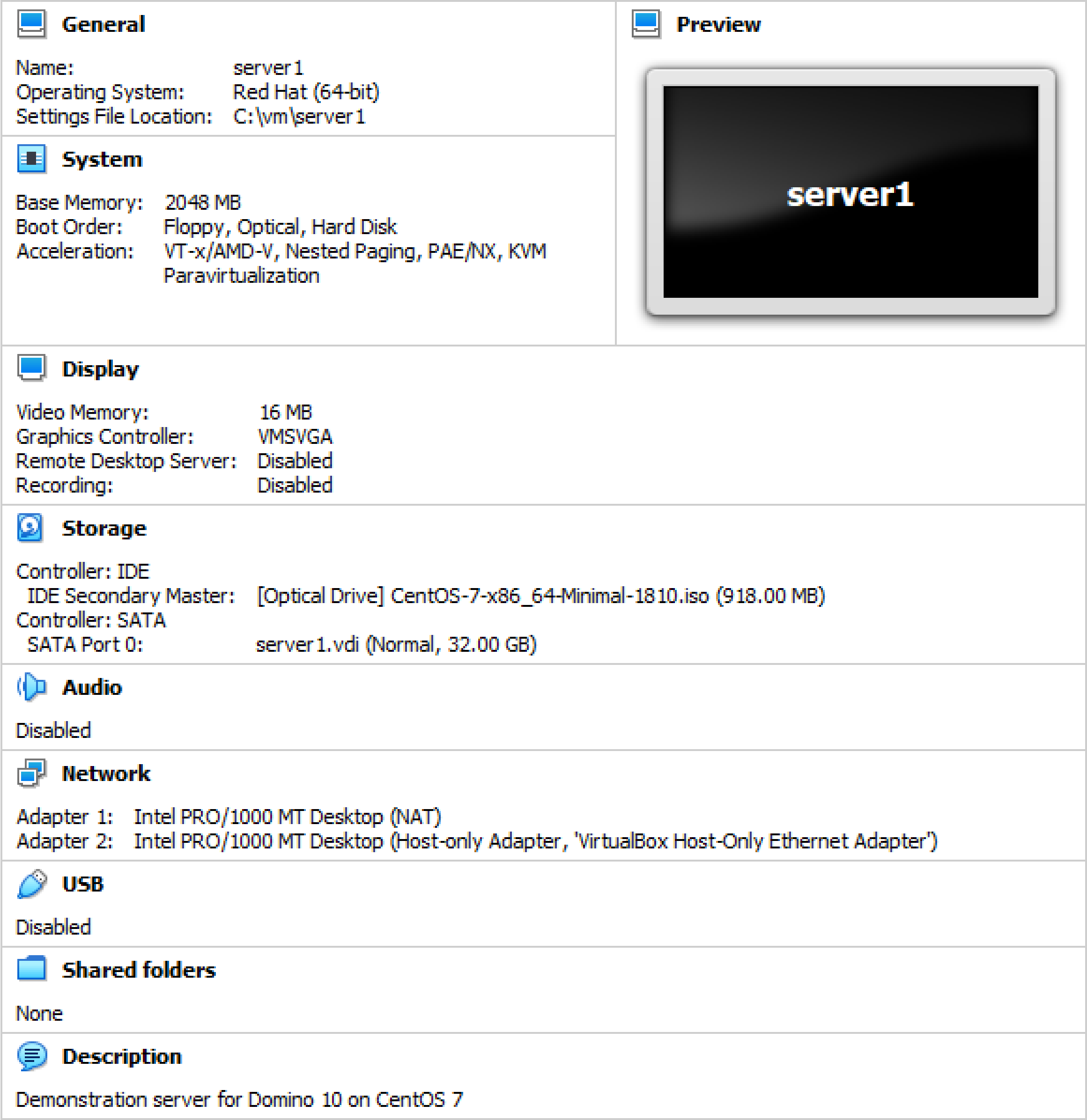
Each of these installation types have their own unique nuances for the installation. I am going to focus on the major points that all of these have in common. Most of the major hosting providers will handle the base Operating System and Network IP configuration automatically. If this fits your situation, just go ahead and skip to the next section, User Account and SSH Configuration.

For the purposes of preparing this document, I performed a Virtual Machine installation using Oracle VirtualBox running in a Windows 10 environment, and used **CentOS-7-x86_64-Minimal-1801.iso**

This is a totally uncompensated plug for the wonderful folks over at prominic (https://prominic.net); in fact, I didn't even bother to get their permission before mentioning them in this document. I have received nothing for this. The reason I'm plugging their company is simple: I have known John (and by proxy Justin) professionally for nearly 20 years. When a business stays focused on core values and takes care of their customers for that long it says something. To me that something is best summed up in a single word: integrity. They have a great and dedicated team over there. If you are looking for a hosting provider, I suggest you give them a call.

[Optional] Create Virtual Machine

- Type = Linux
- Version = Red Hat (64 bit)
- RAM = 2 Gig
- Image Type = VDI (Virtual Disk Image)
- Image Footprint = Dynamically Allocated
- Image Size = 32 Gig
- Optical Disk Drive pointing CentOS iso
- Enable Network Adapter 1 attached to NAT
- Enable Network Adapter 2 attached to Host-only Adapter
- Verify different MAC addresses (Advanced settings)



General

Name: server1
Operating System: Red Hat (64-bit)
Settings File Location: C:\vm\server1

System

Base Memory: 2048 MB
Boot Order: Floppy, Optical, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, PAE/NX, KVM Paravirtualization

Preview

Display

Video Memory: 16 MB
Graphics Controller: VMSVGA
Remote Desktop Server: Disabled
Recording: Disabled

Storage

Controller: IDE
 IDE Secondary Master: [Optical Drive] CentOS-7-x86_64-Minimal-1810.iso (918.00 MB)
Controller: SATA
 SATA Port 0: server1.vdi (Normal, 32.00 GB)

Audio

Disabled

Network

Adapter 1: Intel PRO/1000 MT Desktop (NAT)
Adapter 2: Intel PRO/1000 MT Desktop (Host-only Adapter, 'VirtualBox Host-Only Ethernet Adapter')

USB

Disabled

Shared folders

None

Description

Demonstration server for Domino 10 on CentOS 7

Install 64-bit CentOS 7 Minimal

The Minimal installation version is all that is needed. Latest versions available at www.centos.org.

During the installation you will need to answer various things like language and keyboard layout, installation destination, how to configure the partitions, the root password, time zone, and whether or not to configure a user account. **Do not create an additional user at this time.**

Partitions

Your partition scheme is entirely up to you, and a full discussion of partition layout is beyond the scope of this document. However, in order to make things fairly easy (and remembering that I'm only using a 32 G total disk size and modern SSDs make a lot of the old partitioning schemes irrelevant), I do have a few recommendations.

Dedicated DAOS and Index Partitions

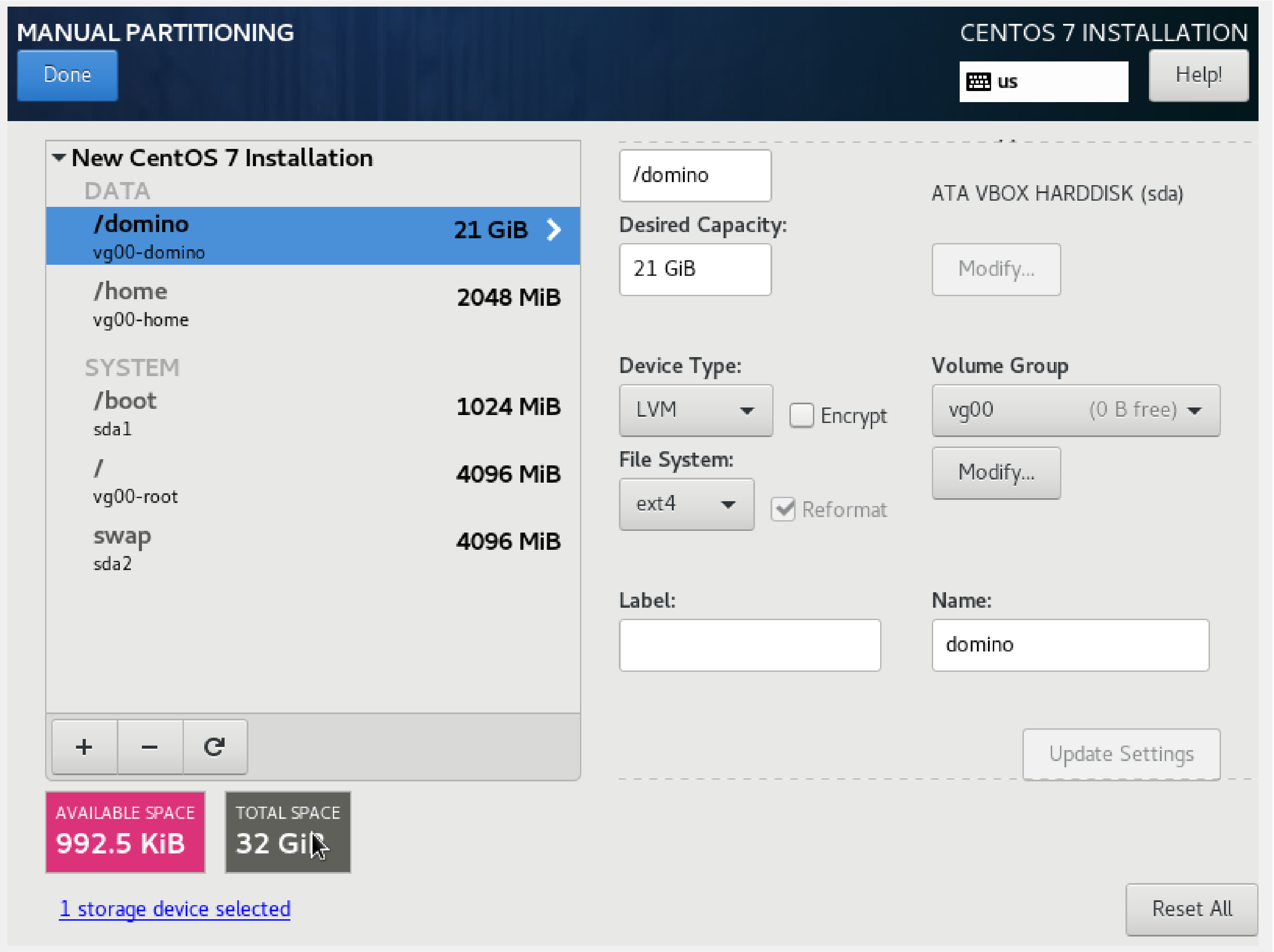
If you have multiple physical drives, then I recommend creating a DAOS partition on another drive, and an index partition on SSD. Creating dedicated partitions for DAOS or index content on the same drive (be it platter or SSD) is pointless -they need to be different physical devices. If you are using RAID, then use a different controller as well.

Regarding XFS vs ext4

Red Hat and CentOS are both pushing the new XFS File system because it is super awesome. However, it is a PIA to resize, and for a Domino server ext4 is just fine. I don't see any real benefit from going to XFS at this time.

Partition	Size	Device Type & File System	Volume Group	Notes
swap	4 Gig	Standard swap	NA	For memory up to 4 Gig, a swap partition of 2x the memory is recommended. 4 – 8 Gig should be 1.5x memory, and 8 – 32 Gig should be 1x memory, with a max of 32 Gig. (I know the math doesn't work out)

boot	1 Gig	Standard ext4	NA	A full Gig here might seem super wasteful, but have you ever had to resize a boot partition? It is a GIGANTIC hassle. Just go ahead and give yourself some extra space now while it is easy to do.
/home	2 Gig	LVM ext4	vg00	Only 2 Gig for /home? Remember, this is a **Domino server** we are setting up. Anything more than this is unnecessary.
/	4 Gig	LVM ext4	vg00	Other than /home and /domino, I am leaving everything else (/tmp, /var/, /etc) here in the root partition. If you are planning on using your server for anything else than a Domino server, you might want to consider adding additional explicit partitions; but I really don’t see a need to do so. Bear in mind this setup **is for minimal installation**. If you have enough disk space, I suggest you make this at least 8 Gig.
/domino	21 Gig	LVM ext4	vg00	I used all remaining space for the dedicated /domino partition. Only you know what your size needs are going to be. Once again, this is for a minimum functional installation.



▾ New CentOS 7 Installation

DATA
/domino 21 GiB ›
vg00-domino
/home 2048 MiB
vg00-home

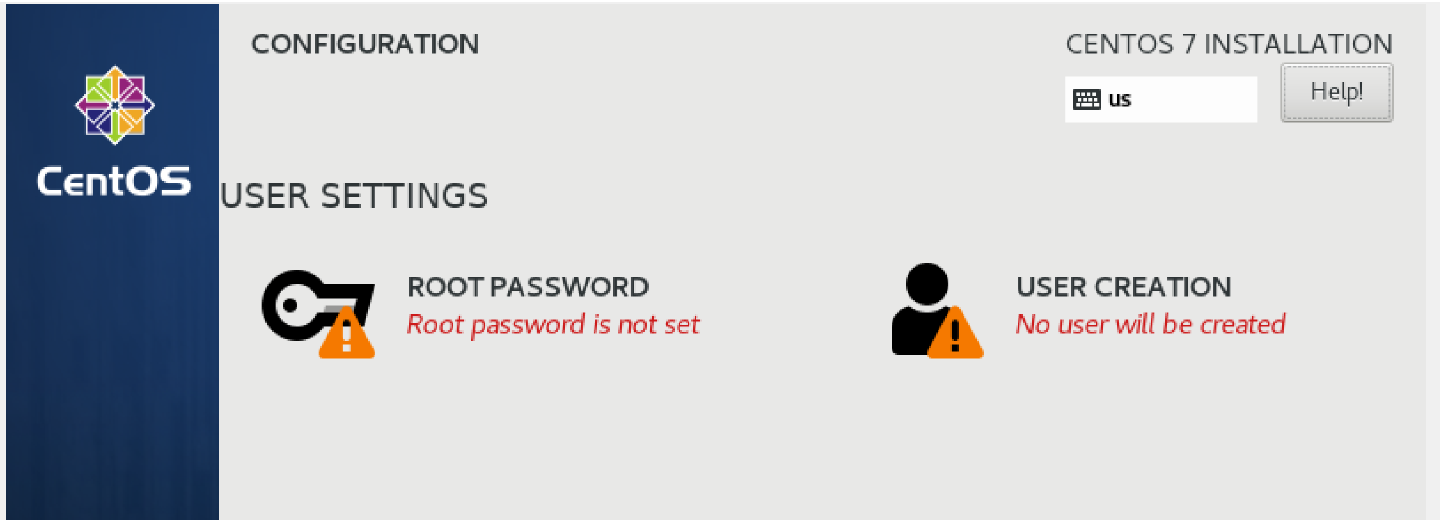
SYSTEM
/boot 1024 MiB
sda1
/ 4096 MiB
vg00-root
swap 4096 MiB
sda2

AVAILABLE SPACE 992.5 KiB
TOTAL SPACE 32 GiB

Mount Point: /domino
ATA VBOX HARDDISK (sda)
Desired Capacity: 21 GiB
Device Type: LVM
□ Encrypt
Volume Group: vg00 (0 B free)
File System: ext4
☑ Reformat
Label:
Name: domino

Root password

Create a proper strong password. Something that is easy for you, a human, to remember; and hard for a machine to guess. Don't fall for that crap being pushed by idiots saying to use passwords like *"37K@D}sk&w"*; that is a crap password because you can't remember it and it is not hard for a machine to guess. Use something like *"LoveTotallyAwesomeDomino"* or *"SpankyBeerIsTheBestInTheWorld"*. Go ahead and throw in a few salted characters replace "e" with "7" and "o" with "@" if it makes you feel better. **Long passwords are strong passwords**. Now, once you have created your password, WRITE IT DOWN and store it someplace safe. I know that will probably irritate some security professionals, but lost passwords are useless.



Root password is not set

No user will be created

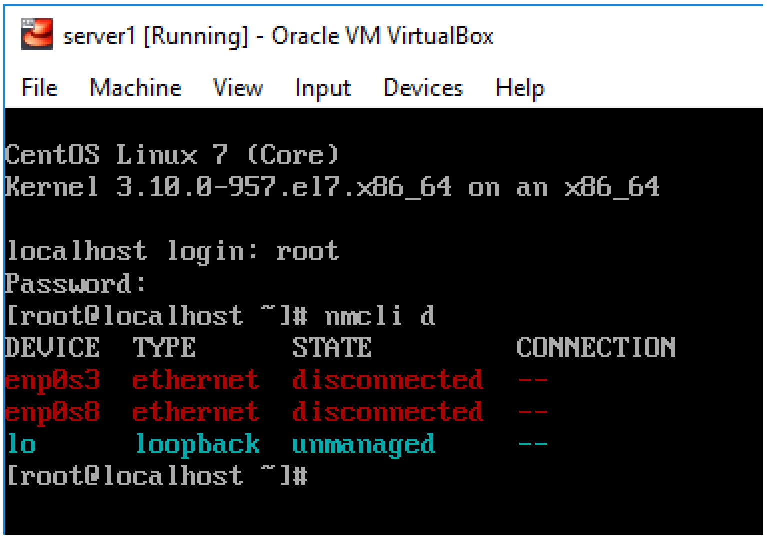
Reboot the server after the Operating System is installed and sign in as root, using the easy-to-remember password you have created.

[OPTIONAL] Configure network

I don’t intend for this document to become a dissertation on networking, protocols, DNS, or IPV4 vs IPV6. To that end I’m going to keep this as simple as possible and focus entirely on IPV4. There is a big wide world of information, written by people much smarter than me, available out there on the web. Take some time to read some of it (packet-switching can be weirdly interesting).

Check the network cards and their status

```
>#  nmcli d
```



```
CentOS Linux 7 (Core)
Kernel 3.10.0-957.el7.x86_64 on an x86_64

localhost login: root
Password:
[root@localhost ~]# nmcli d
DEVICE   TYPE      STATE         CONNECTION
enp0s3   ethernet  disconnected  --
enp0s8   ethernet  disconnected  --
lo       loopback  unmanaged     --
[root@localhost ~]#
```

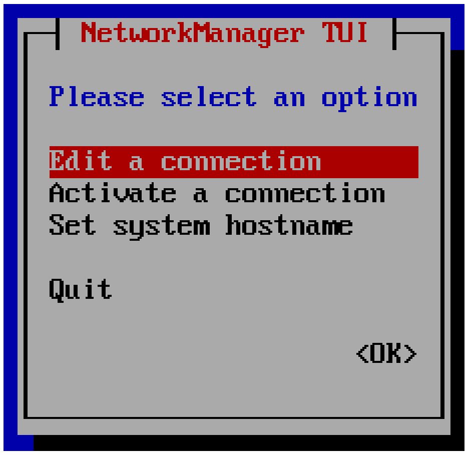
This example shows the loopback device (lo), and two network cards (enp0s3 and enp0s8).

There are two primary ways to configure your network cards in CentOS 7. You can use either the network manager utility (nmtui), or, if you are more comfortable, simply directly edit the appropriate configuration files. It really doesn’t matter which one you choose, they both accomplish the same thing.

Network Manager Utility

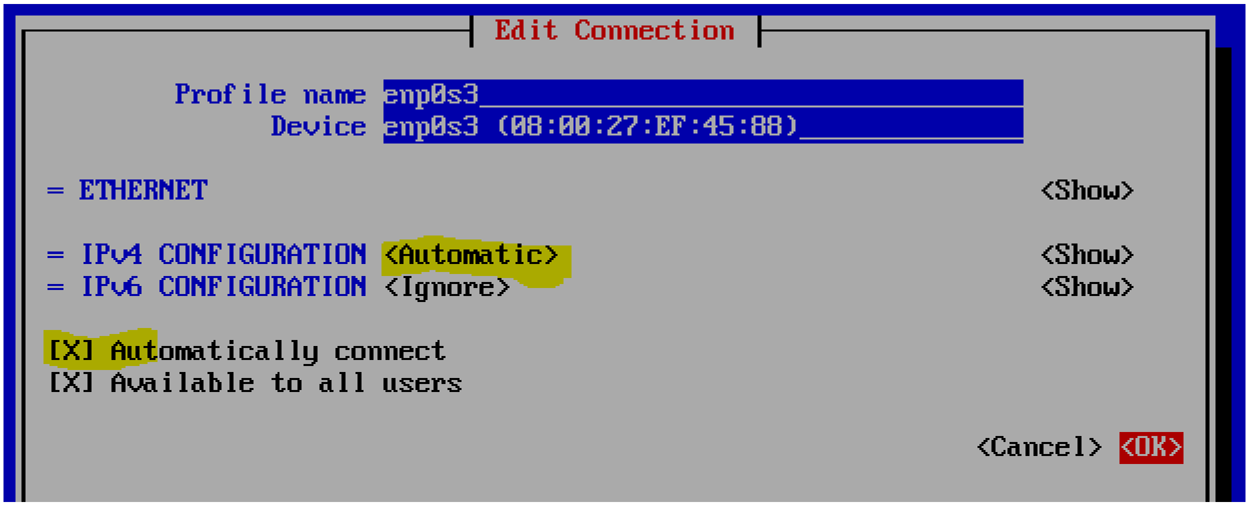
Using the network manager utility is fairly straightforward. You pick the various tasks, make the edits as needed, then save and quit. It does take a bit of patience, as interface navigation is not the best, but it works. If you would prefer to go old school (like me) and directly edit the configuration files, then skip ahead to the next section. Otherwise, fire up the utility.

```
>#  nmtui
```



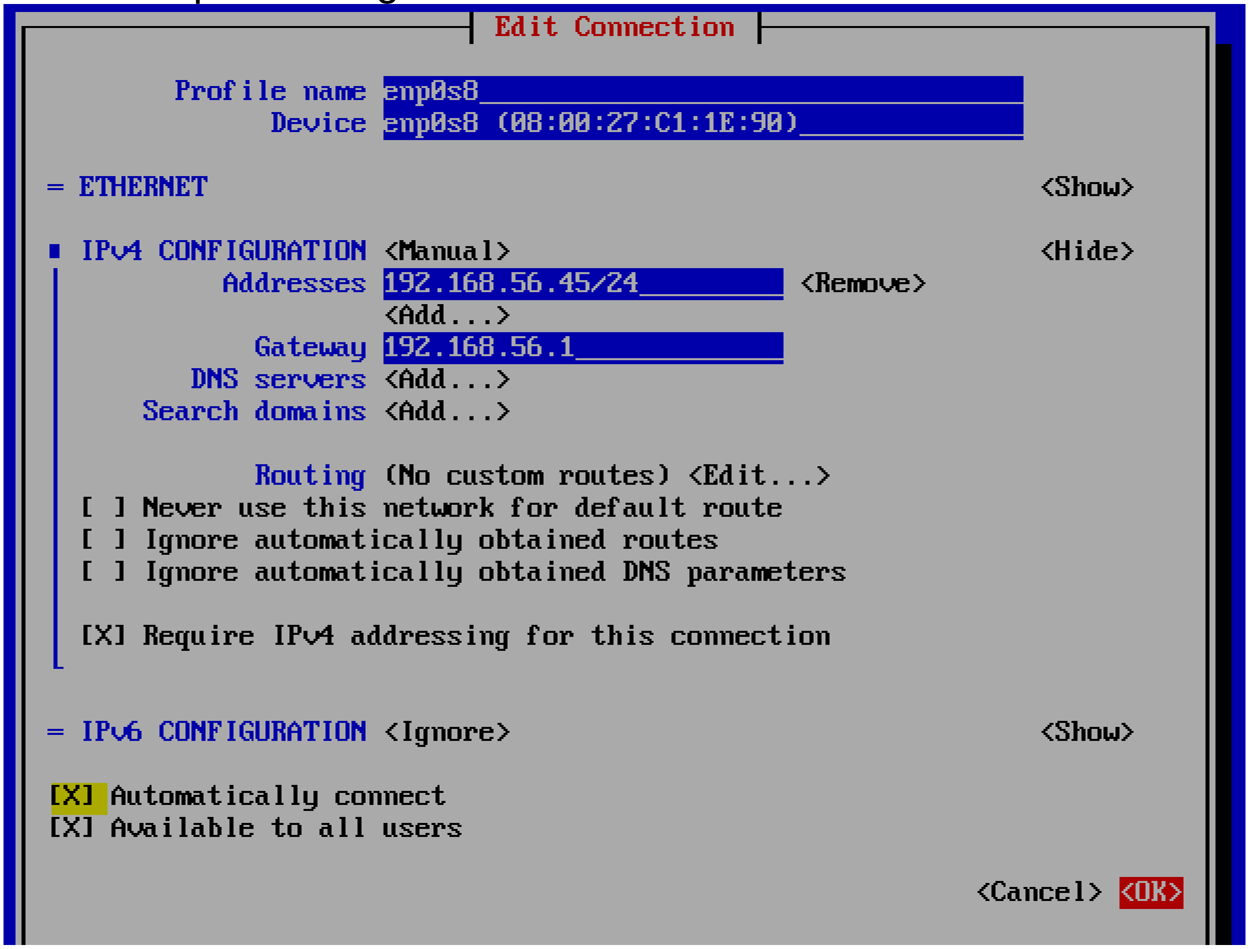
```
┤ NetworkManager TUI ├

 Please select an option

 Edit a connection
 Activate a connection
 Set system hostname

 Quit

                   <OK>
```

Device enp0s3 using DHCP



```
                              ┤ Edit Connection ├

        Profile name enp0s3_____________________________________
              Device enp0s3 (08:00:27:EF:45:88)_________________

 = ETHERNET                                                         <Show>

 = IPv4 CONFIGURATION <Automatic>                                   <Show>
 = IPv6 CONFIGURATION <Ignore>                                      <Show>

 [X] Automatically connect
 [X] Available to all users

                                                               <Cancel> <OK>
```

Device enp0s7 using Static IP



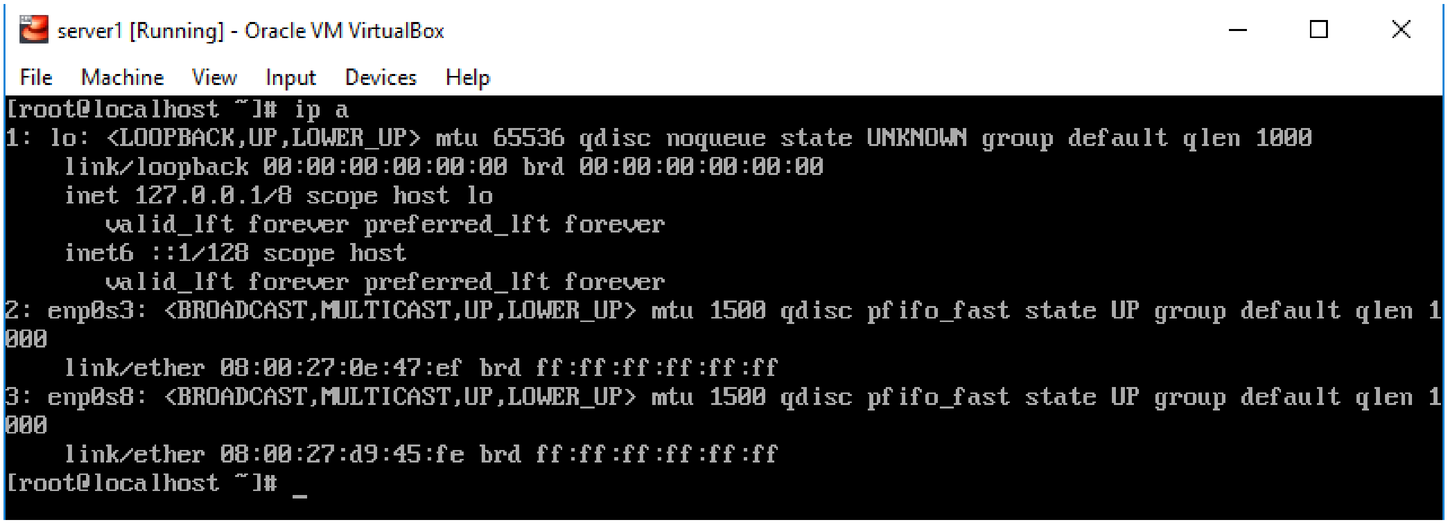
```
                              Edit Connection

        Profile name enp0s8_________________________________________
              Device enp0s8 (08:00:27:C1:1E:90)_____________________

 = ETHERNET                                                      <Show>

 ■ IPv4 CONFIGURATION <Manual>                                   <Hide>
 |           Addresses 192.168.56.45/24_________ <Remove>
 |                     <Add...>
 |             Gateway 192.168.56.1_____________
 |         DNS servers <Add...>
 |      Search domains <Add...>
 |
 |             Routing (No custom routes) <Edit...>
 |  [ ] Never use this network for default route
 |  [ ] Ignore automatically obtained routes
 |  [ ] Ignore automatically obtained DNS parameters
 |
 |  [X] Require IPv4 addressing for this connection
 └

 = IPv6 CONFIGURATION <Ignore>                                   <Show>

 [X] Automatically connect
 [X] Available to all users

                                                          <Cancel> <OK>
```

Directly Edit Configuration Files

This is the method I prefer, simply because it is faster (and easier to check all settings) for me. Use whichever method you prefer.

You can use the network information from the previous command, or you can use the **ip** command.

```
>#  ip a
```



server1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

```
[root@localhost ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1
000
    link/ether 08:00:27:0e:47:ef brd ff:ff:ff:ff:ff:ff
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1
000
    link/ether 08:00:27:d9:45:fe brd ff:ff:ff:ff:ff:ff
[root@localhost ~]# _
```

While this provides more information than the **nmcli** command, it is not nearly as simple and clean an output.

The configuration files which need to be edited can be found in the `/etc/sysconfig/network-scripts` folder. The naming pattern is **`ifcfg-DEVICENAME`**, where DEVICENAME is the name of the device.

I prefer the vi editor. If you prefer another editor, you are wrong. The only editor worth using is vi. All others are imposters to the benevolence and glory that is vi.

```
>#  vi  /etc/sysconfig/network-scripts/ifcfg-DEVICENAME
```

```
*i <return> to insert text
<esc> to finish inserting text
*:wq <return> to save and quit
```

I have noted the necessary changes in **bold green**, and comments in CCP (Consolas Carrot Poop).

DHCP Example

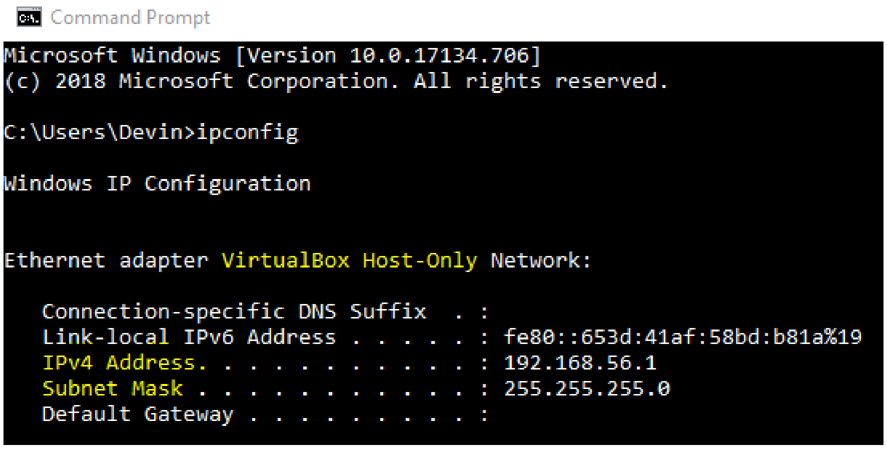
```
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=dhcp         # Use DHCP to assign the IP
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPVC6NIT=no            # Do not initialize IPV6
IPVC6_AUTOCONF=no      # Do not auto configure IPV6
IPVC6_DEFROUTE=no      # IPV6 should not be the default route
IPVC6_FAILURE_FATAL=no # Ignore IPV6 Failures
NAME=enp0s3
UUID=the universally unique id of your device # Do not change
DEVICE=enp0s3
ONBOOT=yes             # Enable interface when booting
```

Static Example

The IP address, netmask, and gateway for the example below are from the VM instance I am using for writing this document, and are unlikely to work in your environment. You can get the correct values from your VM hosting OS. If you are using VirtualBox on Windows, you can get this information from the command prompt:

```
C:\Users\You> ipconfig
```

Look for VirtualBox Host-Only Network. Use the IPv4 Address for your GATEWAY, and the Subnet Mask for your NETMASK. Assign any IPADDR you wish (as long as it is valid for the Mask) and you should be good to go.



```
Microsoft Windows [Version 10.0.17134.706]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Devin>ipconfig

Windows IP Configuration


Ethernet adapter VirtualBox Host-Only Network:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::653d:41af:58bd:b81a%19
   IPv4 Address. . . . . . . . . . . : 192.168.56.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :
```

```
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=static        # Assign a Static IP address
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPVC6NIT=no             # Do not initialize IPV6
IPVC6_AUTOCONF=no       # Do not auto configure IPV6
IPVC6_DEFROUTE=no       # IPV6 should not be the default route
IPVC6_FAILURE_FATAL=no # Ignore IPV6 Failures
NAME=enp0s8
UUID=the universally unique id of your device # Do not change
DEVICE=enp0s8
ONBOOT=yes              # Enable interface when booting
IPADDR=192.168.56.45    # IP address
NETMASK=255.255.255.0   # Subnet Mask
GATEWAY=192.168.56.1    # Default Gateway
```

Verify Changes and Restart Network

Save your changes and exit back to the command prompt. You can verify your changes by displaying the contents of the config files (where DEVICENAME is the name listed from the previous **nmtui** command).

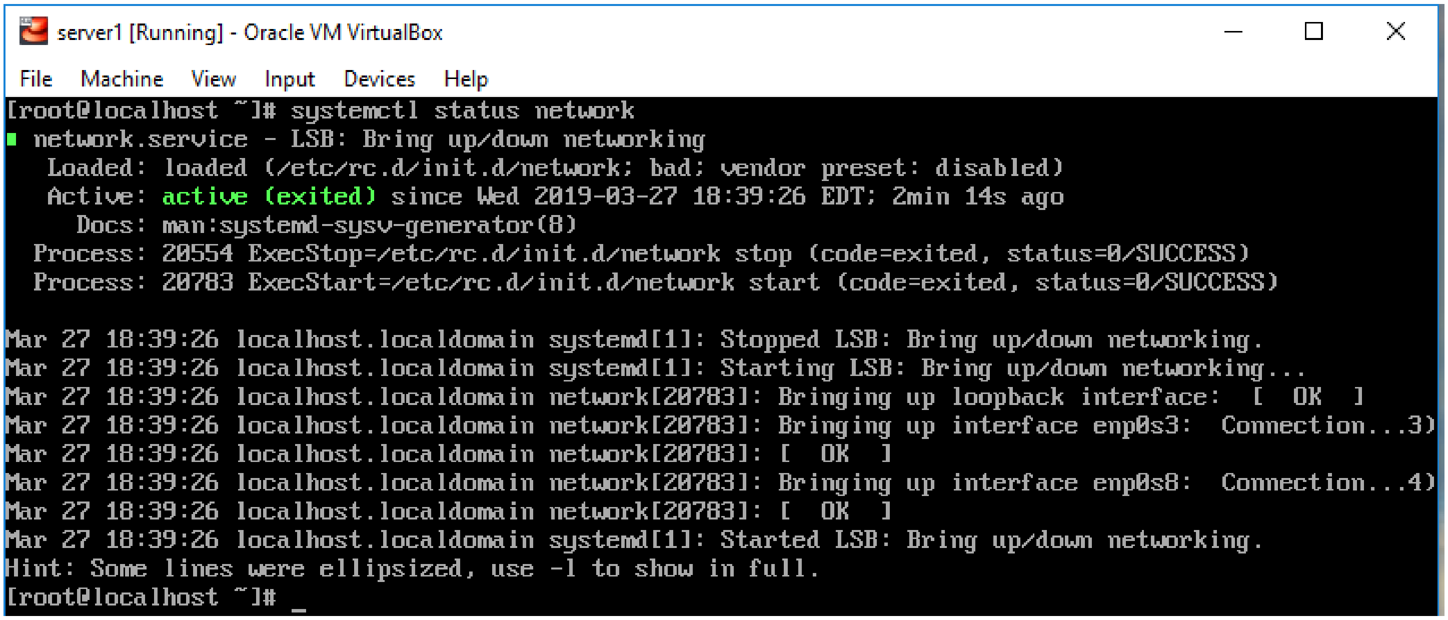
```
>#  cat /etc/sysconfig/network-scripts/ifcfg-DEVICENAME
```

Use systemctl to restart the network:

```
>#  systemctl restart network
```

You can also use systemctl command to check the status of the network:

```
>#  systemctl status network
```



server1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

```
[root@localhost ~]# systemctl status network
● network.service - LSB: Bring up/down networking
   Loaded: loaded (/etc/rc.d/init.d/network; bad; vendor preset: disabled)
   Active: active (exited) since Wed 2019-03-27 18:39:26 EDT; 2min 14s ago
     Docs: man:systemd-sysv-generator(8)
  Process: 20554 ExecStop=/etc/rc.d/init.d/network stop (code=exited, status=0/SUCCESS)
  Process: 20783 ExecStart=/etc/rc.d/init.d/network start (code=exited, status=0/SUCCESS)

Mar 27 18:39:26 localhost.localdomain systemd[1]: Stopped LSB: Bring up/down networking.
Mar 27 18:39:26 localhost.localdomain systemd[1]: Starting LSB: Bring up/down networking...
Mar 27 18:39:26 localhost.localdomain network[20783]: Bringing up loopback interface:  [  OK  ]
Mar 27 18:39:26 localhost.localdomain network[20783]: Bringing up interface enp0s3:  Connection...3)
Mar 27 18:39:26 localhost.localdomain network[20783]: [  OK  ]
Mar 27 18:39:26 localhost.localdomain network[20783]: Bringing up interface enp0s8:  Connection...4)
Mar 27 18:39:26 localhost.localdomain network[20783]: [  OK  ]
Mar 27 18:39:26 localhost.localdomain systemd[1]: Started LSB: Bring up/down networking.
Hint: Some lines were ellipsized, use -l to show in full.
[root@localhost ~]# _
```

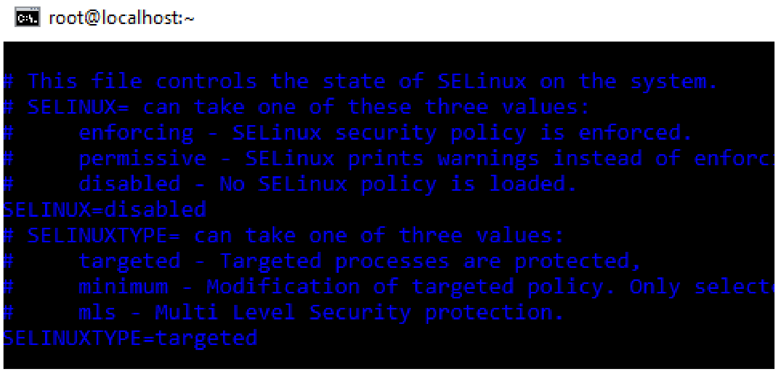
SSH (Secure Shell) - Part 1

SSH Clients

If you are using a Mac or Linux to perform this configuration, the SSH command is included as part of the operating system, because real computers have this. Microsoft finally decided to include this standard command with the release of Windows 10. If you are using a previous version of Windows, you should probably bang your head against the desk in frustration. Assuming you cannot upgrade, you will need to install a 3rd party SSH client in order to proceed. I recommend Putty (available at https://www.putty.org), as it is a simple and easy to use tool that JFW. (If you don't know what JFW means you obviously have never been to one of my sessions. https://www.urbandictionary.com/define.php?term=JFW)

SSH on Windows 10

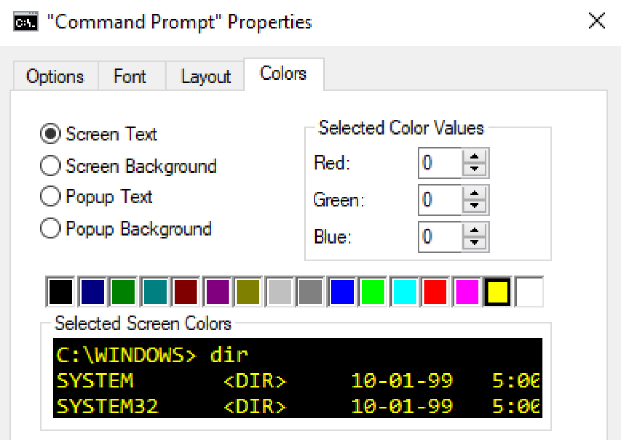
I have noticed some quirky behavior with the SSH command on Windows 10, specifically when starting an editor in the client. The command window font color will sometimes change, making it nearly unusable.



```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforc
#     disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of three values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only select
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

Fortunately, this is easy to change. Right-click on the top bar of the window and select **Properties**.

In the displayed Properties dialog, select the Colors tab, then specify the color combination you want and click the **OK** button.



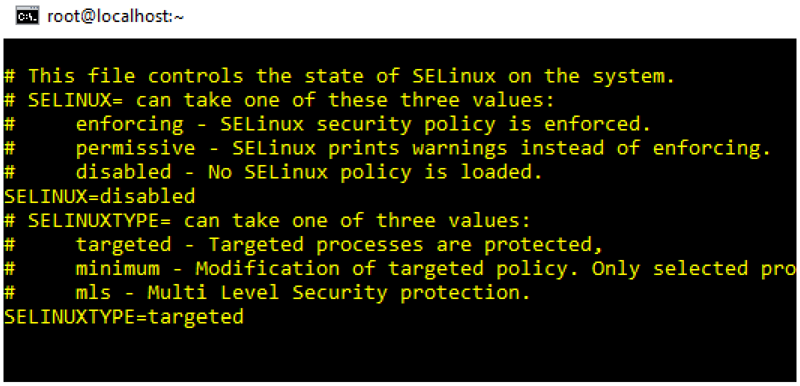
● Screen Text
○ Screen Background
○ Popup Text
○ Popup Background

Selected Color Values
Red: 0
Green: 0
Blue: 0

Selected Screen Colors

```
C:\WINDOWS> dir
SYSTEM         <DIR>       10-01-99    5:0€
SYSTEM32       <DIR>       10-01-99    5:0€
```

Once you have done that the window color will change to the colors you specified.



```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of three values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only selected pro
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

SSH and CentOS7

The CentOS 7 Minimal installation includes the SSHD service and initial configuration information; which makes things much easier than with previous distributions. There is no need to manually install SSH. There is, however, some configuration that we need to do. Ironically, this all starts with opening an SSH session to your server by issuing the following command:

```
C:\Users\You> ssh root@your.server.host.ipaddress.or.fqdn
```

Enter the root password log in and proceed.

User Accounts

Without going into too much unnecessary detail, all Linux user accounts require a minimum of two things: a username, and a primary group with which the user account is associated. Your Domino server will need to run using a Linux account, which means you need to decide upon a group name and a user name for your Domino server. This can be as simple as "servergroup" and "server", or (if you follow IBM's documentation), "Notes" and "Notes". I suggest using your primary organization name for the group, and the server name for the user. This makes it easy to keep track of when setting up multiple servers for an organization. For the rest of this document I will refer to these as groupname and username.

NEVER run Domino AS root

Create the user account for your Domino server as follows:

Create a Group

```
>#  groupadd groupname
```

Add the user

Use the useradd command to add the user. I know the spacing in the example is weird, but it works.

```
-g initial group to which the user should be added
-s shell for the user when signing in.  User scripts are here.
-d home directory
-m make the user
>#  useradd -ggroupname -s/bin/bash -d/home/username -m username
```

Create a password for the user

```
>#  passwd username
```

SSH (Secure Shell) - Part 2

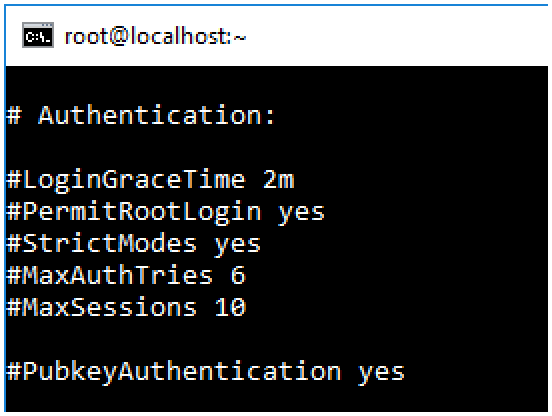
Now that you have added a user account, it is time to configure the SSH daemon to provide a bit more security for your server. As with many other topics in this document, there is massive amount of information and context available that I'm not going to go into. What follows is the absolute bare minimum SSH configuration changes you should make.

Configure and verify SSH

Edit the sshd_**config** file using the greatest editor in the universe

```
>#  vi  /etc/ssh/sshd_config
```

```
*/ searchtext <return> to search for text
```



```
# Authentication:

#LoginGraceTime 2m
#PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes
```

Find the line **#PermitRootLogin yes** and change it to **no**, add additional settings for **AllowUsers** and **AllowGroups**, then save and close the file:

```
PermitRootLogin no
AllowUsers username
AllowGroups groupname
```

Use systemctl to restart the SSHD daemon:

```
>#  systemctl  restart  sshd
```

Start another SSH Session

This is very important, and I cannot stress it enough: **DO NOT CLOSE YOUR CURRENT SSH SESSION**. If you have mis-configured the sever you will not be able to get back in to correct things. So, to be clear, attempt to start a NEW SSH Session for root, and enter your password This should fail.

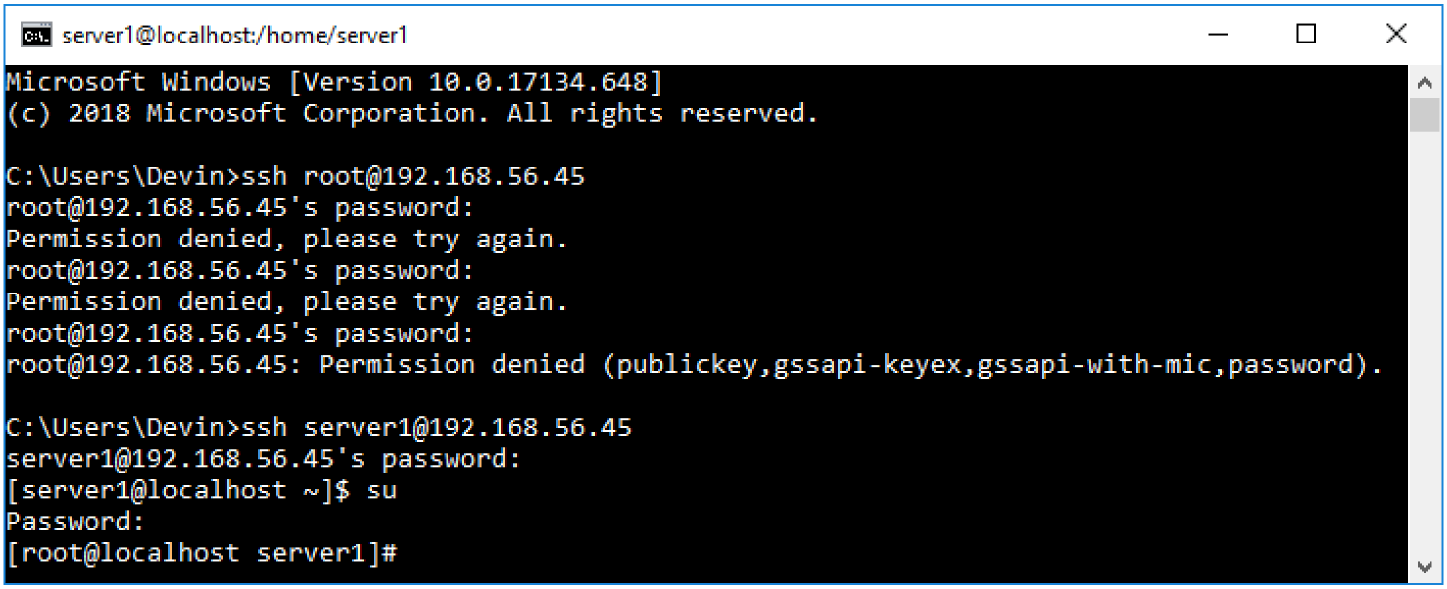
```
C:\Users\You> ssh root@your.server.host.ipaddress.or.fqdn
```

Assuming your login attempt fails (hooray for fail fast, fail early testing), now attempt to start an SSH session using the user credentials you created.

```
C:\Users\You> ssh username@your.server.host.ipaddress.or.fqdn
```

Once you have successfully logged in, use the **su** command and then enter the root password to change to the root user.

```
># su
```



```
Microsoft Windows [Version 10.0.17134.648]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Devin>ssh root@192.168.56.45
root@192.168.56.45's password:
Permission denied, please try again.
root@192.168.56.45's password:
Permission denied, please try again.
root@192.168.56.45's password:
root@192.168.56.45: Permission denied (publickey,gssapi-keyex,gssapi-with-mic,password).

C:\Users\Devin>ssh server1@192.168.56.45
server1@192.168.56.45's password:
[server1@localhost ~]$ su
Password:
[root@localhost server1]#
```

If you cannot login using the new SSH Session, use the first window root session to make the necessary corrections until you can open a new session.

Install Required Packages

The following packages are required for Domino 10 on CentOS. Use the yum command to install them.

- ntp: network time protocol service
- perl: open source general-use interpreted scripting language
- bc: an arbitrary precision numeric processing language

```
>#  yum -y install ntp perl bc
```

Start and enable the cron service

```
>#  systemctl start crond
>#  systemctl enable crond
>#  systemctl status crond
```

Turn off and disable SELinux

Security Enhanced Linux (SELinux) is enabled by default on CentOS 7, and is incompatible with Domino. It needs to be disabled.

```
>#  vi /etc/selinux/config
```

Change the setting to **SELINUX=disabled** and save and close the file. Then set SELINUX enforcing mode to disabled.

```
>#  setenforce 0
```

Enable and activate the time service

The Network Time Protocol service needs to be enabled.

```
>#  ntpdate pool.ntp.org
>#  systemctl start ntpd
>#  systemctl enable ntpd
>#  systemctl status ntpd
```

Firewall Configuration

NOTE: THIS IS THE MINIMAL configuration. A complete set of firewall rules must be configured if your server will be publicly accessible - and is beyond the scope of this document. DO NOT RELY solely on this configuration.

Centos 7 comes, by default, configured to use FirewallD for firewall security and management. FirewallD is dynamically managed firewall with support for network / firewall zones that define the trust level of network connections or interfaces. There is a very nice write-up at linode community for configuring FirewallD on CentOS.

That being said, **I hate FirewallD for servers.** Most of the enhanced security features, zones, and services available with FirewallD are more suited for a desktop / workspace environment, and I'm not willing to put up with the massive headaches that come with dealing with FirewallD.

I prefer iptables; what follows are instructions for **replacing FirewallD with iptables**.

Install iptables

```
>#  yum -y install iptables-services
```

Ensure the ports required by the Domino server are open

Common Ports (for reference, only open the ports you will use)

20	FTP (File Transfer Protocol transfer)
21	FTPC (File Transfer Protocol Command)
22	SSH (Secure SHell)
389	LDAP (Lightweight Directory Access Protocol)
636	LDAPS (Lightweight Directory Access Protocol over SSL)
1352	NRPC (IBM Notes/Domino RPC)
80	HTTP (Hypertext Transfer Protocol)
443	HTTPS (Hypertext Transfer Protocol over SSL)
25	SMTP (Simple Mail Transfer Protocol

- **143** IMAP (Internet Message Access Protocol) -only if you will have IMAP mail clients
- **220** IMAPV3 (Internet Message Access Protocol Version 3) -only if you will have IMAP mail clients
- **993** IMAPS (Internet Message Access Protocol over SSL) -only if you will have IMAP mail clients
- **110** POP3 (Post Office Protocol Version 3) - only if you will have POP3 mail clients
- **995** POP3S (Post Office Protocol Version 3 over SSL) - only if you will have POP3 mail clients
- **8585** Used by Domino Remote Server Setup

In the previous version of this guide I recommended installing and using Webmin. I have since changed my mind. If you want to use Webmin go ahead and follow the instructions found in that document. If you just want to get up and running as quickly as possible, then just manually configure the firewall.

Edit iptables using vi

```
># vi /etc/sysconfig/iptables
```

Find the lines

```
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
```

Change default INPUT / FORWARD policies from ACCEPT to DROP

```
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
```

Find the line:

```
-A INPUT -j REJECT --reject-with icmp-host-prohibited
```

The lines immediately prior to this line identify the ports which are to be opened in the firewall. It is best to keep everything closed and only open the specific ports that you want opened. There is by default an entry to open port 22 (the SSH protocol). Make sure that you keep port 22 open (either by leaving the line in place or by using the line in the following content), otherwise you will effectively lock yourself out of your server. The SSHD service can run and listen forever, but if you close the port you will never get back in. Consider yourself fairly warned.

The following content can be cut and pasted into your vi editor (after hitting “i” to enable insert mode). The green lines beginning with hashtags are comment lines to identify the protocol that each subsequent line is specifying to open in the firewall. If you don’t want a particular protocol, then do not add it. You might already have a line for SSH, you can leave that there.

```
# SSH
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
# LDAP
-A INPUT -m state --state NEW -m tcp -p tcp --dport 389 -j ACCEPT
# LDAPS
-A INPUT -m state --state NEW -m tcp -p tcp --dport 636 -j ACCEPT
# NRPC
-A INPUT -m state --state NEW -m tcp -p tcp --dport 1352 -j ACCEPT
# HTTP
-A INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEPT
# HTTPS
-A INPUT -m state --state NEW -m tcp -p tcp --dport 443 -j ACCEPT
# SMTP
-A INPUT -m state --state NEW -m tcp -p tcp --dport 25 -j ACCEPT
# Domino Remote Server Setup
-A INPUT -m state --state NEW -m tcp -p tcp --dport 8585 -j ACCEPT
```

If you want to only allow SSH from a specific IP address, change the SSH line to:

```
-A INPUT -s your.allowed.ip.address/24 -m state --state NEW -p  tcp
--dport 22 -j A
```

If you want to allow Webmin clients to access, you will need to open a port:

```
# WEBMIN
-A INPUT -m state --state NEW -m tcp -p tcp --dport 10000 -j ACCEPT
```

Save and close the file. You can use the cat command to verify your changes:

```
>#  cat /etc/sysconfig/iptables
```

You now have a choice to make. If you are going to be using IPv6 addressing, then you will need to similar changes as you made to the iptables file.   If you don’t want to use IPv6 then you will need to disable IPv6. You can either edit your systemctl configuration, sshd configuration, and various other configuration files to ensure that IPv6 is properly disabled (and your server still works without getting all wonky), or you can just use your ip6tables firewall configuration to close all the IPv6 ports.

Edit ip6tables using vi

```
>#  vi /etc/sysconfig/ip6tables
```

What follows is the default ip6tables file with my changes applied in **bold green**. Apply the changes to your file.

```
# sample configuration for ip6tables service
# you can edit this manually or use system-config-firewall
# please do not ask us to add additional ports/services to this default
configuration
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state RELATED,ESTABLISHED -j DROP
-A INPUT -p ipv6-icmp -j DROP
-A INPUT -i lo -j DROP
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j DROP
-A INPUT -d fe80::/64 -p udp -m udp --dport 546 -m state --state NEW -j DROP
-A INPUT -j REJECT --reject-with icmp6-adm-prohibited
-A FORWARD -j REJECT --reject-with icmp6-adm-prohibited
COMMIT
```

Test your changes

```
>#  sh -c ‘iptables-restore -t < /etc/sysconfig/iptables’
>#  sh -c 'ip6tables-restore -t < /etc/sysconfig/ip6tables'
```

If you have entered everything correctly, then you will get absolutely zero response when entering the sh commands. That is a good thing.

Shut down FirewallD and Start iptables

Enter the following as one single command:

```
>#  systemctl stop firewalld && systemctl start iptables;
systemctl start ip6tables
```

Verify iptables and ip6tables

```
>#  iptables -S
>#  ip6tables -S
```

SSH from another console

While this might seem a bit redundant, this is where you want to be absolutely certain you can get back into your server. Open another console window and try to ssh to your server.

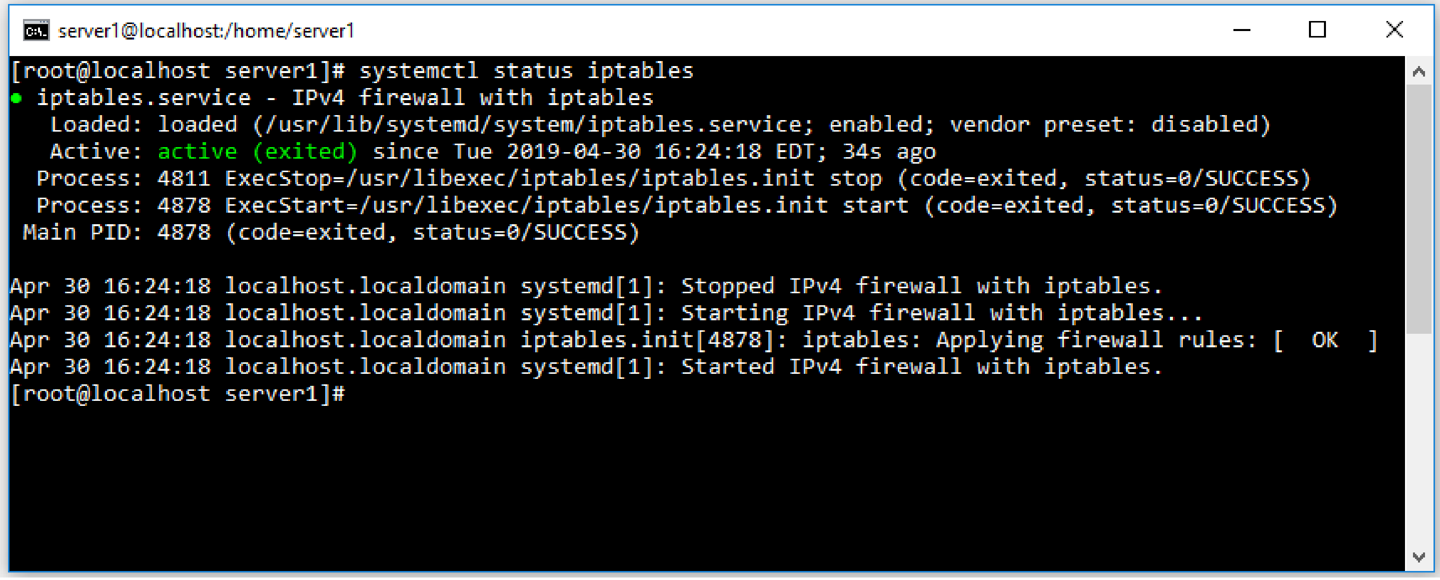
```
C:\Users\You> ssh username@your.server.host.ipaddress.or.fqdn
```

If you are unable to login then you need to go back and correct something. Otherwise we can proceed to remove FirewallD from the system.

Disable FirewallD and Enable iptables

```
>#  systemctl disable firewalld
>#  systemctl mask firewalld
>#  systemctl enable iptables
>#  systemctl enable ip6tables
>#  systemctl status iptables
>#  systemctl status ip6tables
```

The status commands should give a response similar to the following. If you see "Active: active (exited) since ...." that means everything is working normally. (Yes, I realize it can be confusing).



```
[root@localhost server1]# systemctl status iptables
● iptables.service - IPv4 firewall with iptables
   Loaded: loaded (/usr/lib/systemd/system/iptables.service; enabled; vendor preset: disabled)
   Active: active (exited) since Tue 2019-04-30 16:24:18 EDT; 34s ago
  Process: 4811 ExecStop=/usr/libexec/iptables/iptables.init stop (code=exited, status=0/SUCCESS)
  Process: 4878 ExecStart=/usr/libexec/iptables/iptables.init start (code=exited, status=0/SUCCESS)
 Main PID: 4878 (code=exited, status=0/SUCCESS)

Apr 30 16:24:18 localhost.localdomain systemd[1]: Stopped IPv4 firewall with iptables.
Apr 30 16:24:18 localhost.localdomain systemd[1]: Starting IPv4 firewall with iptables...
Apr 30 16:24:18 localhost.localdomain iptables.init[4878]: iptables: Applying firewall rules: [  OK  ]
Apr 30 16:24:18 localhost.localdomain systemd[1]: Started IPv4 firewall with iptables.
[root@localhost server1]#
```

[OPTIONAL] Edit the Hosts file and add the hostname

```
>#  vi  /etc/hosts
```

Add the information for your server to the bottom of the file. Do not change the loopback 127.0.0.1 unless you really know what you are doing and have a very good reason to do so. Changing the loopback hosts entry can cause "very bad things" to happen with your server's network connections.

```
127.0.0.1   localhost   localhost.localdomain localhost4

localhost4.localdomain4

::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
your.server.ip.address fully.qualified.host.name short name
```

Make sure everything is up to date.

Use yum to update installed content.

```
>#  yum  -y update
```

Domino Specific Configuration

Disable Conflicting Services

There are three specific services that conflict with Domino. Depending upon the version of CentOS these may have been automatically installed. They need to be disabled and removed. These services are httpd, sendmail, and postfix. Follow the subsequent instructions for each of these services.

Check the status of the service

```
>#  systemctl status servicename
```

If the service is operational you will need to kill it.

```
>#  systemctl stop servicename
```

Then disable and remove the service.

```
>#  systemctl disable servicename
>#  systemctl mask servicename
```

Set File Handles

Increase the number of file handles available for use.

```
>#  ulimit -n 65535
>#  vi /etc/security/limits.conf
```

Add the following lines to the end of the file, where ***username*** is the name of user you created for running Domino:

```
  username    soft     nofile    65535
  username    hard     nofile    65535
```

Reboot the server

You need to reboot the server at this point.

```
>#  reboot now
```

Log in via SSH

After the server has rebooted, log in again via SSH and change to
Super User (root)

```
>#  su (enter password when prompted)
```

Allow Domino to tune the Linux kernel

Use the ***export*** command to set the operating system variable.

```
>#  export DOMINO_LINUX_SET_PARMS=1
```

Use either ***vi*** or the ***echo*** command to add the export command to the end of the `/home/notes/.bashrc` file:

```
>#  vi /home/username/.bashrc
```

Add the above export command to the end of the file and save it.

-- OR –

```
>#  echo -e "\nexport DOMINO_LINUX_SET_PARMS=1" >>
/home/username/.bashrc
```

Create the directory for your Domino server

You need to decide where to put your Domino Data. If you followed my suggestion when setting up the partitions for your server this will be very easy.

```
>#  mkdir /domino/servername/dominodata -p
```

Verify the directory exists and make note of this directory, you will need it during installation.

```
>#  cd /directory path you just created
```

Get Install File

Now you need to get the Domino installation file onto your newly created server so you can install it. There are a number of ways to do this, I strongly recommend you use SCP (Secure Copy Protocol).   If you are using Linux, Mac OSX, or Windows 10 this command is built into the OS. If you are using a previous version of Windows, you will need to download and install an SCP Tool. I recommend WinSCP.

Use either WinSCP or command line to upload your Domino Installation media to your new server (place it in your server user's home directory for now).

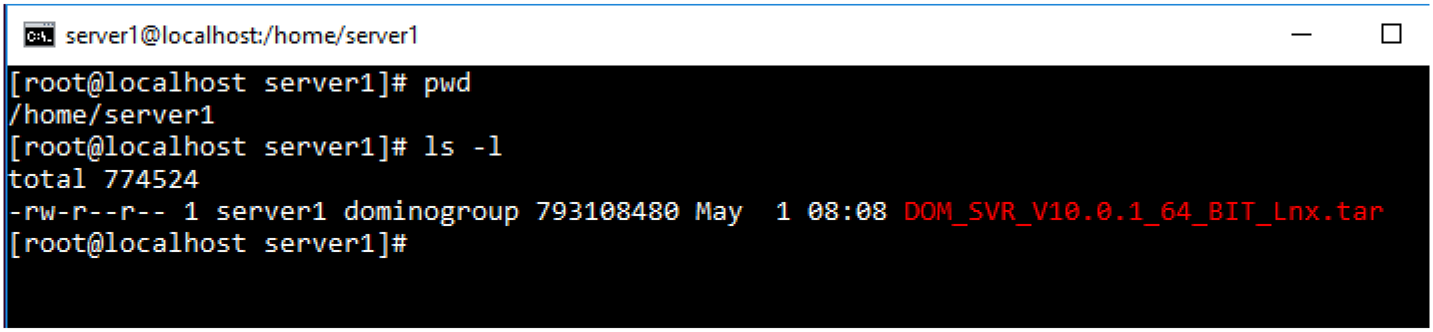
If using command line, change directory to the directory containing your installation media and use the SCP command to upload it to your server.

```
C:\Users\You> scp DOM_SVR_V10.0.1_64_BIT_Lnx.tar
username@your.server.host.ipaddress.or.fqdn:~
```

Do not forget the colon tilde (:~) at the end. That tells the SCP command to place the file in the home folder of the user.

Go back to your server console, switch to your user's directory, and verify the file is there

```
># cd /home/username
># ls -l
```



server1@localhost:/home/server1

```
[root@localhost server1]# pwd
/home/server1
[root@localhost server1]# ls -l
total 774524
-rw-r--r-- 1 server1 dominogroup 793108480 May  1 08:08 DOM_SVR_V10.0.1_64_BIT_Lnx.tar
[root@localhost server1]#
```

Verify File and Unpack

Use the tar command to check the file.

```
-t Table of contents.  List all the files contained in the tar file.
-v Verbose output.
-f Use the filename from the argument parameters.
-x Extract or restore the file(s)
```

```
># tar -tvf DOM_SVR_V10.0.1_64_BIT_Lnx.tar
```

If there are problems with the file, you will need to delete it and download a clean one. If there are no problems, then go ahead and extract the file contents.

```
># tar -xvf DOM_SVR_V10.0.1_64_BIT_Lnx.tar
```

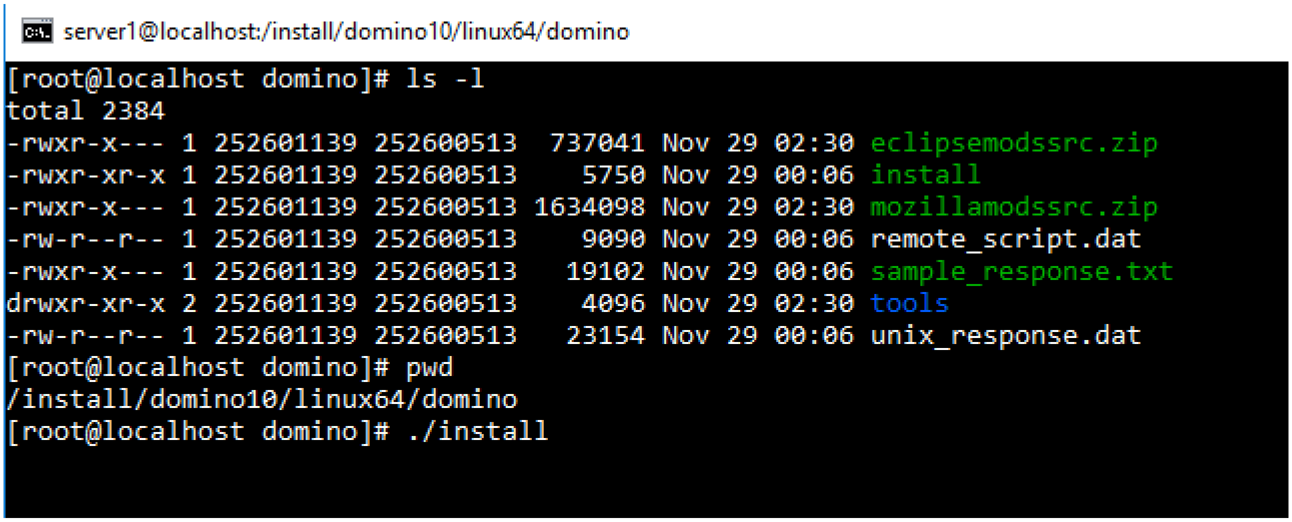
Take a few minutes and go stretch. The server is now ready for installation; you have done a lot of work and deserve a break. You might want to get yourself a tasty beverage or a light snack; or perhaps have a conversation with another human for a few minutes before continuing.

Domino Installation

Find and run the installation

Use the ls command (without the -l argument) to list the directory contents, then navigate down through the folders using the **ls** and **cd** command as needed until you see the **install** file displayed, then run the file:

```
>#  ./install
```



server1@localhost:/install/domino10/linux64/domino

```
[root@localhost domino]# ls -l
total 2384
-rwxr-x--- 1 252601139 252600513  737041 Nov 29 02:30 eclipsemodssrc.zip
-rwxr-xr-x 1 252601139 252600513    5750 Nov 29 00:06 install
-rwxr-x--- 1 252601139 252600513 1634098 Nov 29 02:30 mozillamodssrc.zip
-rw-r--r-- 1 252601139 252600513    9090 Nov 29 00:06 remote_script.dat
-rwxr-x--- 1 252601139 252600513   19102 Nov 29 00:06 sample_response.txt
drwxr-xr-x 2 252601139 252600513    4096 Nov 29 02:30 tools
-rw-r--r-- 1 252601139 252600513   23154 Nov 29 00:06 unix_response.dat
[root@localhost domino]# pwd
/install/domino10/linux64/domino
[root@localhost domino]# ./install
```

Follow Prompts

Answer the questions as follows (pressing <enter> will accept the bracketed choice):

- Do you want to continue installation in console mode? [Yes]
- Welcome to the InstallShield Wizard. [1]
- Read and agree to the license agreement. [1]
- Press 1 for Next, 2 for Previous, 3 to Cancel or 4 to Redisplay [1]
- Install Data Directories Only (*out of scope for this document*): [0]
- Press 1 for Next, 2 for Previous, 3 to Cancel or 4 to Redisplay [1]
- Program Files Directory Name [/opt/ibm/domino]
- Partitioned Server (*out of scope for this document*): [No]
- Press 1 for Next, 2 for Previous, 3 to Cancel or 4 to Redisplay [1]
- Data Files Directory Name [/domino/servername/dominodata] **ENTER THE DIRECTORY PATH YOU CREATED IN DOMINO SPECIFIC CONFIGURATION**
- Press 1 for Next, 2 for Previous, 3 to Cancel or 4 to Redisplay [1]
- User Name [username] **ENTER THE USER NAME YOU CREATED IN USER ACCOUNT & SSH SERVICE**

- Group Name [groupname] **ENTER THE GROUP NAME YOU CREATED IN USER ACCOUNT & SSH SERVICE**
- Press 1 for Next, 2 for Previous, 3 to Cancel or 4 to Redisplay [1]
- Select Server Setup (Manual) [3]
- Press 1 for Next, 2 for Previous, 3 to Cancel or 4 to Redisplay [1]
- Choose the setup type: Domino Utility Server, Domino Messaging Server, Domino Enterprise Server, Customize Domino Server (*Customize is out of scope for this document*)
- Press 1 for Next, 2 for Previous, 3 to Cancel or 4 to Redisplay [1] •
  Installation Summary: Press ENTER to read the text [Type q to quit]
- Press 1 for Next, 2 for Previous, 3 to Cancel or 4 to Redisplay [1]

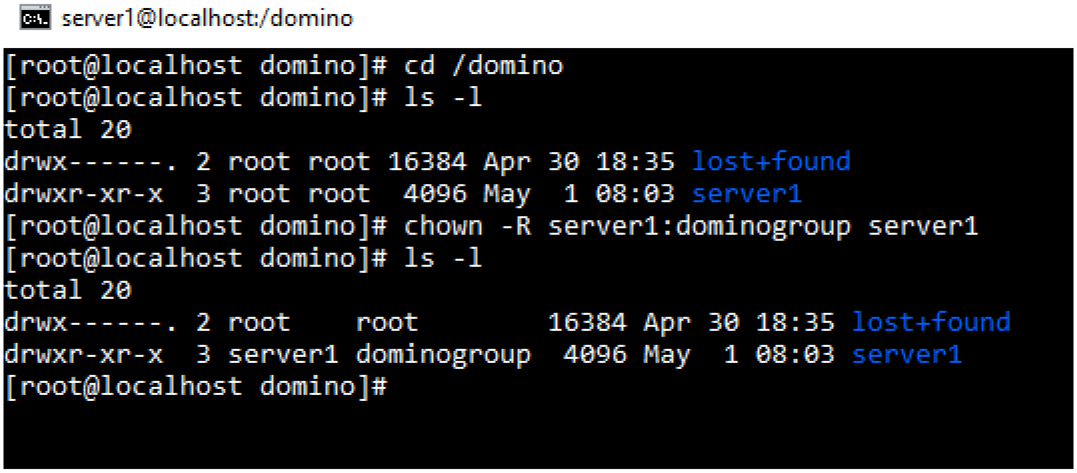
The server installation will now begin.   Once finished, you will be presented with a post-installation instructions page.

- Press 3 to Finish or 4 to Redisplay [3]

Change Directory Ownership

Use the cd command to change to the domino directory and then change ownership of the server’s directory to the username for the server.

```
>$ cd /domino
>$ ls -l
>$ chown -R username:groupname username
>$ ls -l
```



server1@localhost:/domino

```
[root@localhost domino]# cd /domino
[root@localhost domino]# ls -l
total 20
drwx------. 2 root root 16384 Apr 30 18:35 lost+found
drwxr-xr-x  3 root root  4096 May  1 08:03 server1
[root@localhost domino]# chown -R server1:dominogroup server1
[root@localhost domino]# ls -l
total 20
drwx------. 2 root    root        16384 Apr 30 18:35 lost+found
drwxr-xr-x  3 server1 dominogroup  4096 May  1 08:03 server1
[root@localhost domino]#
```

Congratulations. The server is now installed. Now we just need to get it configured and operational.

Domino Setup

Switch ID

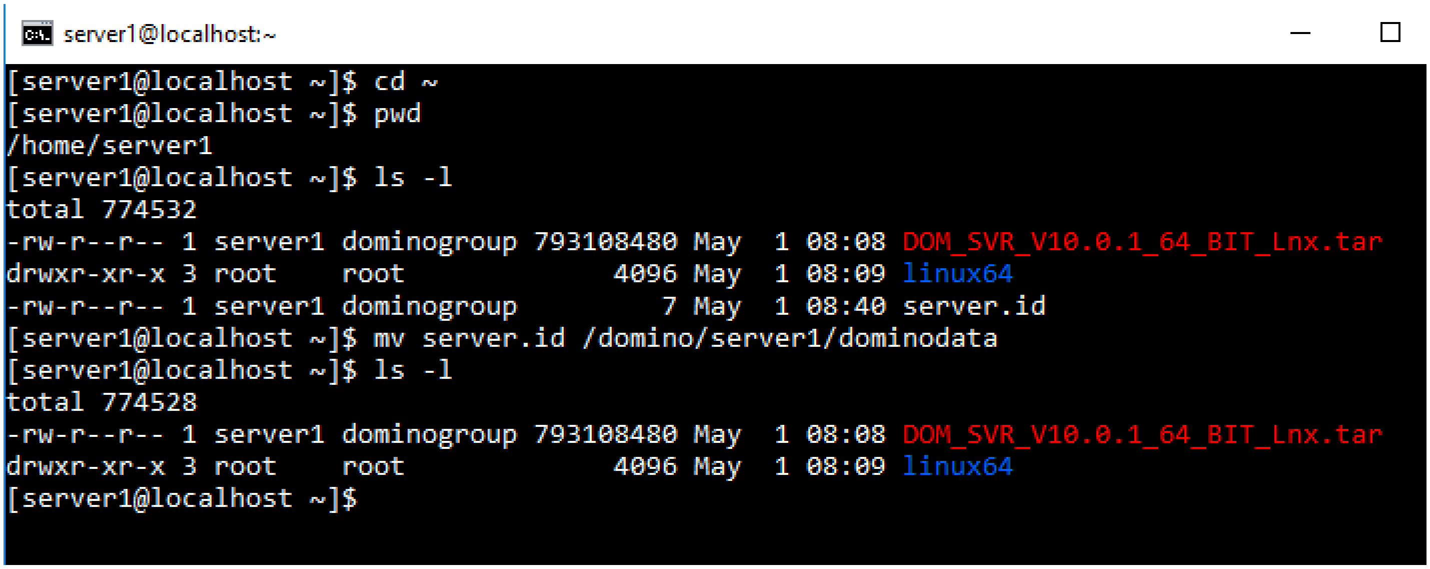
Change back to the user id you used to log into the SSH session:

```
>#  exit
```

Server ID

If this is the first server in your organization, go ahead and skip to the next step. Otherwise you have a bit of administrative work to do. Use your domino administrator to create a new server id file. Once you have the file, you need to put it on your new server. Use SCP in the same manner as the installation files to place the file into your server's home directory. Then use the mv command from the console to move the file from the home directory to the domino directory.

```
>$ cd ~
>$ pwd
>$ ls -l
># mv server.id /domino/servername/dominodata
>$ ls -l
```



```
[server1@localhost ~]$ cd ~
[server1@localhost ~]$ pwd
/home/server1
[server1@localhost ~]$ ls -l
total 774532
-rw-r--r-- 1 server1 dominogroup 793108480 May  1 08:08 DOM_SVR_V10.0.1_64_BIT_Lnx.tar
drwxr-xr-x 3 root    root             4096 May  1 08:09 linux64
-rw-r--r-- 1 server1 dominogroup         7 May  1 08:40 server.id
[server1@localhost ~]$ mv server.id /domino/server1/dominodata
[server1@localhost ~]$ ls -l
total 774528
-rw-r--r-- 1 server1 dominogroup 793108480 May  1 08:08 DOM_SVR_V10.0.1_64_BIT_Lnx.tar
drwxr-xr-x 3 root    root             4096 May  1 08:09 linux64
[server1@localhost ~]$
```

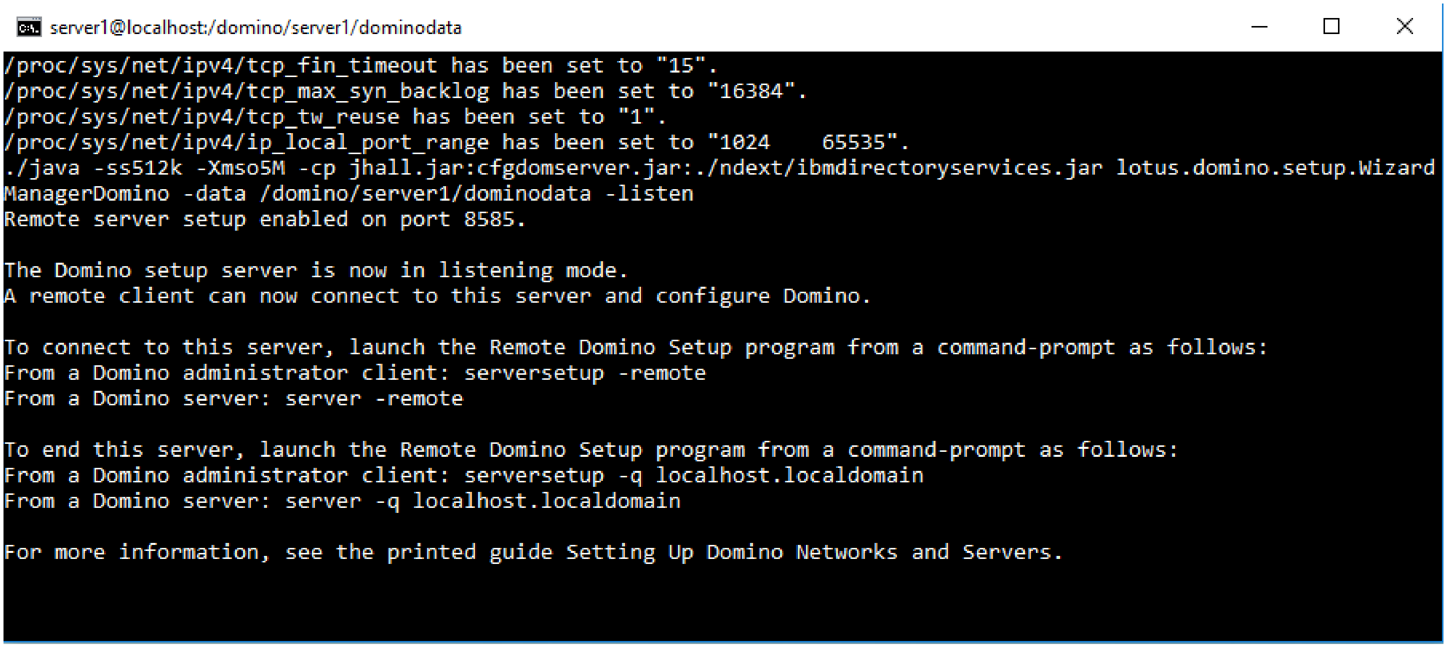
Launch Server in listen mode

Change to the /domino/servername/dominodata folder:

```
>$ cd /domino/servername/dominodata
```

Launch the server and put it into listen mode for remote server setup access:

```
>$ /opt/ibm/domino/bin/server -listen
```



```
/proc/sys/net/ipv4/tcp_fin_timeout has been set to "15".
/proc/sys/net/ipv4/tcp_max_syn_backlog has been set to "16384".
/proc/sys/net/ipv4/tcp_tw_reuse has been set to "1".
/proc/sys/net/ipv4/ip_local_port_range has been set to "1024    65535".
./java -ss512k -Xmso5M -cp jhall.jar:cfgdomserver.jar:./ndext/ibmdirectoryservices.jar lotus.domino.setup.Wizard
ManagerDomino -data /domino/server1/dominodata -listen
Remote server setup enabled on port 8585.

The Domino setup server is now in listening mode.
A remote client can now connect to this server and configure Domino.

To connect to this server, launch the Remote Domino Setup program from a command-prompt as follows:
From a Domino administrator client: serversetup -remote
From a Domino server: server -remote

To end this server, launch the Remote Domino Setup program from a command-prompt as follows:
From a Domino administrator client: serversetup -q localhost.localdomain
From a Domino server: server -q localhost.localdomain

For more information, see the printed guide Setting Up Domino Networks and Servers.
```

Domino Remote Server Setup Utility

On another Machine, start the Domino Remote Server Setup Utility Client. Enter the ip address of your server in the dialog box, and follow the prompts to configure your server. I'm not going to walk you through this, it is very straight-forward, and you have probably already done it dozens of times anyway.

When you finish the configuration, a dialog will ask you if you want to shut down the listening server. Do so, then go back to the SSH console from which you launched the server. It should be back to a command prompt.

Launch Domino

Launch the server, only this time do not add any parameters.

```
>$ /opt/ibm/domino/bin/server
```

Celebrate

Congratulations. Your server is now operational. A quick and simple test to verify is to type in the server's ip address in s browser URL window. You should see the new IBM Domino Start Page. Throw your feet up on the desk, pop open a bottle of your favorite beverage (I suggest anything from **Spanky's Brewery**), and call it a day.

You have done well padawan.
-Devin.

APPENDIX A - STICKY BITS

The Sticky bit problem that has plagued Domino since R7 (7.0.3 specifically) seems to finally have been resolved with the release of version 10; which means this appendix is no longer needed. I have, however, decided to keep it intact. If you run into any weird bindsock issues with your Domino server I suggest you try these instructions first.

Set Sticky bit on Bindsock

When your Domino Server attempts to start any service that needs to **bind** to a **socket** (port); it does so by invoking the ***bindsock*** program.   There is a problem with the installation / setup of IBM Domino Server, in that it does not set the sticky-bit on the bindsock program file. In a Linux environment, a sticky-bit causes an executable file to be run using the credentials of the file owner instead of the invoking user. The proper term for this is ***Set User Identification Attribute***, but everybody just calls it sticky-bit. The bindsock program must be run using the root user's credentials, and therefore must have this sticky-bit set.

If you attempt to start your Domino server immediately after you finish configuring it using the remote server setup you will discover that your HTTP server cannot start, because it cannot bind to port 80. Fortunately, the solution to this is very simple. From the console, change to the Super User and enter the password:

```
>#  su  (enter password when prompted)
```

Change to the domino server program “root” directory:

```
>#  cd /opt/ibm/domino
```

Use the ***find*** command to search for the bindsock program file.

```
>#  find -name 'bindsock'
./notes/90000/linux/bindsock
```

Change to the directory containing the file:

```
>#  cd notes/90000/linux
```

Now use the ls -l command to display the information about bindsock:

```
>#  ls -l bindsock
-r-sr-xr-x 1 root bin 9880 Mar  9 02:41 bindsock
```

This indicates that this bindsock file is owned by the root user, and the bin group. The block of text at the beginning (**`-r-sr-xr-x`**) indicates the type and permissions for the entry. The very first character "-" indicates the type of entry, in this case a normal file. There are several other possible characters with different meanings, such as "d" for directory, or "b" for block device, but for now all we care about is the "-".

What follows after this first character are three sub-blocks of three characters each. Each sub-block defines which specific permissions are enabled for the entry, and the position of the sub-block defines for whom the permissions are to be applied. The first sub-block is for the owner of the entry, the second for the group, and the third for everybody else. There are three standard permissions: ***r**ead*, ***w**rite*, and *e**x**ecute*. We can interpret the result of the ls-l command as follows:

```
-r-sr-xr-x 1 root bin 9880 Mar  9 02:41 bindsock
```

The first block "r-s" tells us that the owner of the entry (in this case root) is allowed to **r**ead the entry, is not allowed to write to the entry (no "**w**"), and can execute the entry using the owner's id (the "s" denotes that Set User Identification Attribute is in force). Normal executable files just have this set as "**x**".

The second block “r-x” tells us that the group for the entry (in this case bin) can **r**ead the entry, is not allowed to write to the entry (no “**w**”), and may e**x**ecute the entry.

The third block “r-x” tells us that everybody else (users who are neither the owner of the file nor a member of the group) can **r**ead the entry, cannot write to the entry (again, no “**w**”), and may e**x**ecute the entry.

These settings are entirely appropriate for most executable files. In this case however they are not correct, because the bindsock program makes changes to system resources (ports) that for security reasons are only allowed to be made by the root user. This is why the HTTP (and possibly LDAP, IMAP, SMTP, etc.) service is failing to start. To correct the problem, we need to add the stick-bit to bindsock. This is accomplished using the chmod utility, followed by an ls-l to verify our changes: ist it again to verify

```
>#  chmod +s bindsock
>#  ls -l bindsock
-r-sr-sr-x 1 root bin 9880 Mar  9 02:41 bindsock
```

This change will now cause all users who are neither the root user, nor members of the bin group, to execute the bindsock using the root user's credentials. Which means our Domino services can now be properly bound to a socket when starting up.

APPENDIX B - ADDITIONAL RESOURCES

There is a lot more configuration you need to do to make your server production ready. Do the appropriate research, read the appropriate blogs, etc.

'nixCraft has a great article about iptables configuration.
https://www.cyberciti.biz/tips/linux-iptables-examples.html

One incredibly useful resource is Daniel Nashed's blog and his wonderful Domino on Unix/Linux Start Script.
https://www.nashcom.de/nshweb/pages/startscript.htm

I cannot imagine running a Production level Domino server without Daniel's script. This is an absolute must-have. If you happen to run into him anywhere, take the time to thank him for his hard work.

Bill Malchisky's blog is a great resource for Linux / Domino content.
http://www.billmal.com/billmal/billmal.nsf

If you are interested in the thoughts and opinions of the person whom I personally consider to be the very best Domino Administrator on the planet, then check out Turtle Partnership's blog:
https://turtleblog.info

If you are looking for a hosting provider, I recommend:

- DigitalOcean: https://www.digitalocean.com – simple hosting
- Prominic: https://prominic.net – Domino Hosting Experts

To be clear, I have received no compensation (or even permission) from these individuals. I mention them because I believe them to be absolute experts at what they do, and I trust their professional advice and judgement.