INSTALLING AND UPGRADING BROCADE SANNAV MANAGEMENT PORTAL V2.1.1 FOR STORAGE ADMINISTRATORS V1.00

NISHAAN DOCRAT

IBM Systems Hardware

Version 1.00 – March 2022

Trademarks

Broadcom, the pulse logo, Brocade, the stylized B logo, Fabric OS, and SANnav are among the trademarks of Broadcom in the United States, the EU, and/or other countries. The term "Broadcom" refers to Broadcom Inc. and/or its subsidiaries.

Red Hat, Inc. Red Hat, and the Red Hat logo are trademarks or registered trademarks of Red Hat, Inc. or its subsidiaries in the United States and other countries. All other trademarks are the property of their respective owners.

Overview

Brocade SANnav is a management, monitoring and alerting tool that provides deep insight into your SAN fabrics and also assists with troubleshooting. As a replacement for Brocade Network Advisor, SANnav was purpose built to transform SAN telemetry data into useful insights like health and performance scores. A key feature is the ability to visualise your SAN and traffic flows. SANnav is split into two products, SANnav Global view which aggregates information from multiple SANnav Management Portals and SANnav Management Portal which is the primary tool to deliver the above features.

For most storage and SAN administrators, this tool is critical in order to manage and maintain large fabrics. It is also responsible for call home which is required to ensure that hardware failures are reported timeously to the switch vendor. The deployment of SANnav Management portal though can be quite daunting for storage administrators who don't have Linux/Unix skills. Whereas it's predecessor offered both Windows and Linux as installation options, SANnav is only available to deploy on Linux. Most storage management teams usually don't have deep platform skills so deploying SANnav usually requires assistance from the Linux administrators in order to successfully deploy the product. This guide is primarily intended for those storage administrators that need to deploy SANnav Management Portal on their own without being completely fluent in Linux. It will also cover how to apply maintenance patches to ensure that your installation is always current.

Hardware and Software Pre-requisites for V2.1.1

The SANnav Management Portal Installation and Migration guide is quite comprehensive and is available at the following URL: https://docs.broadcom.com/doc/SANnav-211x-Install-IG. SANnav Management Portal is available as a standalone installation or as a VMWare OVA image. The OVA image is based on Centos 8 which is end of life (EOL) as of 31/12/21. Apart from that, most organisations prefer to run a fully supported version of Linux like RHEL or SLES. Deploying the OVA is usually frowned on and won't be supported by the relevant Linux teams if allowed. The standalone installation package can be deployed on a supported Linux distribution like RHEL (note, SLES is not a supported SANnav server distribution).

It is important to familiarise yourself with the following sections in order to deploy the correct infrastructure required to run the application and also prior to the deployment of the software.

Table 2: SANnav Installation Customizations
- Overview of the OS and application customisations

Table 3: System and Server Requirements for SANnav Management Portal Installation
- Infrastructure requirements in terms of hardware resources and software versions

NOTE: Whilst the installation guide clearly states that future minor OS revisions are supported but not tested, as of publication of this guide SANnav Management Portal V2.1.1 will NOT install on RHEL 8.4 and 8.5. The installation fails on both of these releases just after installing the bundled version of Docker. Containerd has a segmentation fault and Docker which is dependent on Containerd to operate never starts halting the SANnav installation.

Table 4: Installation Prerequisites
- The pre-requisites for the OS prior to the installation including OS tunables, firewall rules etc.

Installation SANnav Management Portal V2.1.1 on RHEL 8.2

All steps that follow assume that the required hardware resources have been provisioning and you have access to the SANnav virtual machine for deployment.

Using your favourite terminal emulation program SSH into the VM to determine where to transfer the SANnav installation files.

```
Nishaans-MacBook-Pro:~ nishaandocrat$ ssh nishaan@192.168.226.133
nishaan@192.168.226.133's password:

Last login: Fri Mar 18 14:06:55 2022 from 192.168.226.1
[nishaan@sannav ~]$
```

Check the available disk space. In my example /data is the filesystem that is to be used for the installation. It's important to note that the default docker installation directory /var/lib/docker is a separate filesystem that meets the required size guideline of 120GB (the SANnav installation script will check this). Most Linux admins don't like to oversize their root filesystems so a separate Docker installation directory is probably the most common deployment.

```
[nishaan@sannav ~]$ df -h
Filesystem                   Size  Used Avail Use% Mounted on
devtmpfs                     1.9G     0  1.9G   0% /dev
tmpfs                        1.9G     0  1.9G   0% /dev/shm
tmpfs                        1.9G   19M  1.9G   1% /run
tmpfs                        1.9G     0  1.9G   0% /sys/fs/cgroup
/dev/nvme0n1p3                46G  5.1G   41G  12% /
/dev/nvme0n1p1              295M  212M   84M  72% /boot
tmpfs                        376M  1.2M  375M   1% /run/user/42
tmpfs                        376M  5.7M  370M   2% /run/user/1000
/dev/mapper/datavg-dockerlv  120G  889M  120G   1% /var/lib/docker
/dev/mapper/datavg-datalv    455G  3.3G  452G   1% /data
```

Check that you have read and write permissions to the installation directory. In the example below, my standard user account has **rwx** permission (permissions are depicted as OWNER/GROUP/OTHER). Most Linux installations won't allow file transfers even via SSH as root so you will most likely need to transfer it using your own userid.

```
[nishaan@sannav ~]$ cd /data
[nishaan@sannav data]$

[nishaan@sannav data]$ id
uid=1000(nishaan) gid=1000(nishaan) groups=1000(nishaan) context=unconfined_u:unconfined_r:unconfined_t:s0-
s0:c0.c1023
[nishaan@sannav data]$ ls -al
total 0
drwxrwxrwx.  2 root root   6 Mar 18 14:10 .
dr-xr-xr-x. 18 root root 236 Mar 18 14:11 ..
[nishaan@sannav data]$
```

Now we can copy the installation files to the SANnav server. You can use any tool you feel comfortable with like SCP or winSCP or FileZilla. Note, I copied the 2.1.1 base installation file and the 2.1.1.7 patch installation file. The patch can only be applied to the base so you cannot install 2.1.1.7 directly.

```
Nishaans-MacBook-Pro:Downloads nishaandocrat$ scp Portal_2.1.1-distribution.tar.gz
nishaan@192.168.226.133:/data/Portal_2.1.1-distribution.tar.gz
nishaan@192.168.226.133's password:
Portal_2.1.1-distribution.tar.gz
100% 5562MB  29.0MB/s   03:11
Nishaans-MacBook-Pro:Downloads nishaandocrat$

Nishaans-MacBook-Pro:Downloads nishaandocrat$ scp Portal_2.1.1.7.tar.gz
nishaan@192.168.226.133:/data/Portal_2.1.1.7.tar.gz
nishaan@192.168.226.133's password:
Portal_2.1.1.7.tar.gz
100%  958MB  55.0MB/s   00:17
Nishaans-MacBook-Pro:Downloads nishaandocrat$
```

Now log back into the SANnav VM and verify the files transferred correctly.

```
[nishaan@sannav ~]$ cd /data
[nishaan@sannav data]$ ls -al
total 6676844
drwxrwxrwx.  2 root    root             75 Mar 18 14:44 .
dr-xr-xr-x. 18 root    root            236 Mar 18 14:11 ..
-rw-r--r--.  1 nishaan nishaan 5832130513 Mar 18 14:21 Portal_2.1.1-distribution.tar.gz
-rw-r--r--.  1 nishaan nishaan 1004954995 Mar 18 14:44 Portal_2.1.1.7.tar.gz
[nishaan@sannav data]$
```

Now that we have the installation files, we can start with ensuring the pre-requisites are in place. From now on we need to work as the root user so that we can effect any changes that are required to the OS.

```
[nishaan@sannav data]$ su -
Password:
Last login: Fri Mar 18 14:16:19 SAST 2022 on pts/1
[root@sannav ~]#
```

We will start by checking that no previous docker installation exists. If it does, you need to remove the RPM packages using "yum remove <PACKAGE_NAME>". In my case docker is not installed.

```
[root@sannav ~]# rpm -qa | grep docker
[root@sannav ~]#
```

We need to record the interface our IP is set to. We can do this with the "ip a" command. In my case, my IP which is 192.168.226.133 is bound to the ens160 interface. We also need to make note of our IP as we need to confirm that it doesn't conflict with the default docker internal IP ranges.

```
[root@sannav ~]# 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: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:0c:29:f5:d4:f2 brd ff:ff:ff:ff:ff:ff
    inet 192.168.226.133/24 brd 192.168.226.255 scope global dynamic noprefixroute ens160
       valid_lft 978sec preferred_lft 978sec
    inet6 fe80::2f87:7c4e:234:3b8f/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
    link/ether 52:54:00:4f:dc:d1 brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
       valid_lft forever preferred_lft forever
4: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc fq_codel master virbr0 state DOWN group default qlen 1000
    link/ether 52:54:00:4f:dc:d1 brd ff:ff:ff:ff:ff:ff
```

We need to record the interface our IP is set to. We can do this with the "ip a" command. In my case, my IP which is 192.168.226.133 is bound to the ens160 interface. We also need to make note of our IP as we need to confirm that it doesn't conflict with the default docker internal IP ranges.

Now we can check if the pre-requisite RPMs are installed. In my case, they are already installed.

```
[root@sannav ~]# rpm -qa | grep -E "lsof|bind-utils"
lsof-4.91-2.el8.x86_64
bind-utils-9.11.13-6.el8_2.1.x86_64
[root@sannav ~]#
```

If they are not installed, you can use "yum install <PACKAGE_NAME>" to install them. Your Linux administrator would most likely have already setup the installation repositories but if you get any errors, contact them to resolve it.

```
[root@sannav ~]# yum install lsof bind-utils
Updating Subscription Management repositories.
Last metadata expiration check: 0:50:10 ago on Fri 18 Mar 2022 01:56:14 PM SAST.
Package lsof-4.91-2.el8.x86_64 is already installed.
Package bind-utils-32:9.11.13-6.el8_2.1.x86_64 is already installed.
Dependencies resolved.
```

```
Nothing to do.
Complete!
[root@sannav ~]#
```

As per the installation guide, if your docker installation directory is type xfs you need to make sure ftype is set to 1. You can verify the filesystem type by issuing the "mount" command. Your Linux admin has either ensure /var is big enough or you have a separate filesystem created for docker which in my case is /var/lib/docker. The default root filesystem type for a RHEL server is xfs.

```
[root@sannav ~]# xfs_info /var/lib/docker
meta-data=/dev/mapper/datavg-dockerlv isize=512    agcount=4, agsize=7864320 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1
data     =                       bsize=4096   blocks=31457280, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0, ftype=1
log      =internal log           bsize=4096   blocks=15360, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
[root@sannav ~]#
```

Now we are ready to unpack the installation files. You need to navigate to the directory where you transferred them too (via scp) and untar the files. You need to make sure firstly that you umask is set to 0022 (default permissions for new files) prior to unpacking the installation files.

```
[root@sannav ~]# cd /data
[root@sannav data]# umask
0022
[root@sannav data]# umask 0022
[root@sannav data]# umask
0022
[root@sannav data]#
```

Untar the installation files as per the command below.

```
[root@sannav data]# tar -xvzf Portal 2.1.1-distribution.tar.gz
Portal_2.1.1_bld184/jre/
Portal_2.1.1_bld184/jre/man/
Portal_2.1.1_bld184/jre/man/man1/
Portal_2.1.1_bld184/jre/man/ja/
Portal_2.1.1_bld184/jre/man/ja_JP.UTF-8/
Portal_2.1.1_bld184/jre/man/ja_JP.UTF-8/man1/
Portal_2.1.1_bld184/jre/lib/
Portal_2.1.1_bld184/jre/lib/security/
Portal_2.1.1_bld184/jre/lib/security/policy/
Portal_2.1.1_bld184/j
.
.
.
.
Portal_2.1.1_bld184/lib/mybatis-migrations.jar
Portal_2.1.1_bld184/lib/mybatis.jar
Portal_2.1.1_bld184/lib/postgresql.jar
Portal_2.1.1_bld184/conf/postgres/postgresql.conf
Portal_2.1.1_bld184/bin/merge-files.sh
Portal_2.1.1_bld184/bin/readme_split-file.txt
Portal_2.1.1_bld184/bin/readme_merge-files.txt
Portal_2.1.1_bld184/bin/split-file.sh
Portal_2.1.1_bld184/lib/FilesMerger.jar
Portal_2.1.1_bld184/lib/FileSplitter.jar
Portal_2.1.1_bld184/conf/jre/java.security
Portal_2.1.1_bld184/swidtag/www.broadcom.com_SANnav_Management_Portal-2.1.1.swidtag
Portal_2.1.1_bld184/bin/lib/configure-prerequisites.sh
Portal_2.1.1_bld184/bin/lib/copy-templates-files.sh
Portal_2.1.1_bld184/bin/lib/check-system-requirements.sh
Portal_2.1.1_bld184/bin/lib/deploy-docker-stack.sh
Portal_2.1.1_bld184/bin/lib/export-env-util.sh
Portal_2.1.1_bld184/bin/lib/port-util.sh
[root@sannav data]#
```

Verify the unpacked files. This is now your <installation_home> directory as documented in the installation guide. In my case, the installation home directory is /data/Portal_2.1.1_bld184.

```
[root@sannav data]# ls -al
total 6676848
drwxrwxrwx.  3 root    root          102 Mar 18 14:48 .
```

```
dr-xr-xr-x. 18 root     root              236 Mar 18 14:11 ..
-rw-r--r--.  1 nishaan nishaan 1004954995 Mar 18 14:44 Portal_2.1.1.7.tar.gz
drwxr-xr-x. 22 root     root             4096 Mar 18 14:53 Portal_2.1.1_bld184
-rw-r--r--.  1 nishaan nishaan 5832130513 Mar 18 14:21 Portal_2.1.1-distribution.tar.gz
[root@sannav data]#
```

A few more modifications before we start the SANnav installation. The first is to adjust the ulimit value for the user elasticsearch in the /etc/security/limits.conf file. You can check if this already exists issuing "grep elasticsearch /etc/security/limit.conf". If it returns nothing it means you need to add it. You can do it as follows and then check that it was added successfully. Note, make sure you specify ">>" otherwise you will overwrite the file instead of append to it.

```
[root@sannav data]# echo "elasticsearch - nofile 65536" >> /etc/security/limits.conf
[root@sannav data]# cat /etc/security/limits.conf
# /etc/security/limits.conf
#
#This file sets the resource limits for the users logged in via PAM.
#It does not affect resource limits of the system services.
#
#Also note that configuration files in /etc/security/limits.d directory,
#which are read in alphabetical order, override the settings in this
#file in case the domain is the same or more specific.
#That means for example that setting a limit for wildcard domain here
#can be overriden with a wildcard setting in a config file in the
#subdirectory, but a user specific setting here can be overriden only
#with a user specific setting in the subdirectory.
#
.
.
.
.
.
#@faculty        hard    nproc           50
#ftp             hard    nproc           0
#@student        -       maxlogins       4

# End of file
elasticsearch - nofile 65536
[root@sannav data]#
```

The next modification is perhaps the most significant. SANnav uses it's own SSH server to communicate with the switches and directors. This port is currently being used though for your SSH session into the VM. So that means we need to change the port to something else otherwise we won't be able to collect supportsaves and push firmware without using a separate server. You need to communicate this to your Linux administrator to ensure they are aware we need to change the default SSH port. In our case we will change it to port 6022. Use your favourite editor (vi, vim or nano) and edit the /etc/ssh/sshd_config file to change the port to 6022. If you are not familiar with editing on Linux, just do a quick Google search to find a tutorial.

```
[root@sannav data]# vi /etc/ssh/sshd_config

.
.
.
# default value.

# If you want to change the port on a SELinux system, you have to tell
# SELinux about this change.
# semanage port -a -t ssh_port_t -p tcp #PORTNUMBER
#
#Port 22
Port 6022
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_ecdsa_key
.
.
.
```

Check that the port is changed to 6022.

```
[root@sannav data]# grep -i port /etc/ssh/sshd_config
# If you want to change the port on a SELinux system, you have to tell
# semanage port -a -t ssh_port_t -p tcp #PORTNUMBER
```

```
#Port 22
Port 6022
# WARNING: 'UsePAM no' is not supported in Fedora and may cause several
#GatewayPorts no
[root@sannav data]#
```

We need to restart the SSH daemon for this to take effect. Your current SSH session will remain though any new sessions can only be established via port 6022.

```
[root@sannav data]# systemctl restart sshd
Job for sshd.service failed because the control process exited with error code.
See "systemctl status sshd.service" and "journalctl -xe" for details.
[root@sannav data]#
```

Whoops! The SSH daemon did not restart. Let's check why...

```
[root@sannav data]# journalctl -xe
                                       If you want to allow sshd to bind to network port 6022
                                       Then you need to modify the port type.
                                       Do
                                       # semanage port -a -t PORT_TYPE -p tcp 6022
                                           where PORT_TYPE is one of the following: ssh_port_t, vnc_port_t,
xserver_port_t.

                                       *****  Plugin catchall_boolean (7.83 confidence) suggests
******************
.
.
.
```

Ok so we need to modify the SELinux policy to allow us to change the SSH service port. Do that, restart the SSH daemon and check the status of the SSH with systemctl to make sure it is running as per below.

```
[root@sannav data]# semanage port -a -t ssh_port_t -p tcp 6022
[root@sannav data]#

[root@sannav data]# systemctl restart sshd
[root@sannav data]# systemctl status sshd
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2022-03-18 15:05:46 SAST; 8s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
 Main PID: 8043 (sshd)
    Tasks: 1 (limit: 23814)
   Memory: 1.1M
   CGroup: /system.slice/sshd.service
           └─8043 /usr/sbin/sshd -D -oCiphers=aes256-gcm@openssh.com,chacha20-poly1305@openssh.com,aes256-
ctr,aes256-cbc,aes128-gcm@openssh.com,aes>

Mar 18 15:05:46 sannav systemd[1]: Starting OpenSSH server daemon...
Mar 18 15:05:46 sannav sshd[8043]: Server listening on 0.0.0.0 port 6022.
Mar 18 15:05:46 sannav sshd[8043]: Server listening on :: port 6022.
Mar 18 15:05:46 sannav systemd[1]: Started OpenSSH server daemon.
```

You can also check if SSH is listening for new connections on port 6022 with the netstat command.

```
[root@sannav data]# netstat -an | grep LISTEN | grep -v STREAM
tcp        0      0 0.0.0.0:6022         0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:5355         0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:111          0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:53           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:631        0.0.0.0:*               LISTEN
tcp6       0      0 :::6022              :::*                    LISTEN
tcp6       0      0 :::5355              :::*                    LISTEN
tcp6       0      0 :::111               :::*                    LISTEN
tcp6       0      0 :::53                :::*                    LISTEN
tcp6       0      0 ::1:631              :::*                    LISTEN
unix  2      [ ACC ]      SEQPACKET  LISTENING    25446    /run/udev/control
unix  2      [ ACC ]      SEQPACKET  LISTENING    25483    /run/systemd/coredump
l2cap  *              *                 LISTEN       3 0x0000 0x0000    1013       0 SDP
l2cap  f0:18:98:49:9c:9b *              LISTEN      25 0x0000 0x0000     672       0 MEDIUM
l2cap  f0:18:98:49:9c:9b *              LISTEN      27 0x0000 0x0000     672       0 MEDIUM
l2cap  f0:18:98:49:9c:9b *              LISTEN      23 0x0000 0x0000     672       0 MEDIUM
l2cap  f0:18:98:49:9c:9b *              LISTEN      15 0x0000 0x0000    1691    1691 MEDIUM
l2cap  f0:18:98:49:9c:9b *              LISTEN      19 0x0000 0x0000     672       0 LOW
l2cap  f0:18:98:49:9c:9b *              LISTEN      17 0x0000 0x0000     672       0 LOW
l2cap  *              *                 LISTEN       1 0x0000 0x0000     672       0 SDP
```

```
rfcomm f0:18:98:49:9c:9b *              LISTEN          13
rfcomm f0:18:98:49:9c:9b *              LISTEN          12
rfcomm f0:18:98:49:9c:9b *              LISTEN           3
[root@sannav data]#
```

Before you attempt to connect to the SANnav VM on port 6022 to continue the install we need to check if the default RHEL firewall is running. If it is, we need to allow port 6022 otherwise you won't be able to connect.

```
[root@sannav data]# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2022-03-18 14:02:05 SAST; 1h 4min ago
     Docs: man:firewalld(1)
 Main PID: 1176 (firewalld)
    Tasks: 2 (limit: 23814)
   Memory: 26.4M
   CGroup: /system.slice/firewalld.service
           └─1176 /usr/libexec/platform-python -s /usr/sbin/firewalld --nofork --nopid

Mar 18 14:02:04 sannav systemd[1]: Starting firewalld - dynamic firewall daemon...
Mar 18 14:02:05 sannav systemd[1]: Started firewalld - dynamic firewall daemon.
Mar 18 14:02:06 sannav firewalld[1176]: WARNING: AllowZoneDrifting is enabled. This is considered an insecure
configuration option. It will be remo>
[root@sannav data]#
```

If you get output saying firewalld is not running then consider yourself lucky. If no, you need to add the following rules (I've included not just the ones require for the new SSH config but also the ones recommended in the installation guide for SANnav to work through a firewall).

```
[root@sannav data]# firewall-cmd --permanent --add-port=6022/tcp
success
[root@sannav data]# firewall-cmd --permanent --add-service=http
success
[root@sannav data]# firewall-cmd --permanent --add-service=https
success
[root@sannav data]# firewall-cmd --zone=public --add-port=2377/tcp --permanent
success
[root@sannav data]# firewall-cmd --zone=public --add-port=7946/tcp --permanent
success
[root@sannav data]# firewall-cmd --zone=public --add-port=7946/udp --permanent
success
[root@sannav data]# firewall-cmd --zone=public --add-port=4789/udp --permanent
success
[root@sannav data]#
```

Check our new rules before activation.

```
[root@sannav data]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens160
  sources:
  services: cockpit dhcpv6-client http https ssh
  ports: 6022/tcp 2377/tcp 7946/tcp 7946/udp 4789/udp
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

[root@sannav data]#
```

The installation guide asks us to check if the interface bound to the firewall rules is correct (it should be). In our case, we confirmed above ens160 is our public interface so I don't have to change anything. If you do, then you need to issue "firewall-cmd --permanent --zone=public --change-interface=<interface_name>" where interface name matches yours. Reload the new rules so they become active.

```
[root@sannav data]# firewall-cmd --reload
Success
```

Now try and log back in using SSH on port 6022 (don't kill your current session until you can login using port 6022).

```
Nishaans-MacBook-Pro:~ nishaandocrat$ ssh nishaan@192.168.226.133 -p 6022
nishaan@192.168.226.133's password:

Last login: Fri Mar 18 15:24:07 2022
[nishaan@sannav ~]$
```

Now try and log back in using SSH on port 6022 (don't kill your current session until you can login using port 6022). We almost ready.. Just a few more modifications. Let's su back to root and check if we have IPv4 forwarding enabled.

```
Last login: Fri Mar 18 15:24:07 2022
[nishaan@sannav ~]$ su -
Password:
Last login: Fri Mar 18 14:50:24 SAST 2022 on pts/2
[root@sannav ~]#

[root@sannav ~]# /sbin/sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
```

In my case IPv4 forwarding is already enabled. However, if yours comes back as 0 then you need to modify it and also set it to be enabled on reboot. Issue this command to enable it immediately.

```
[root@sannav ~]# /sbin/sysctl net.ipv4.ip_forward
net.ipv4.ip_forward = 1
[root@sannav ~]#
```

To ensure it persists across reboots, edit the /etc/sysctl.conf file again with your preferred editor and add these 2 lines.

```
[root@sannav data]# vi /etc/sysctl.conf
.
.
    # IP Forwarding is enabled for Broadcom SANnav
    net.ipv4.ip forward = 1
.
.
```

Okay now we can just confirm our hostname resolution works. You can find out your hostname by issuing "hostname". We need to ensure it resolves to a valid IP address.

```
[root@sannav ~]# nslookup sannav
Server:         127.0.0.1
Address:    127.0.0.1#53

Name:       sannav
Address: 192.168.226.133

[root@sannav ~]#
```

Note: Curiously on RHEL 8.4 and 8.5 the ipcalc RPM is not loaded as part of the OS. This causes the SANnav install script to continually fail with name resolution issues. Whilst I tested SANnav Management Portal V2.1.1 doesn't work on RHEL 8.4 and 8.5 if you trying this yourself make sure ipcal is loaded (you can issue "yum install ipcalc") to fix that.

Right, we ready to install. Lets go to our <installation_home>/bin directory and see what to do. We need to make sure we root and our umask is 0022 as well.

```
[root@sannav ~]# cd /data
[root@sannav data]# ls
Portal_2.1.1.7.tar.gz  Portal_2.1.1_bld184  Portal_2.1.1-distribution.tar.gz
[root@sannav data]# cd Portal_2.1.1_bld184/
[root@sannav Portal_2.1.1_bld184]# pwd
/data/Portal_2.1.1_bld184
[root@sannav Portal_2.1.1_bld184]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@sannav Portal_2.1.1_bld184]# umask
0022
[root@sannav Portal_2.1.1_bld184]#
```

```
[root@sannav bin]# ls
backuprestore                      lib                                replace-kafka-certificates.sh
split-file.sh                      update-reports-purge-settings.sh
change-docker-subnet.sh             manage-high-granular-data-collection.sh  replace-server-certificates.sh
start-sannav.sh                    update-storage-auto-enclosure-feature.sh
change-ipv4-installation-to-ipv6.sh  manage-sannav-whitelisting.sh      restart-sannav.sh
stop-sannav.sh                     usage-data-collection.sh
check-sannav-status.sh              merge-files.sh                     sannav-management-console.sh
tools
delete-ssh-key.sh                   readme_merge-files.txt             sannav_support_data
trigger-trufos-check-and-renew.sh
diag                               readme_split-file.txt              show-sannav-configurations.sh
uninstall-sannav.sh
install-sannav.sh                   reconfigure-sannav-for-96GB.sh     show-sannav-open-source-software.sh
update-events-purge-settings.sh
[root@sannav bin]#
```

Note: In the <installation_home>/bin directory there is a lib directory that contains the installation pre-check script which is call *check-system-requirements.sh*. If you need to modify it to match your configuration then you need to edit this file. Do this at your own risk however as modifying what Brocade has deemed minimum requirements is dangerous. I had too in order to ensure I could bypass the HW checks for my laptop VM.

Folks.. We good to go... Let's start the SANnav installation. Remember, we have navigated to OUR <installation_home>/bin directory which in my case is /data/ Portal_2.1.1_bld184/bin. We then issue "./install-sannav.sh".

```
[root@sannav bin]# ./install-sannav.sh pwd
Checking ports that must be free.
All the requirements are met, proceeding with SANnav installation.


BROCADE COMMUNICATIONS SYSTEMS LLC
END USER SOFTWARE LICENSE AGREEMENT FOR
Brocade® SANnav Management Portal and SANnav Global View
IMPORTANT: READ THIS CAREFULLY BEFORE INSTALLING, USING OR ELECTRONICALLY ACCESSING THIS PROPRIETARY PRODUCT!

THIS END USER LICENSE AGREEMENT ("Agreement") GOVERNS THE DOWNLOAD, INSTALLATION, USE, POSTING, DISTRIBUTING AND
OTHERWISE MAKING AVAILABLE OF BROCADE SANNAV MANAGEMENT PORTAL AND SANNAV GLOBAL VIEW SOFTWARE, AND ACCOMPANYING
DOCUMENTATION (collectively the "Software"). BY DOWNLOADING, INSTALLING, USING, POSTING, DISTRIBUTING OR OTHERWISE
MAKING AVAILABLE THE SOFTWARE YOU ARE AGREEING TO BE BOUND ON AN ONGOING BASIS BY THE TERMS AND CONDITIONS HEREIN,
WHICH MAY BE UPDATED BY BROCADE FROM TIME TO TIME. IF AT ANY TIME YOU DO NOT AGREE TO ALL OF THE TERMS OF THIS
AGREEMENT, PROMPTLY STOP USE OF THE SOFTWARE AND DESTROY ALL COPIES OF THE SOFTWARE IN YOUR POSSESSION OR CONTROL,
AND CERTIFY IN WRITING TO BROCADE SUCH CESSATION OF USE AND DESTRUCTION.

SINGLE USER LICENSE.  Subject to the terms and conditions of this Agreement and payment of the applicable product
license fees, Brocade Communications Systems LLC ("Brocade") and its suppliers grant to you ("End User") a non-
exclusive, non-transferable, non-assignable, non-sub licensable license to use the Software in object code form
solely for the purpose of managing Brocade storage area networking switches in accordance with the section entitled
"Restrictions" below. Brocade licenses the End User the right to use Software acquired only through an authorized
channel.  Obtaining the Software from an unauthorized source is strictly forbidden and is a violation of this
Agreement.  End User may make backup copies of the Software as may be necessary for End User's lawful use, provided
End User affixes to such copies all trademark, copyright, patent, and notices of other proprietary rights that
appear on the original.

SOFTWARE LICENSES AND LICENSE KEYS.  Brocade Software Transaction Keys and Software Licenses shall only be
purchased from a Brocade authorized distributor or official OEM partner, and Software License Keys shall only be
obtained through Brocade's License Portal or an authorized OEM License Portal.  Brocade Software Licenses and
License Keys shall not be resold, assigned, sub-licensed or otherwise transferred to a third party.

RESTRICTIONS.  Except as otherwise expressly provided in this Agreement or expressly authorized by Brocade, End
User shall have no right, and End User specifically agrees not to, and not to permit third parties to:  (1) copy or
modify the Software; (2) create derivative works based on all or any portion of the Software; (3) remove or modify
any notice of any patent, copyright, trademark or other proprietary rights that appear on or in the Software; (4)
reverse engineer, decompile, translate, disassemble or perform discovery of the source code of all or any portion
of the Software; (5) distribute, disclose, market, lease, resell, assign, loan, sublicense, rent or transfer to any
third party, including End User's customers, interactive or other service; (6) merge the Software into another
product; (7)  install or use multiple instances of the Software in excess of the number Licenses for which the End
User has purchased through Brocade or an authorized channel; (8) disclose to any third party the results of any
testing or performance benchmarks of the Software or other Brocade product without Brocade's prior written consent;
(9) use the software on any device that does not meet the hardware and operating system requirements called out in
the Software release notes; (10) use, download, or make available for use or download, any versions of the Software
that have reached End of Support, as declared by Brocade; (11) sub-license or permit third parties to post or
otherwise make available, or use the Software to provide support for Brocade products without Brocade's express
written consent; (12) use the Software with or for the management of Brocade products that have reached End of
Support, as declared by Brocade; or (13) provide access to any performance metrics or telemetry data made available
from the Software to any third parties or third party software, analytics tools, or services without Brocade's
express written consent.  Use of the Software in violation of these limitations shall be deemed infringement of
Brocade's intellectual property for which Brocade will avail itself of all legal remedies.
```

THIRD PARTY AND OPEN SOURCE SOFTWARE COMPONENTS. Third party and "open source" software components may be provided with the Software pursuant to the terms of the applicable licenses governing the use and distribution of the third party and/or open source software. Open source software is subject to the applicable open source license (e.g., the GNU General Public License available at https://www.gnu.org/licenses/old-licenses/gpl-2.0.en.html) and is not subject to this Agreement. To obtain a copy of the source code and applicable licensing terms for third party components used in the Software, please download the applicable Open Source Code Disclosure documents in the Broadcom Customer Support Portal (CSP). If you do not have a CSP account or are unable to login, please contact your support provider for this information.

Updating and maintaining open source components provided with the Software including obtaining and applying patches or updates to address defects, security vulnerabilities, or Common Vulnerabilities and Exposures (CVEs) is the sole responsibility of the End User. Brocade shall not be responsible for notifying the End User about or providing the End User with updates to open source components unless explicitly agreed to by Brocade.

BROCADE THEREFORE PROVIDES OPEN SOURCE SOFTWARE ON AN "AS IS" BASIS WITHOUT ANY WARRANTY WHATSOEVER AND HEREBY EXPRESSLY DISCLAIMS WITH RESPECT TO ANY SUCH OPEN SOURCE SOFTWARE AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW: (A) ALL WARRANTIES, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING WITHOUT LIMITATION THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PUROSE AND NON-INFRINGEMENT; AND (B) ALL LIABILITY FOR DIRECT, INDIRECT, INCIDENTAL, SPECIAL, COVER, PUNITIVE, EXEMPLARY OR CONSEQUENTIAL DAMAGES, INCLUDING WITHOUT LIMITATION LOST DATA OR LOST PROFITS, HOWEVER ARISING, WHETHER BASED IN CONTRACT, TORT, OR ANY OTHER LEGAL THEORY, EVEN WHERE ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

NO SUPPORT. Brocade may, but is not required to, provide technical support for the Software.

TERM AND TERMINATION. This Agreement is effective until terminated. End User's license rights under this Agreement will terminate immediately without notice from Brocade if End User fails to comply with any provision of this Agreement. Upon termination, End User must stop using the Software and destroy all copies of Software and the corresponding license keys in its possession or control.

SECURITY NOTICE. Software may include security features that End User can use, along with other security tools, to implement increased security in End User's storage area network ("SAN"). However, use of the security features of the Software does not guarantee the security of End User's SAN or overall network. There are numerous factors that affect the security of a SAN, including, without limitation, correct installation and setup of the security features of Software and all related requirements, correctly configured security policies, selection of hardware and software (including network security tools), correct installation, configuration, and maintenance of the hardware and software, the interoperability of the various components of End User's SAN and End User's network, and a physically and electronically secure operating environment. In addition, Software may utilize digital certificates in connection with its access control features. Although digital certificates are a useful authentication security measure that improves overall security, they do not guarantee authenticity or security. In designing the security of End User's SAN, it is End User's responsibility to evaluate all of these factors to ensure End User's SAN meets End User's security requirements. End User's experience may vary based on these and other factors. End User's use of Software, including the digital certificates, is subject to and governed by the terms of this Agreement and to End User's compliance with the policies and procedures for the use of the security features of Software and digital certificates, which may be made available to End User by Brocade from time to time. If Brocade becomes aware of a breach of the security of its digital certificate infrastructure, Brocade reserves the right to re-issue digital certificates. In that event, End User will be required to submit new certificate signing requests and install reissued certificates across End User's SAN. End User should plan for any network disruption that this may cause. END USER ACKNOWLEDGES THAT END USER HAS ACCESS TO SUFFICIENT INFORMATION TO ENSURE THAT END USER CAN MAKE AN INFORMED DECISION AS TO THE EXTENT TO WHICH END USER CHOOSES TO RELY ON DIGITAL CERTIFICATES AND OTHER SECURITY FEATURES IN SOFTWARE ("SECURITY").

PROPRIETARY RIGHTS. The Software, including all intellectual property rights therein, is and will remain the sole and exclusive property of Brocade or its licensors. Brocade retains all title, ownership, interests and intellectual property rights in and to the Software and deems the Software to be confidential information. To the extent the Software contains any materials licensed from third parties, third party suppliers may own such licensed materials. Brocade retains all rights not expressly granted to Licensee in this EULA. Licensee agrees to protect and maintain the confidentiality of the Software, and not to disclose the Software to any third parties except as expressly permitted in this EULA. Licensee acknowledges that its breach of confidentiality shall cause Brocade irreparable injury and entitle Brocade to obtain equitable relief, in addition to all other remedies available. Licensee further acknowledges that Licensee is acquiring only a limited license to use the Software and not any title to or ownership of the Software or any part thereof and that the Software is proprietary to and copyrighted by Brocade. BROCADE SOFTWARE IS COPYRIGHTED AND LICENSED, NOT SOLD.

LIMITED WARRANTY. BROCADE PROVIDES THE SOFTWARE "AS IS" WITH NO OTHER WARRANTIES, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL BROCADE AND ITS SUPPLIERS, BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF OR INABILITY TO USE THE SOFTWARE PROGRAM OR RELATED MATERIALS, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SPECIFICALLY, BROCADE AND ITS SUPPLIERS, ARE NOT RESPONSIBLE FOR ANY COSTS, INCLUDING, BUT NOT LIMITED TO, THOSE INCURRED AS A RESULT OF LOST PROFITS OR REVENUE, LOSS OF USE OF THE SOFTWARE PROGRAM, LOSS OF DATA, THE COST OF RECOVERING SUCH SOFTWARE PROGRAMS OR DATA, THE COST OF ANY SUBSTITUTE SOFTWARE PROGRAM, CLAIMS BY THIRD PARTIES OR FOR OTHER SIMILAR COSTS. SOME STATES MAY NOT ALLOW THE EXCLUSION, DISCLAIMER OR LIMITATION OF IMPLIED WARRANTIES OR LIMITATION OF LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CERTAIN SITUATIONS, SO THE ABOVE LIMITATION, DISCLAIMER OR EXCLUSION MAY NOT APPLY TO YOU. IN ANY AND ALL EVENTS, IT IS AGREED THAT BROCADE's LIABILITY SHALL NOT EVER EXCEED THE AMOUNT OF THE LICENSE. THE TERMS AND PROVISIONS OF THE UNITED NATIONS CONVENTION ON CONTRACTS FOR INTERNATIONAL SALES OF GOODS ARE HEREBY EXPRESSLY EXCLUDED.

NUCLEAR, AVIATION, OR LIFE SUPPORT APPLICATION. Brocade specifically disclaims liability for use of the Software in connection with the design, construction, maintenance, and/or operation of any (i) nuclear facility, (ii) aircraft, aircraft communication or aircraft ground support system, or (iii) safety or health care control system, including without limitation, life support system. Such use is entirely End User's own risk. Brocade shall not be liable to End User, its vendors, or its customers, in whole or in part, for any claims or damages arising from such use. End User agrees to defend, indemnify and hold Brocade harmless from and against any and all claims (including reasonable attorneys' fees) arising out of any such use by End User or its customers.

TAXES. End User will pay the license fees exclude all sales, use, VAT and other transaction taxes, personal property taxes and other taxes. End User will be responsible for payment of all such taxes (other than those based on Brocade's net income), including any related penalties and interest, arising from the payment of license fees or the delivery or license of the Software to End User, unless End User furnishes satisfactory proof of exemption. End User will make all payments of license fees to Brocade free and clear of, and without reduction for, any withholding taxes; any such taxes imposed on payments of license fees to Brocade will be End User's sole responsibility.

COMPLIANCE WITH LAW. Each party agrees to comply with all applicable laws, rules and regulations in connection with its activities under this Agreement. Without limiting the foregoing, End User acknowledges and agrees that the Software, including technical data, is subject to United States export control laws, including the United States Export Administration Act and its associated regulations, and may be subject to export or import regulations in other countries. End User agrees to comply strictly with all such regulations and acknowledges that End User has the responsibility to obtRESTRICTED RIGHTS. The Software under this Agreement is commercial computer software as that term is described in 48 C.F.R. 252.227-7014(a)(1). If acquired by or on behalf of a civilian agency, the U.S. Government acquires this commercial computer software and/or commercial computer software documentation subject to the terms of this Agreement as specified in 48 C.F.R. 12.212 (Computer Software) and 12.211 (Technical Data) of the Federal Acquisition Regulations ("FAR") and its successors. If acquired by or on behalf of any agency within the Department of Defense ("DOD"), the U.S. Government acquires this commercial computer software and/or commercial computer software documentation subject to the terms of this Agreement as specified in 48 C.F.R. 227.7202-3 of the DOD FAR Supplement ("DFAR") and its successors.

GENERAL. This Agreement will bind and inure to the benefit of each party's successors and assigns, provided that End User may not assign or transfer this Agreement, in whole or in part, without Brocade's written consent. This Agreement shall be governed by and construed in accordance with the laws of the State of California, United States of America, as if performed wholly within the state and without giving effect to the principles of conflict of law. No failure of either party to exercise or enforce any of its rights under this Agreement will act as a waiver of such rights. If any portion hereof is found to be void or unenforceable, the remaining provisions of this Agreement shall remain in full force and effect. This Agreement is the complete and exclusive agreement between the parties with respect to the subject matter hereof, superseding and replacing any and all prior agreements, communications, and understandings (both written and oral) regarding such subject matter.

Should you have any questions concerning this Agreement, please contact Brocade at 1320 Ridder Park Dr, San Jose, CA 95131, USA.   Attention: Legal Department.

Do you agree with these terms and conditions? (Yes / No): [No]
Yes

In order to install SANnav Management Portal services and Containers, the address 192.168.226.133 will be used. Is this the correct and valid IP address? (Y / y / N / n)? [Y/y] Y

SANnav Management Portal Docker Containers use the IP address range "192.168.255.240/28" (that is, 192.168.255.240 to 192.168.255.255) for its default pool. Does this IP address range need to be modified? (Y / y / N / n)? [N/n]N

To proceed with the default home directory (/var/lib/docker) for installing SANnav Management Portal docker files, Press Enter. To change the default directory, please enter a different path for this directory.
Note: Minimum of 120GB of disk space required.
Note: This directory cannot be customized after installation.
/var/lib/docker
Installing SANnav Management Portal docker files. This may take a few minutes.
Successfully installed SANnav Management Portal docker files.

Press Enter to create a swap file in the / directory, or enter a different directory location for the swap file.

Creating swap file. This may take a few minutes.
Created swap file under / directory.

Press Enter to proceed with server installation.

Configure SANnav Management Portal clients automatic redirection from HTTP to HTTPS:
        0 For automatic redirection of SANnav Management Portal clients from HTTP to HTTPS
        1 For no automatic redirection of SANnav Management Portal client from HTTP to HTTPS redirection
0
Automatic redirection of SANnav Management Portal client from HTTP to HTTPS configured.

To configure HTTP or HTTPS connections between SANnav Management Portal and the SAN switches, please select one of the following options:
        0 For HTTP
        1 For HTTPS (SAN switches must be configured for HTTPS connection)
        2 For HTTPS first then HTTP (if HTTPS fails)
2
HTTPS first then HTTP (if HTTPS fails) configured.

To configure the method by which SANnav Management Portal launches WebTools, please select one of the following options:
        0 to always require login when launching WebTools
        1 to launch WebTools with Single Sign On (SSO) using the managed SAN switch credentials
        2 to launch WebTools with SSO using the SANnav Management Portal user's credentials
2
Launch WebTools with SSO using the SANnav MAnagement Portal user's credentials configured

To run the SANnav SSH server on port 22, press Enter to proceed or enter another port number (1- 65535).
Note: If port other than 22 is selected, you cannot use the internal SSH server for downloading firmware to switches and chassis that are running a firmware version lower than 8.2.2

Press Enter to continue with default port number (443) for SANnav to receive HTTPS requests,
or enter a new port number (1-65535).

Press Enter to continue with default port number (162) for SANnav to receive SNMP traps,
or enter a new port number (1-65535).

Press Enter to continue with default port number (514) for SANnav to receive syslog message,
or enter a new port number (1-65535).

```
Press Enter to continue with default port number (6514) for SANnav to receive secure syslog message,
or enter a new port number (1-65535).

The IPv4 address used for SAN Switch to SANnav Management Portal Server communication is 192.168.226.133

Press Enter to continue with default database password, or enter a new password manually.
Password must be between 8 to 64 characters, alphanumeric. Spaces are not allowed. Allowed special characters are !
# $ * ( )

Press Enter to continue with default internal SFTP/SCP password, or enter a new password manually.
Password must be between 8 to 64 characters, alphanumeric. Spaces are not allowed. Allowed special characters are !
# $ * ( )

Enable SANnav Management Portal license automatic renewal? (Note that Internet connectivity for the SANnav
Management Portal server is required for this feature to work.) (Y / y / N / n): [Y/y]
Y

In order to improve the user experience of SANnav product features in the future, SANnav server will collect usage
data. Allow this data to be sent to Broadcom? (Y / y / N / n): [Y/y]N

Loading SANnav services images from File System, this might take a few minutes
Successfully loaded SANnav services images
Created symlink /etc/systemd/system/multi-user.target.wants/sannaviptablesetup.service →
/usr/lib/systemd/system/sannaviptablesetup.service.

SANnav Management Portal server has been successfully installed and started.
SANnav Management Portal server startup may take up to 15 minutes.
To check SANnav Management Portal  server status, run /data/Portal_2.1.1_bld184/bin/check-sannav-status.sh.
When startup has completed, launch the client using [https://192.168.226.133].
[root@sannav bin]#

[root@sannav bin]#
```

Note: The default passwords for the DB and internal SFTP/SCP server is passw0rd.

If you need to check the progress during the install or if you need to find out why it failed, then you
can check the installation log file which is located in the <installation_home>/logs directory. In my
case, the installation log file was called install-sannav-2022_03_18_1538.log.

```
[root@sannav logs]# pwd
/data/Portal_2.1.1_bld184/logs
[root@sannav logs]#
[root@sannav logs]# ls -al
total 8
drwxr-xr-x. 10 root root  190 Mar 18 15:38 .
drwxr-xr-x. 22 root root 4096 Mar 18 14:53 ..
drwxr-xr-x.  2 root root    6 Dec 13  2020 FFDC
drwxr-xr-x.  2 root root    6 Dec 13  2020 ignite-grid-node1
drwxr-xr-x.  2 root root    6 Dec 13  2020 ignite-grid-node2
-rw-r--r--.  1 root root 1813 Mar 18 15:39 install-sannav-2022_03_18_1538.log
drwxr-xr-x.  2 root root    6 Dec 13  2020 kafka-1
drwxr-xr-x.  2 root root    6 Dec 13  2020 nginx
drwxr-xr-x.  2 root root    6 Dec 13  2020 schema-registry
drwxr-xr-x.  2 root root    6 Dec 13  2020 wso2
drwxr-xr-x.  2 root root    6 Dec 13  2020 zookeeper
[root@sannav logs]#

[root@sannav logs]# cat install-sannav-2022_03_18_1538.log | more
2
1
1

Looks docker is not installed on this VM / Host. So no case of migration
Starting system requirements check of the server. Is this migration ? false
Current memory: 3 GB
Number of CPU: 2
Current umask is 0022
UMASK set to: 0022
Number of CPU sockets available: 2
2
1
1

Detected distribution is rhel
Detected os version is 8.2
.

.

.

.

.
Chain OUTPUT (policy ACCEPT)
```

```
target      prot opt source            destination
ACCEPT      udp  -- anywhere           anywhere            udp dpt:bootpc

Chain DOCKER-USER (1 references)
target      prot opt source            destination
RETURN      all  -- anywhere           anywhere
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Chain INPUT (policy ACCEPT)
target      prot opt source            destination

Chain FORWARD (policy ACCEPT)
target      prot opt source            destination

Chain OUTPUT (policy ACCEPT)
target      prot opt source            destination
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
SANnav Management Portal server has been successfully installed and started.
SANnav Management Portal server startup may take up to 15 minutes.
To check SANnav Management Portal  server status, run /data/Portal 2.1.1 bld184/bin/check-sannav-status.sh.
When startup has completed, launch the client using [https://192.168.226.133].
[root@sannav logs]#
```

Let's check the status of SANnav after the installation.

```
[root@sannav bin]# ./check-sannav-status.sh
Following services are currently down
topology-middleware
backuprestore-mw
authentication-rbac-middleware
collections-middleware
performancemanagement-middleware
filetransfer
supportsave-mw
elasticsearch
switch-asset-collectors-mw
schema-registry
license-mw
kafka-1
dashboard-summaryprovider
cfgmgmt-policy-middleware
switch-filetransfer-mw
filters-contextsearch-middleware
fe-consolidated-1
fe-consolidated-2
performancemanagement-statscollector
performancemanagement-datastore
faultmanagement-collector
ignite-grid-object-manager-node
asyncjobscheduler-manager
ignite-grid-node1
flow-management-mw
g-agent
dcm-wso2
faultmanagement-backend
externalapi-middleware
dashboard-middleware
troubleshooting
asyncjobscheduler-worker
```
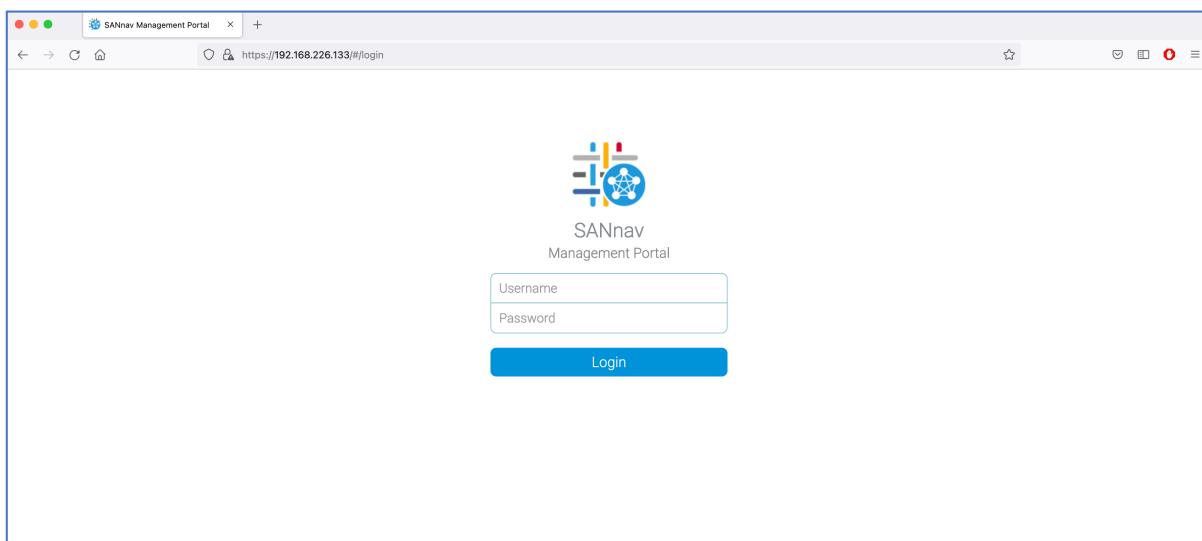
Wait for all the services to start. You will know when the script returns nothing.

```
[root@sannav bin]# ./check-sannav-status.sh
[root@sannav bin]#
```

Hooray! Now we should be able to login. You can now login and start adding switches and directors as required.

Note: The default username is "Administrator" and the default password is "password"

If you are really interested, you can query the SANnav containers using docker.

```
[root@sannav logs]# docker ps
CONTAINER ID      IMAGE                                               COMMAND
CREATED           STATUS          PORTS           NAMES
c4ff3c085d4d      192.168.226.133:5000/filetransfer:sann2.1.1         "/bin/sh /dcm1.0.0/w…"    8
minutes ago       Up 8 minutes                    dcm_2_1_1_filetransfer.1.y6mu7nwegcw46x68vg3curifu
2817b49ef136      192.168.226.133:5000/faultmanagement-collector:sann2.1.1    "/bin/sh /dcm1.0.0/w…"    8
minutes ago       Up 8 minutes                    dcm_2_1_1_faultmanagement-
collector.1.9sbke1rdsps01e3vqg6sp5zty
f34fa34b8598      192.168.226.133:5000/proxy-local:sann2.1.1          "nginx -g 'daemon of…"    8
minutes ago       Up 8 minutes                    dcm_2_1_1_proxy.1.9clco6ee1gbpey8hxckjz3unc
f629d2126a7d      192.168.226.133:5000/mw-cfgmgmt-policy:sann2.1.1    "/bin/sh /dcm1.0.0/w…"    8
minutes ago       Up 8 minutes                    dcm_2_1_1_cfgmgmt-policy-
middleware.1.rwv5zcchgawtrlxcpvi871js8
153c0950b7e9      192.168.226.133:5000/be-summaryprovider:sann2.1.1   "/bin/sh /dcm1.0.0/w…"    8
minutes ago       Up 8 minutes                    dcm 2 1 1 dashboard-
summaryprovider.1.2mq1195ieko42ruh4dzw8np3f
f86465684193      192.168.226.133:5000/fe-troubleshooting:sann2.1.1   "/bin/sh /dcm1.0.0/w…"    8
minutes ago       Up 8 minutes                    dcm_2_1_1_troubleshooting.1.055rn1bqnl7xqgbjbljg5vjrg
500129f2e737      192.168.226.133:5000/node-exporter:sann2.1.1        "/bin/node_exporter"      8
minutes ago       Up 8 minutes                    dcm_2_1_1_nodeexporter.1.tbilf6863cya3rq4ub6m2c72a
1d6f3564cc27      192.168.226.133:5000/ignite-grid-node:sann2.1.1     "/opt/ignite/gridgai…"    8
minutes ago       Up 8 minutes                    dcm_2_1_1_ignite-grid-node1.1.c3jo0a4yj6ws18xh7fboxdr1x
c7f2e34b6e36      192.168.226.133:5000/cs-topology:sann2.1.1          "/bin/sh /dcm1.0.0/w…"    8
minutes ago       Up 8 minutes                    dcm_2_1_1_topology-middleware.1.xh4oy0k3b36agqrpdtnfug4qa
bede20507f0b      192.168.226.133:5000/reportgenerator:sann2.1.1      "/bin/sh -c '\"node\" …"  8
minutes ago       Up 8 minutes                    dcm_2_1_1_reportgenerator.1.vshzirtftsgbd6rk3k5ssucl6
91a9c62cfd16      192.168.226.133:5000/mw-dashboard:sann2.1.1         "/bin/sh /dcm1.0.0/w…"    8
minutes ago       Up 8 minutes                    dcm_2_1_1_dashboard-
middleware.1.v2n1mkagpkn5c6m6frjzhcb8g
74927ea37fa2      192.168.226.133:5000/mw-authentication-rbac:sann2.1.1    "/bin/sh /dcm1.0.0/w…"    8
minutes ago       Up 8 minutes                    dcm_2_1_1_authentication-rbac-
middleware.1.tgdkz0am6uii4lqkf1n3flky3
53202f40eda9      192.168.226.133:5000/datastore-performancemanagement:sann2.1.1    "/bin/sh /dcm1.0.0/w…"    8
minutes ago       Up 8 minutes                    dcm_2_1_1_performancemanagement-
datastore.1.f7blaqwslyuniz7lbxl3en6df
d67f3b56c839      192.168.226.133:5000/asyncjobscheduler-worker:sann2.1.1    "/bin/sh /dcm1.0.0/w…"    8
minutes ago       Up 8 minutes                    dcm 2 1 1 asyncjobscheduler-
worker.1.ix9da8ufq562q7cjuoh54xiya
29892ca7cb38      192.168.226.133:5000/ignite-grid-object-manager-node:sann2.1.1    "/bin/sh /dcm1.0.0/w…"    8
minutes ago       Up 8 minutes                    dcm 2 1 1 ignite-grid-object-manager-
node.1.oaqq72chqicefsfudr5onldti
3dae033d0a9a      192.168.226.133:5000/be-faultmanagement:sann2.1.1   "/bin/sh /dcm1.0.0/w…"    8
minutes ago       Up 8 minutes                    dcm 2 1 1 faultmanagement-
backend.1.ja9hyu3khqim1xsh1c34uj6vx
280c2cbc7b85      192.168.226.133:5000/fe-consolidated-2:sann2.1.1    "/bin/sh /dcm1.0.0/w…"    8
minutes ago       Up 8 minutes                    dcm_2_1_1_fe-consolidated-2.1.kvd84x5mhvldh0checqupn6o1
fa71e4ca108d      192.168.226.133:5000/mw-filters-contextsearch:sann2.1.1    "/bin/sh /dcm1.0.0/w…"    8
minutes ago       Up 8 minutes                    dcm_2_1_1_filters-contextsearch-
middleware.1.kxgmlck324v945wzo5nt8elqv
a3d79a8d2255      192.168.226.133:5000/system-monitor:sann2.1.1       "/bin/sh /dcm1.0.0/w…"    8
minutes ago       Up 8 minutes                    dcm_2_1_1_system-monitor.1.romhbo72b1cty87kbc6z562ua
5e552508ef3e      192.168.226.133:5000/g-agent:sann2.1.1              "/bin/sh /dcm1.0.0/w…"    8
minutes ago       Up 8 minutes                    dcm_2_1_1_g-agent.1.ld2e7ysyytzebt2p5xkgpvtip
```

```
c4f594c037e5        192.168.226.133:5000/be-performancemanagement:sann2.1.1        "/bin/sh /dcm1.0.0/w…"    8
minutes ago      Up 8 minutes                       dcm_2_1_1_performancemanagement-
statscollector.1.66lkuijw2xwh597psuofb8vcs
46fd7a047609        192.168.226.133:5000/licensing-middleware:sann2.1.1        "/bin/sh /dcm1.0.0/w…"    8
minutes ago      Up 8 minutes                       dcm_2_1_1_license-mw.1.lvz5bof0ng1b0oxuc22jmyc2j
ecc2a0e94168        192.168.226.133:5000/dcm-postgres-db:sann2.1.1        "docker-entrypoint.s…"    8
minutes ago      Up 8 minutes                       dcm_2_1_1_dcm-postgres-db.1.qbwkqmfitw4xe387b5pdwycbi
162cc89b8e7f        192.168.226.133:5000/prometheus:sann2.1.1        "/bin/prometheus --c…"    8
minutes ago      Up 8 minutes                       dcm_2_1_1_prometheus.1.j1jpostau4ln13dtqvcr790kq
67f88a5c84ee        192.168.226.133:5000/fe-consolidated-1:sann2.1.1        "/bin/sh /dcm1.0.0/w…"    8
minutes ago      Up 8 minutes                       dcm_2_1_1_fe-consolidated-1.1.6vyyebx4yl90j9zliif72kf5x
610cafac13ca        192.168.226.133:5000/mw-externalapi:sann2.1.1        "/bin/sh /dcm1.0.0/w…"    8
minutes ago      Up 8 minutes                       dcm_2_1_1_externalapi-
middleware.1.k2agwq0uwv4levx1sue0t4ljq
38d92ff472de        192.168.226.133:5000/flow-management-mw:sann2.1.1        "/bin/sh /dcm1.0.0/w…"    8
minutes ago      Up 8 minutes                       dcm_2_1_1_flow-management-mw.1.7dxhq7h16k0sg6jk09q22446n
472e1c0b3fb3        192.168.226.133:5000/cp-zookeeper:sann2.1.1        "/etc/confluent/dock…"    8
minutes ago      Up 8 minutes                       dcm_2_1_1_zookeeper.1.o8qccxfd3spstsr8ahhgm44lj
cf6d9786ccd8        192.168.226.133:5000/cadvisor:sann2.1.1        "/usr/bin/cadvisor -…"    8
minutes ago      Up 8 minutes                       dcm_2_1_1_cadvisor.1.5mx9owdusnjxn4k296wwahzxr
81093c298241        192.168.226.133:5000/mw-switch-filetransfer:sann2.1.1        "/bin/sh /dcm1.0.0/w…"    8
minutes ago      Up 8 minutes                       dcm_2_1_1_switch-filetransfer-
mw.1.1nglwmthmhd90vqkyed2w2ym0
4f5280b6e4c6        192.168.226.133:5000/asyncjobscheduler-manager:sann2.1.1        "/bin/sh /dcm1.0.0/w…"    8
minutes ago      Up 8 minutes                       dcm_2_1_1_asyncjobscheduler-
manager.1.khucd11r2ut3u7re7rohpr4zj
47019f446ade        192.168.226.133:5000/cp-schema-registry:sann2.1.1        "/etc/confluent/dock…"    9
minutes ago      Up 8 minutes                       dcm_2_1_1_schema-registry.1.gj2tiaxiznjnw9554fqq1sy38
834cba4a809b        192.168.226.133:5000/mw-collections:sann2.1.1        "/bin/sh /dcm1.0.0/w…"    9
minutes ago      Up 8 minutes                       dcm_2_1_1_collections-
middleware.1.w3jdv0ax0i7tyglobmde791i4
185a91d372bc        192.168.226.133:5000/mw-supportsave:sann2.1.1        "/bin/sh /dcm1.0.0/w…"    9
minutes ago      Up 8 minutes                       dcm_2_1_1_supportsave-mw.1.19jnowrmaz5t2zl5gfxzy71ng
126452ffd52d        192.168.226.133:5000/dcm-wso2:sann2.1.1        "/opt/wso2cep-4.2.0/…"    9
minutes ago      Up 8 minutes                       dcm_2_1_1_dcm-wso2.1.tymacz07bs9zfsmlus8tgb1ip
b5acf5c40e99        192.168.226.133:5000/switch-asset-collectors-mw:sann2.1.1        "/bin/sh /dcm1.0.0/w…"    9
minutes ago      Up 8 minutes                       dcm_2_1_1_switch-asset-collectors-
mw.1.sjfyq8h08jlm9rwd2grmmn0ni
1e84d4bf49b3        192.168.226.133:5000/mw-backuprestore:sann2.1.1        "/bin/sh /dcm1.0.0/w…"    9
minutes ago      Up 8 minutes                       dcm_2_1_1_backuprestore-mw.1.vg8lz0s4p1kp683vmzdn6ecls
8e90e86fa505        192.168.226.133:5000/mw-performancemanagement:sann2.1.1        "/bin/sh /dcm1.0.0/w…"    9
minutes ago      Up 8 minutes                       dcm_2_1_1_performancemanagement-
middleware.1.afvif00ty1qwtnpqcmpr2y0qg
705d0f9b467a        192.168.226.133:5000/cp-kafka:sann2.1.1        "/etc/confluent/dock…"    9
minutes ago      Up 8 minutes                       dcm_2_1_1_kafka-1.1.mw52hw36lmq1i53mvlfjwohv5
[root@sannav logs]#
```

If for any reason you have a installation failure and want to re-try the installation, you can issue the uninstall command to cleanup the previous installation. The uninstall script is located in the <installation_home>/bin directory.

```
[root@sannav bin]# ./uninstall-sannav.sh
Stopping and uninstalling SANnav Management Portal services. This may take a few minutes.
 [|]  Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the docker daemon running?
Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the docker daemon running?
iptables: No chain/target/match by that name.
iptables v1.8.4 (nf_tables): Chain 'SANNAV-CHAIN' does not exist
Try `iptables -h' or 'iptables --help' for more information.
iptables: No chain/target/match by that name.
ip6tables: No chain/target/match by that name.
ip6tables v1.8.4 (nf_tables): Chain 'SANNAV-CHAIN' does not exist
Try `ip6tables -h' or 'ip6tables --help' for more information.
ip6tables: No chain/target/match by that name.
error reading information on service sannavRCLocal: No such file or directory
SANnav Management Portal server uninstallation successful.
SANnav Management Portal server uninstallation is complete.
Server reboot is recommended. Reboot server now ? (Y / y / N / n): [N/n]
N
[root@sannav bin]#
```

You can now fix whatever issue it complained about and re-try the installation.

Upgrading SANnav Management Portal to V2.1.1.7

Right we have our base installation and now need to apply the latest path. As of the publication of this document the latest version is 2.1.1.7. Let's navigate to our <installation_home> directory and unpack the patch that we previously transferred with the installation image and run it. We need to make sure

we are issuing commands as the root user ofcourse and have the correct umask set prior to unpacking and upgrading.

```
[root@sannav ~]# cd /data/
[root@sannav data]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@sannav data]# ls -l
total 6676848
-rw-r--r--.  1 nishaan nishaan 1004954995 Mar 18 14:44 Portal_2.1.1.7.tar.gz
drwxr-xr-x. 22 root    root          4096 Mar 18 14:53 Portal_2.1.1_bld184
-rw-r--r--.  1 nishaan nishaan 5832130513 Mar 18 14:21 Portal_2.1.1-distribution.tar.gz
[root@sannav data]# cd Portal_2.1.1_bld184/bin
[root@sannav bin]# pwd
/data/Portal_2.1.1_bld184/bin
[root@sannav bin]#
[root@sannav Portal_2.1.1_bld184]# umask
0022
[root@sannav bin]# tar -xvzf /data/Portal_2.1.1.7.tar.gz
patch-deploy-portal_2.1.1.7.sh
delete_auto_enclosures.sh
patch-log4j.sh
fix-log4j-vulnerability.sh
Portal_2.1.1.7_bld007/
Portal_2.1.1.7_bld007/be-faultmanagement:sann2.1.1.7.tar
Portal_2.1.1.7_bld007/fe-consolidated-1:sann2.1.1.7.tar
Portal_2.1.1.7_bld007/fe-troubleshooting:sann2.1.1.7.tar
Portal_2.1.1.7_bld007/mw-switch-filetransfer:sann2.1.1.7.tar
Portal_2.1.1.7_bld007/proxy-local:sann2.1.1.7.tar
Portal_2.1.1.7_bld007/switch-asset-collectors-mw:sann2.1.1.7.tar
Portal_2.1.1.7_bld007/externalapi-middleware-1.0.0.jar
Portal_2.1.1.7_bld007/log4j-1.2.14.jar
Portal_2.1.1.7_bld007/log4j-1.2.17.jar
Portal_2.1.1.7_bld007/logback-classic-1.1.11.jar
[root@sannav bin]#
[root@sannav bin]# ls -al ./patch-deploy-portal_2.1.1.7.sh
-rwxrwxr-x. 1 root root 5281 Feb 28 22:20 ./patch-deploy-portal 2.1.1.7.sh
[root@sannav bin]#
```

Now that everything is unpacked, we can run the patch installation script.

```
[root@sannav bin]# ./patch-deploy-portal_2.1.1.7.sh
################################################
############# GENERIC PATCH DEPLOY ##############
################################################
DCM_HOME:/data/Portal_2.1.1_bld184
Build Number is bld007
FOLDER_NAME is Portal_2.1.1.7_bld007
SANnav host IP: 192.168.226.133
Stopping the server:/data/Portal_2.1.1_bld184

Stopping SANnav services

dcm_2_1_1_asyncjobscheduler-manager scaled to 0
dcm_2_1_1_asyncjobscheduler-worker scaled to 0
dcm_2_1_1_authentication-rbac-middleware scaled to 0
dcm_2_1_1_backuprestore-mw scaled to 0
dcm_2_1_1_cadvisor scaled to 0
dcm_2_1_1_cfgmgmt-policy-middleware scaled to 0
dcm_2_1_1_collections-middleware scaled to 0
dcm_2_1_1_dashboard-middleware scaled to 0
dcm_2_1_1_dashboard-summaryprovider scaled to 0
dcm_2_1_1_dcm-postgres-db scaled to 0
dcm_2_1_1_dcm-wso2 scaled to 0
dcm_2_1_1_elasticsearch scaled to 0
dcm_2_1_1_externalapi-middleware scaled to 0
dcm_2_1_1_faultmanagement-backend scaled to 0
dcm_2_1_1_faultmanagement-collector scaled to 0
dcm_2_1_1_fe-consolidated-1 scaled to 0
dcm_2_1_1_fe-consolidated-2 scaled to 0
dcm_2_1_1_filetransfer scaled to 0
dcm_2_1_1_filters-contextsearch-middleware scaled to 0
dcm_2_1_1_flow-management-mw scaled to 0
dcm_2_1_1_g-agent scaled to 0
dcm_2_1_1_ignite-grid-node1 scaled to 0
dcm_2_1_1_ignite-grid-object-manager-node scaled to 0
dcm_2_1_1_kafka-1 scaled to 0
dcm_2_1_1_license-mw scaled to 0
dcm_2_1_1_nodeexporter scaled to 0
dcm_2_1_1_performancemanagement-datastore scaled to 0
dcm_2_1_1_performancemanagement-middleware scaled to 0
dcm_2_1_1_performancemanagement-statscollector scaled to 0
dcm_2_1_1_prometheus scaled to 0
dcm_2_1_1_proxy scaled to 0
dcm_2_1_1_reportgenerator scaled to 0
dcm_2_1_1_schema-registry scaled to 0
dcm_2_1_1_supportsave-mw scaled to 0
```

```
dcm_2_1_1_switch-asset-collectors-mw scaled to 0
dcm_2_1_1_switch-filetransfer-mw scaled to 0
dcm_2_1_1_system-monitor scaled to 0
dcm_2_1_1_topology-middleware scaled to 0
dcm_2_1_1_troubleshooting scaled to 0
dcm_2_1_1_zookeeper scaled to 0
Waiting for 30 Seconds to shutdown the SANnav
be-faultmanagement:sann2.1.1.7
be-faultmanagement
Docker service ID: vfdm5zgtnxgr
Loading the DCM services images from File System
241acdbeebbb: Loading layer [==================================================>]  6.036MB/6.036MB
421f35f3b61d: Loading layer [==================================================>]  1.319MB/1.319MB
fc10c80e5957: Loading layer [==================================================>]  1.308MB/1.308MB
94a0cae3c076: Loading layer [==================================================>]  2.336MB/2.336MB
4a88b987ba01: Loading layer [==================================================>]  1.536kB/1.536kB
9dfcba32893e: Loading layer [==================================================>]  2.048kB/2.048kB
eb5798fe876a: Loading layer [==================================================>]  2.048kB/2.048kB
68151e11a81d: Loading layer [==================================================>]  2.048kB/2.048kB
24f8f4046308: Loading layer [==================================================>]  2.048kB/2.048kB
d602b1c365da: Loading layer [==================================================>]   2.56kB/2.56kB
1eab60ed1285: Loading layer [==================================================>]  2.048kB/2.048kB
f97ed24b7eed: Loading layer [==================================================>]   2.56kB/2.56kB
4ad2f5474ac1: Loading layer [==================================================>]  130.6MB/130.6MB
bb743aec3f0b: Loading layer [==================================================>]  3.584kB/3.584kB
Loaded image: localhost:5000/be-faultmanagement:sann2.1.1.7
Successfully loaded the DCM services images from File System
image 192.168.226.133:5000/be-faultmanagement:sann2.1.1.7 could not be accessed on a registry to record
its digest. Each node will access 192.168.226.133:5000/be-faultmanagement:sann2.1.1.7 independently,
possibly leading to different nodes running different
versions of the image.

vfdm5zgtnxgr
overall progress: 0 out of 0 tasks
verify: Service converged
fe-consolidated-1:sann2.1.1.7
fe-consolidated-1
Docker service ID: l3fuczo9zqli
Loading the DCM services images from File System
d78f440d0eb0: Loading layer [==================================================>]   2.56kB/2.56kB
a1be06d6b150: Loading layer [==================================================>]  110.5MB/110.5MB
0a3a3d67a21f: Loading layer [==================================================>]  3.584kB/3.584kB
Loaded image: localhost:5000/fe-consolidated-1:sann2.1.1.7
Successfully loaded the DCM services images from File System
image 192.168.226.133:5000/fe-consolidated-1:sann2.1.1.7 could not be accessed on a registry to record
its digest. Each node will access 192.168.226.133:5000/fe-consolidated-1:sann2.1.1.7 independently,
possibly leading to different nodes running different
versions of the image.

l3fuczo9zqli
overall progress: 0 out of 0 tasks
verify: Service converged
fe-troubleshooting:sann2.1.1.7
fe-troubleshooting
Docker service ID: plx5ow9918fs
Loading the DCM services images from File System
364638092ea8: Loading layer [==================================================>]  90.24MB/90.24MB
fef7f0852769: Loading layer [==================================================>]  3.584kB/3.584kB
Loaded image: localhost:5000/fe-troubleshooting:sann2.1.1.7
Successfully loaded the DCM services images from File System
image 192.168.226.133:5000/fe-troubleshooting:sann2.1.1.7 could not be accessed on a registry to record
its digest. Each node will access 192.168.226.133:5000/fe-troubleshooting:sann2.1.1.7 independently,
possibly leading to different nodes running different
versions of the image.

plx5ow9918fs
overall progress: 0 out of 0 tasks
verify: Service converged
mw-switch-filetransfer:sann2.1.1.7
mw-switch-filetransfer
Docker service ID: 9oq5w8uj857h
Loading the DCM services images from File System
8bd4bb2cd1e1: Loading layer [==================================================>]   2.56kB/2.56kB
395bac1287a4: Loading layer [==================================================>]  93.48MB/93.48MB
f83438fd3012: Loading layer [==================================================>]  3.584kB/3.584kB
Loaded image: localhost:5000/mw-switch-filetransfer:sann2.1.1.7
Successfully loaded the DCM services images from File System
image 192.168.226.133:5000/mw-switch-filetransfer:sann2.1.1.7 could not be accessed on a registry to record
its digest. Each node will access 192.168.226.133:5000/mw-switch-filetransfer:sann2.1.1.7 independently,
possibly leading to different nodes running different
versions of the image.

9oq5w8uj857h
overall progress: 0 out of 0 tasks
verify: Service converged
proxy-local:sann2.1.1.7
proxy-local
Docker service ID: j0kkxhpuf3o3
Loading the DCM services images from File System
9b9559fcfc33: Loading layer [==================================================>]   2.56kB/2.56kB
73127cdf6c94: Loading layer [==================================================>]  25.05MB/25.05MB
```

```
Loaded image: localhost:5000/proxy-local:sann2.1.1.7
Successfully loaded the DCM services images from File System
image 192.168.226.133:5000/proxy-local:sann2.1.1.7 could not be accessed on a registry to record
its digest. Each node will access 192.168.226.133:5000/proxy-local:sann2.1.1.7 independently,
possibly leading to different nodes running different
versions of the image.

j0kkxhpuf3o3
overall progress: 0 out of 0 tasks
verify: Service converged
switch-asset-collectors-mw:sann2.1.1.7
switch-asset-collectors-mw
Docker service ID: qqfhmug8x51v
Loading the DCM services images from File System
e07b68e2d96f: Loading layer [==================================================>]   2.56kB/2.56kB
6b4dcc971bcf: Loading layer [==================================================>]  3.072kB/3.072kB
a44cd2e0b583: Loading layer [==================================================>]   2.56kB/2.56kB
e5a4fe9a504f: Loading layer [==================================================>]  111.3MB/111.3MB
7cc5d2e760a9: Loading layer [==================================================>]  3.584kB/3.584kB
Loaded image: localhost:5000/switch-asset-collectors-mw:sann2.1.1.7
Successfully loaded the DCM services images from File System
image 192.168.226.133:5000/switch-asset-collectors-mw:sann2.1.1.7 could not be accessed on a registry to record
its digest. Each node will access 192.168.226.133:5000/switch-asset-collectors-mw:sann2.1.1.7 independently,
possibly leading to different nodes running different
versions of the image.

qqfhmug8x51v
overall progress: 0 out of 0 tasks
verify: Service converged
Executing fix_log4j_vulnerability.sh script

##############################################
########## Fix Log4j Vulnerability ###########
##############################################

Stop the SANnav services.

Stopping SANnav services

dcm_2_1_1_asyncjobscheduler-manager scaled to 0
dcm_2_1_1_asyncjobscheduler-worker scaled to 0
dcm_2_1_1_authentication-rbac-middleware scaled to 0
dcm_2_1_1_backuprestore-mw scaled to 0
dcm_2_1_1_cadvisor scaled to 0
dcm_2_1_1_cfgmgmt-policy-middleware scaled to 0
dcm_2_1_1_collections-middleware scaled to 0
dcm_2_1_1_dashboard-middleware scaled to 0
dcm_2_1_1_dashboard-summaryprovider scaled to 0
dcm_2_1_1_dcm-postgres-db scaled to 0
dcm_2_1_1_dcm-wso2 scaled to 0
dcm_2_1_1_elasticsearch scaled to 0
dcm_2_1_1_externalapi-middleware scaled to 0
dcm_2_1_1_faultmanagement-backend scaled to 0
dcm_2_1_1_faultmanagement-collector scaled to 0
dcm_2_1_1_fe-consolidated-1 scaled to 0
dcm_2_1_1_fe-consolidated-2 scaled to 0
dcm_2_1_1_filetransfer scaled to 0
dcm_2_1_1_filters-contextsearch-middleware scaled to 0
dcm_2_1_1_flow-management-mw scaled to 0
dcm_2_1_1_g-agent scaled to 0
dcm_2_1_1_ignite-grid-node1 scaled to 0
dcm_2_1_1_ignite-grid-object-manager-node scaled to 0
dcm_2_1_1_kafka-1 scaled to 0
dcm_2_1_1_license-mw scaled to 0
dcm_2_1_1_nodeexporter scaled to 0
dcm_2_1_1_performancemanagement-datastore scaled to 0
dcm_2_1_1_performancemanagement-middleware scaled to 0
dcm_2_1_1_performancemanagement-statscollector scaled to 0
dcm_2_1_1_prometheus scaled to 0
dcm_2_1_1_proxy scaled to 0
dcm_2_1_1_reportgenerator scaled to 0
dcm_2_1_1_schema-registry scaled to 0
dcm_2_1_1_supportsave-mw scaled to 0
dcm_2_1_1_switch-asset-collectors-mw scaled to 0
dcm_2_1_1_switch-filetransfer-mw scaled to 0
dcm_2_1_1_system-monitor scaled to 0
dcm_2_1_1_topology-middleware scaled to 0
dcm_2_1_1_troubleshooting scaled to 0
dcm_2_1_1_zookeeper scaled to 0
Waiting for 30 Seconds to shutdown the SANnav
Stop the Docker Service.

Successfully stopped the docker service.

Run the script patch-log4j.sh.

/var/lib/docker/docker/overlay2/ba0af5a21a2edc833809240093585f10c32b5e0507334ad541119b724ac61b14/diff/usr/share/jav
a/kafka/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/f7aee4936593c477d77410e43d07f991b50c06b3f51d7af53e191a31a5b08ef8/diff/opt/ignite/gr
idgain-professional-fabric-2.5.11/benchmarks/libs/log4j-1.2.17.jar
```

```
/var/lib/docker/docker/overlay2/f7aee4936593c477d77410e43d07f991b50c06b3f51d7af53e191a31a5b08ef8/diff/opt/ignite/gr
idgain-professional-fabric-2.5.11/libs/optional/ignite-log4j/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/f7aee4936593c477d77410e43d07f991b50c06b3f51d7af53e191a31a5b08ef8/diff/opt/ignite/gr
idgain-professional-fabric-2.5.11/libs/optional/ignite-osgi-paxlogging/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/f7aee4936593c477d77410e43d07f991b50c06b3f51d7af53e191a31a5b08ef8/diff/opt/ignite/gr
idgain-professional-fabric-2.5.11/libs/optional/ignite-rest-http/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/f7aee4936593c477d77410e43d07f991b50c06b3f51d7af53e191a31a5b08ef8/diff/opt/ignite/gr
idgain-professional-fabric-2.5.11/libs/optional/ignite-zookeeper/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/80d69aaefc80b5b24055f102af7d96f8d45b133968096f3009dd5a1e569e38e9/diff/opt/ignite/gr
idgain-professional-fabric-2.5.11/libs/ignite-rest-http/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/dea19ee9f30baf2aa736a9d26b3f4c24f1ac2b6ebdd4a48d4bb001ef13e86af5/diff/usr/share/jav
a/kafka/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/da20ec45451c54388661ef3017aada5f213f2c756af45ab27513ad0caa2a2305/diff/usr/share/jav
a/confluent-control-center/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/da20ec45451c54388661ef3017aada5f213f2c756af45ab27513ad0caa2a2305/diff/usr/share/jav
a/confluent-rebalancer/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/da20ec45451c54388661ef3017aada5f213f2c756af45ab27513ad0caa2a2305/diff/usr/share/jav
a/schema-registry/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/da20ec45451c54388661ef3017aada5f213f2c756af45ab27513ad0caa2a2305/diff/usr/share/jav
a/acl/acl-5.2.2.jar
/var/lib/docker/docker/overlay2/f7aee4936593c477d77410e43d07f991b50c06b3f51d7af53e191a31a5b08ef8/diff/opt/ignite/gr
idgain-professional-fabric-2.5.11/libs/optional/ignite-yarn/ignite-yarn-2.5.11.jar
/var/lib/docker/docker/overlay2/cbb5555681eb062d46da7779cbfca32ef864fbec9b972c836df6c49fddc991dc/diff/opt/wso2cep-
4.2.0/lib/org.wso2.carbon.logging-4.4.9.jar
/var/lib/docker/docker/overlay2/cbb5555681eb062d46da7779cbfca32ef864fbec9b972c836df6c49fddc991dc/diff/opt/wso2cep-
4.2.0/repository/components/plugins/org.wso2.carbon.logging_4.4.7.jar
/var/lib/docker/docker/overlay2/cbb5555681eb062d46da7779cbfca32ef864fbec9b972c836df6c49fddc991dc/diff/opt/wso2cep-
4.2.0/repository/components/plugins/org.wso2.carbon.logging_4.4.9.jar
/var/lib/docker/docker/overlay2/f7aee4936593c477d77410e43d07f991b50c06b3f51d7af53e191a31a5b08ef8/diff/opt/ignite/gr
idgain-professional-fabric-2.5.11/libs/optional/ignite-log4j2/log4j-core-2.11.0.jar
/var/lib/docker/docker/overlay2/e09f6f2cff829d97a8089d4a0032b415d6f222e4a61987164ea4fbf8eeaeb7d7/diff/opt/ignite/gr
idgain-professional-fabric-2.5.11/libs/ignite-log4j2/log4j-core-2.11.0.jar
/var/lib/docker/docker/overlay2/80832bc05353cab816c95267a17342b6d49ca0a613462b7fc04f4c013634c624/diff/usr/share/ela
sticsearch/lib/log4j-core-2.11.1.jar
/var/lib/docker/docker/overlay2/ba0af5a21a2edc833809240093585f10c32b5e0507334ad541119b724ac61b14/diff/usr/share/jav
a/kafka/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/f7aee4936593c477d77410e43d07f991b50c06b3f51d7af53e191a31a5b08ef8/diff/opt/ignite/gr
idgain-professional-fabric-2.5.11/benchmarks/libs/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/f7aee4936593c477d77410e43d07f991b50c06b3f51d7af53e191a31a5b08ef8/diff/opt/ignite/gr
idgain-professional-fabric-2.5.11/libs/optional/ignite-log4j/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/f7aee4936593c477d77410e43d07f991b50c06b3f51d7af53e191a31a5b08ef8/diff/opt/ignite/gr
idgain-professional-fabric-2.5.11/libs/optional/ignite-osgi-paxlogging/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/f7aee4936593c477d77410e43d07f991b50c06b3f51d7af53e191a31a5b08ef8/diff/opt/ignite/gr
idgain-professional-fabric-2.5.11/libs/optional/ignite-rest-http/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/f7aee4936593c477d77410e43d07f991b50c06b3f51d7af53e191a31a5b08ef8/diff/opt/ignite/gr
idgain-professional-fabric-2.5.11/libs/optional/ignite-zookeeper/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/80d69aaefc80b5b24055f102af7d96f8d45b133968096f3009dd5a1e569e38e9/diff/opt/ignite/gr
idgain-professional-fabric-2.5.11/libs/ignite-rest-http/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/dea19ee9f30baf2aa736a9d26b3f4c24f1ac2b6ebdd4a48d4bb001ef13e86af5/diff/usr/share/jav
a/kafka/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/da20ec45451c54388661ef3017aada5f213f2c756af45ab27513ad0caa2a2305/diff/usr/share/jav
a/confluent-control-center/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/da20ec45451c54388661ef3017aada5f213f2c756af45ab27513ad0caa2a2305/diff/usr/share/jav
a/confluent-rebalancer/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/da20ec45451c54388661ef3017aada5f213f2c756af45ab27513ad0caa2a2305/diff/usr/share/jav
a/schema-registry/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/da20ec45451c54388661ef3017aada5f213f2c756af45ab27513ad0caa2a2305/diff/usr/share/jav
a/acl/acl-5.2.2.jar
/var/lib/docker/docker/overlay2/f7aee4936593c477d77410e43d07f991b50c06b3f51d7af53e191a31a5b08ef8/diff/opt/ignite/gr
idgain-professional-fabric-2.5.11/libs/optional/ignite-yarn/ignite-yarn-2.5.11.jar
/var/lib/docker/docker/overlay2/cbb5555681eb062d46da7779cbfca32ef864fbec9b972c836df6c49fddc991dc/diff/opt/wso2cep-
4.2.0/lib/org.wso2.carbon.logging-4.4.9.jar
/var/lib/docker/docker/overlay2/cbb5555681eb062d46da7779cbfca32ef864fbec9b972c836df6c49fddc991dc/diff/opt/wso2cep-
4.2.0/repository/components/plugins/org.wso2.carbon.logging_4.4.7.jar
/var/lib/docker/docker/overlay2/cbb5555681eb062d46da7779cbfca32ef864fbec9b972c836df6c49fddc991dc/diff/opt/wso2cep-
4.2.0/repository/components/plugins/org.wso2.carbon.logging_4.4.9.jar
/var/lib/docker/docker/overlay2/ba0af5a21a2edc833809240093585f10c32b5e0507334ad541119b724ac61b14/diff/usr/share/jav
a/kafka/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/f7aee4936593c477d77410e43d07f991b50c06b3f51d7af53e191a31a5b08ef8/diff/opt/ignite/gr
idgain-professional-fabric-2.5.11/benchmarks/libs/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/f7aee4936593c477d77410e43d07f991b50c06b3f51d7af53e191a31a5b08ef8/diff/opt/ignite/gr
idgain-professional-fabric-2.5.11/libs/optional/ignite-log4j/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/f7aee4936593c477d77410e43d07f991b50c06b3f51d7af53e191a31a5b08ef8/diff/opt/ignite/gr
idgain-professional-fabric-2.5.11/libs/optional/ignite-osgi-paxlogging/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/f7aee4936593c477d77410e43d07f991b50c06b3f51d7af53e191a31a5b08ef8/diff/opt/ignite/gr
idgain-professional-fabric-2.5.11/libs/optional/ignite-rest-http/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/f7aee4936593c477d77410e43d07f991b50c06b3f51d7af53e191a31a5b08ef8/diff/opt/ignite/gr
idgain-professional-fabric-2.5.11/libs/optional/ignite-zookeeper/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/80d69aaefc80b5b24055f102af7d96f8d45b133968096f3009dd5a1e569e38e9/diff/opt/ignite/gr
idgain-professional-fabric-2.5.11/libs/ignite-rest-http/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/dea19ee9f30baf2aa736a9d26b3f4c24f1ac2b6ebdd4a48d4bb001ef13e86af5/diff/usr/share/jav
a/kafka/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/da20ec45451c54388661ef3017aada5f213f2c756af45ab27513ad0caa2a2305/diff/usr/share/jav
a/confluent-control-center/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/da20ec45451c54388661ef3017aada5f213f2c756af45ab27513ad0caa2a2305/diff/usr/share/jav
a/confluent-rebalancer/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/da20ec45451c54388661ef3017aada5f213f2c756af45ab27513ad0caa2a2305/diff/usr/share/jav
a/schema-registry/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/da20ec45451c54388661ef3017aada5f213f2c756af45ab27513ad0caa2a2305/diff/usr/share/jav
a/acl/acl-5.2.2.jar
```

```
/var/lib/docker/docker/overlay2/f7aee4936593c477d77410e43d07f991b50c06b3f51d7af53e191a31a5b08ef8/diff/opt/ignite/gr
idgain-professional-fabric-2.5.11/libs/optional/ignite-yarn/ignite-yarn-2.5.11.jar
/var/lib/docker/docker/overlay2/cbb5555681eb062d46da7779cbfca32ef864fbec9b972c836df6c49fddc991dc/diff/opt/wso2cep-
4.2.0/lib/org.wso2.carbon.logging-4.4.9.jar
/var/lib/docker/docker/overlay2/cbb5555681eb062d46da7779cbfca32ef864fbec9b972c836df6c49fddc991dc/diff/opt/wso2cep-
4.2.0/repository/components/plugins/org.wso2.carbon.logging_4.4.7.jar
/var/lib/docker/docker/overlay2/cbb5555681eb062d46da7779cbfca32ef864fbec9b972c836df6c49fddc991dc/diff/opt/wso2cep-
4.2.0/repository/components/plugins/org.wso2.carbon.logging_4.4.9.jar
/var/lib/docker/docker/overlay2/ba0af5a21a2edc833809240093585f10c32b5e0507334ad541119b724ac61b14/diff/usr/share/jav
a/kafka/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/f7aee4936593c477d77410e43d07f991b50c06b3f51d7af53e191a31a5b08ef8/diff/opt/ignite/gr
idgain-professional-fabric-2.5.11/benchmarks/libs/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/f7aee4936593c477d77410e43d07f991b50c06b3f51d7af53e191a31a5b08ef8/diff/opt/ignite/gr
idgain-professional-fabric-2.5.11/libs/optional/ignite-log4j/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/f7aee4936593c477d77410e43d07f991b50c06b3f51d7af53e191a31a5b08ef8/diff/opt/ignite/gr
idgain-professional-fabric-2.5.11/libs/optional/ignite-osgi-paxlogging/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/f7aee4936593c477d77410e43d07f991b50c06b3f51d7af53e191a31a5b08ef8/diff/opt/ignite/gr
idgain-professional-fabric-2.5.11/libs/optional/ignite-rest-http/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/f7aee4936593c477d77410e43d07f991b50c06b3f51d7af53e191a31a5b08ef8/diff/opt/ignite/gr
idgain-professional-fabric-2.5.11/libs/optional/ignite-zookeeper/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/80d69aaefc80b5b24055f102af7d96f8d45b133968096f3009dd5a1e569e38e9/diff/opt/ignite/gr
idgain-professional-fabric-2.5.11/libs/ignite-rest-http/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/dea19ee9f30baf2aa736a9d26b3f4c24f1ac2b6ebdd4a48d4bb001ef13e86af5/diff/usr/share/jav
a/kafka/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/da20ec45451c54388661ef3017aada5f213f2c756af45ab27513ad0caa2a2305/diff/usr/share/jav
a/confluent-control-center/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/da20ec45451c54388661ef3017aada5f213f2c756af45ab27513ad0caa2a2305/diff/usr/share/jav
a/confluent-rebalancer/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/da20ec45451c54388661ef3017aada5f213f2c756af45ab27513ad0caa2a2305/diff/usr/share/jav
a/schema-registry/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/da20ec45451c54388661ef3017aada5f213f2c756af45ab27513ad0caa2a2305/diff/usr/share/jav
a/acl/acl-5.2.2.jar
/var/lib/docker/docker/overlay2/f7aee4936593c477d77410e43d07f991b50c06b3f51d7af53e191a31a5b08ef8/diff/opt/ignite/gr
idgain-professional-fabric-2.5.11/libs/optional/ignite-yarn/ignite-yarn-2.5.11.jar
/var/lib/docker/docker/overlay2/cbb5555681eb062d46da7779cbfca32ef864fbec9b972c836df6c49fddc991dc/diff/opt/wso2cep-
4.2.0/lib/org.wso2.carbon.logging-4.4.9.jar
/var/lib/docker/docker/overlay2/cbb5555681eb062d46da7779cbfca32ef864fbec9b972c836df6c49fddc991dc/diff/opt/wso2cep-
4.2.0/repository/components/plugins/org.wso2.carbon.logging_4.4.7.jar
/var/lib/docker/docker/overlay2/cbb5555681eb062d46da7779cbfca32ef864fbec9b972c836df6c49fddc991dc/diff/opt/wso2cep-
4.2.0/repository/components/plugins/org.wso2.carbon.logging_4.4.9.jar
/var/lib/docker/docker/overlay2/ba0af5a21a2edc833809240093585f10c32b5e0507334ad541119b724ac61b14/diff/usr/share/jav
a/kafka/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/f7aee4936593c477d77410e43d07f991b50c06b3f51d7af53e191a31a5b08ef8/diff/opt/ignite/gr
idgain-professional-fabric-2.5.11/benchmarks/libs/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/f7aee4936593c477d77410e43d07f991b50c06b3f51d7af53e191a31a5b08ef8/diff/opt/ignite/gr
idgain-professional-fabric-2.5.11/libs/optional/ignite-log4j/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/f7aee4936593c477d77410e43d07f991b50c06b3f51d7af53e191a31a5b08ef8/diff/opt/ignite/gr
idgain-professional-fabric-2.5.11/libs/optional/ignite-osgi-paxlogging/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/f7aee4936593c477d77410e43d07f991b50c06b3f51d7af53e191a31a5b08ef8/diff/opt/ignite/gr
idgain-professional-fabric-2.5.11/libs/optional/ignite-rest-http/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/f7aee4936593c477d77410e43d07f991b50c06b3f51d7af53e191a31a5b08ef8/diff/opt/ignite/gr
idgain-professional-fabric-2.5.11/libs/optional/ignite-zookeeper/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/80d69aaefc80b5b24055f102af7d96f8d45b133968096f3009dd5a1e569e38e9/diff/opt/ignite/gr
idgain-professional-fabric-2.5.11/libs/ignite-rest-http/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/dea19ee9f30baf2aa736a9d26b3f4c24f1ac2b6ebdd4a48d4bb001ef13e86af5/diff/usr/share/jav
a/kafka/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/da20ec45451c54388661ef3017aada5f213f2c756af45ab27513ad0caa2a2305/diff/usr/share/jav
a/confluent-control-center/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/da20ec45451c54388661ef3017aada5f213f2c756af45ab27513ad0caa2a2305/diff/usr/share/jav
a/confluent-rebalancer/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/da20ec45451c54388661ef3017aada5f213f2c756af45ab27513ad0caa2a2305/diff/usr/share/jav
a/schema-registry/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/da20ec45451c54388661ef3017aada5f213f2c756af45ab27513ad0caa2a2305/diff/usr/share/jav
a/acl/acl-5.2.2.jar
/var/lib/docker/docker/overlay2/f7aee4936593c477d77410e43d07f991b50c06b3f51d7af53e191a31a5b08ef8/diff/opt/ignite/gr
idgain-professional-fabric-2.5.11/libs/optional/ignite-yarn/ignite-yarn-2.5.11.jar
/var/lib/docker/docker/overlay2/cbb5555681eb062d46da7779cbfca32ef864fbec9b972c836df6c49fddc991dc/diff/opt/wso2cep-
4.2.0/lib/org.wso2.carbon.logging-4.4.9.jar
/var/lib/docker/docker/overlay2/cbb5555681eb062d46da7779cbfca32ef864fbec9b972c836df6c49fddc991dc/diff/opt/wso2cep-
4.2.0/repository/components/plugins/org.wso2.carbon.logging_4.4.7.jar
/var/lib/docker/docker/overlay2/cbb5555681eb062d46da7779cbfca32ef864fbec9b972c836df6c49fddc991dc/diff/opt/wso2cep-
4.2.0/repository/components/plugins/org.wso2.carbon.logging_4.4.9.jar
/var/lib/docker/docker/overlay2/ba0af5a21a2edc833809240093585f10c32b5e0507334ad541119b724ac61b14/diff/usr/share/jav
a/kafka/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/f7aee4936593c477d77410e43d07f991b50c06b3f51d7af53e191a31a5b08ef8/diff/opt/ignite/gr
idgain-professional-fabric-2.5.11/benchmarks/libs/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/f7aee4936593c477d77410e43d07f991b50c06b3f51d7af53e191a31a5b08ef8/diff/opt/ignite/gr
idgain-professional-fabric-2.5.11/libs/optional/ignite-log4j/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/f7aee4936593c477d77410e43d07f991b50c06b3f51d7af53e191a31a5b08ef8/diff/opt/ignite/gr
idgain-professional-fabric-2.5.11/libs/optional/ignite-osgi-paxlogging/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/f7aee4936593c477d77410e43d07f991b50c06b3f51d7af53e191a31a5b08ef8/diff/opt/ignite/gr
idgain-professional-fabric-2.5.11/libs/optional/ignite-rest-http/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/f7aee4936593c477d77410e43d07f991b50c06b3f51d7af53e191a31a5b08ef8/diff/opt/ignite/gr
idgain-professional-fabric-2.5.11/libs/optional/ignite-zookeeper/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/80d69aaefc80b5b24055f102af7d96f8d45b133968096f3009dd5a1e569e38e9/diff/opt/ignite/gr
idgain-professional-fabric-2.5.11/libs/ignite-rest-http/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/dea19ee9f30baf2aa736a9d26b3f4c24f1ac2b6ebdd4a48d4bb001ef13e86af5/diff/usr/share/jav
a/kafka/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/da20ec45451c54388661ef3017aada5f213f2c756af45ab27513ad0caa2a2305/diff/usr/share/jav
a/confluent-control-center/log4j-1.2.17.jar
```

```
/var/lib/docker/docker/overlay2/da20ec45451c54388661ef3017aada5f213f2c756af45ab27513ad0caa2a2305/diff/usr/share/jav
a/confluent-rebalancer/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/da20ec45451c54388661ef3017aada5f213f2c756af45ab27513ad0caa2a2305/diff/usr/share/jav
a/schema-registry/log4j-1.2.17.jar
/var/lib/docker/docker/overlay2/da20ec45451c54388661ef3017aada5f213f2c756af45ab27513ad0caa2a2305/diff/usr/share/jav
a/acl/acl-5.2.2.jar
/var/lib/docker/docker/overlay2/f7aee4936593c477d77410e43d07f991b50c06b3f51d7af53e191a31a5b08ef8/diff/opt/ignite/gr
idgain-professional-fabric-2.5.11/libs/optional/ignite-yarn/ignite-yarn-2.5.11.jar
/var/lib/docker/docker/overlay2/cbb5555681eb062d46da7779cbfca32ef864fbec9b972c836df6c49fddc991dc/diff/opt/wso2cep-
4.2.0/lib/org.wso2.carbon.logging-4.4.9.jar
/var/lib/docker/docker/overlay2/cbb5555681eb062d46da7779cbfca32ef864fbec9b972c836df6c49fddc991dc/diff/opt/wso2cep-
4.2.0/repository/components/plugins/org.wso2.carbon.logging_4.4.7.jar
/var/lib/docker/docker/overlay2/cbb5555681eb062d46da7779cbfca32ef864fbec9b972c836df6c49fddc991dc/diff/opt/wso2cep-
4.2.0/repository/components/plugins/org.wso2.carbon.logging_4.4.9.jar


Removal of vulnerable classes from the log4j/lockback/slf4j jars packaged in SANnav services jar files may take 20-
30 minutes, wait for the completion.
grep:
/var/lib/docker/docker/overlay2/f0935cc2ad7687e0820a4533aac2bd274b079768863bdee8895510b7032ffbb7/diff/usr/share/jav
a/apache-ant/lib/junit.jar: No such file or directory
grep:
/var/lib/docker/docker/overlay2/80832bc05353cab816c95267a17342b6d49ca0a613462b7fc04f4c013634c624/diff/usr/share/ela
sticsearch/jdk/legal/jdk.jartool: Is a directory

zip error: Nothing to do! (collections-middleware-1.0.0.jar)

zip error: Nothing to do! (objectmanager-1.0.0.jar)

zip error: Nothing to do! (system-monitor-1.0.0.jar)

zip error: Nothing to do! (perfmon-collecter-service-1.0.0.jar)
Successfully patched the log4j-1.2.17.jar,logback-classic-1.1.11.jar,slf4j-log4j12-1.7.25.jar from SANnav.

Start Docker Service and sleep for 30 seconds.

Successfully started the docker service.

Starting the SANnav Services.

Deleting the Zookeeper and KAFKA data directories...

Starting SANnav infrastructure services

dcm 2 1 1 dcm-postgres-db scaled to 1
dcm_2_1_1_zookeeper scaled to 1
dcm_2_1_1_kafka-1 scaled to 1
dcm_2_1_1_schema-registry scaled to 1
dcm 2 1 1 elasticsearch scaled to 1
dcm_2_1_1_ignite-grid-node1 scaled to 1

Starting SANnav services

dcm_2_1_1_ignite-grid-object-manager-node scaled to 1
dcm_2_1_1_authentication-rbac-middleware scaled to 1
dcm_2_1_1_dashboard-middleware scaled to 1
dcm_2_1_1_dashboard-summaryprovider scaled to 1
dcm_2_1_1_faultmanagement-backend scaled to 1
dcm_2_1_1_faultmanagement-collector scaled to 1
dcm_2_1_1_dcm-wso2 scaled to 1
dcm_2_1_1_fe-consolidated-1 scaled to 1
dcm_2_1_1_fe-consolidated-2 scaled to 1
dcm_2_1_1_switch-asset-collectors-mw scaled to 1
dcm_2_1_1_performancemanagement-middleware scaled to 1
dcm_2_1_1_performancemanagement-datastore scaled to 1
dcm_2_1_1_performancemanagement-statscollector scaled to 1
dcm_2_1_1_cfgmgmt-policy-middleware scaled to 1
dcm_2_1_1_topology-middleware scaled to 1
dcm_2_1_1_filters-contextsearch-middleware scaled to 1
dcm_2_1_1_reportgenerator scaled to 1
dcm_2_1_1_asyncjobscheduler-manager scaled to 1
dcm_2_1_1_asyncjobscheduler-worker scaled to 1
dcm_2_1_1_g-agent scaled to 1
dcm 2 1 1 supportsave-mw scaled to 1
dcm_2_1_1_filetransfer scaled to 1
dcm_2_1_1_license-mw scaled to 1
dcm 2 1 1 backuprestore-mw scaled to 1
dcm_2_1_1_switch-filetransfer-mw scaled to 1
dcm_2_1_1_externalapi-middleware scaled to 1
dcm_2_1_1_proxy scaled to 1
dcm_2_1_1_system-monitor scaled to 1
dcm_2_1_1_troubleshooting scaled to 1
dcm_2_1_1_flow-management-mw scaled to 1
dcm_2_1_1_collections-middleware scaled to 1

Starting SANnav monitoring services

dcm_2_1_1_cadvisor scaled to 1
dcm_2_1_1_nodeexporter scaled to 1
dcm_2_1_1_prometheus scaled to 1

SANnav server has been successfully started.
```

```
SANnav server startup may take up to 15 minutes.
To check SANnav server status, run /data/Portal_2.1.1_bld184/bin/check-sannav-status.sh.
When startup has completed, launch the client using [https://192.168.226.133].

Successfully removed all log4j chainsaw,JndiLookup,SMTPAppender,JMSAppender,JMSSink,SocketServer and JDBCAppender
classes from the SANnav and started SANnav services.To verify the SANnav service status, please use the script
check-sannav-status.sh after 12-15 minutes.

[root@sannav bin]#
```

Check to see if all SANnav services started.. If it returns nothing then everything is up and you should be fine.

```
[root@sannav bin]# ./check-sannav-status.sh
[root@sannav bin]#
```

If you need to check the progress of the patch installation or debug if it failed, you can refer to the patch log file which is located in <installation_home>/logs directory.

```
[root@sannav logs]# pwd
/data/Portal_2.1.1_bld184/logs
[root@sannav logs]# ls -alt | more
total 28620
-rw-r--r--.  1 root root  28512 Mar 18 20:43 fix_log4j_vulnerability_2022_03_18_20_39_47_PM.log
-rw-r--r--.  1 root root     97 Mar 18 20:40 stop-sannav-2022_03_18_2039.log
drwxr-xr-x. 12 root root  16384 Mar 18 20:39 .
-rw-r--r--.  1 root root   1345 Mar 18 20:39 patch-deploy-portal 2.1.1.7 2022 03 18 2038.log
-rw-r--r--.  1 root root   5973 Mar 18 20:38 stop-sannav-2022_03_18_2038.log
-rw-r--r--.  1 root root 730468 Mar 18 20:38 system-monitor.log
drwxr-xr-x. 23 root root   4096 Mar 18 20:38 ..
-rw-r--r--.  1 root root  20917 Mar 18 20:37 system-monitor-2022-03-18_20-37-08-gc.log.0.current
drwxr-xr-x.  2 root root   4096 Mar 18 20:37 ignite-grid-node1
-rw-r--r--.  1 root root   4760 Mar 18 20:37 reportgenerator-2022-03-18.log
-rw-r--r--.  1 root root 267462 Mar 18 20:36 topology-middleware-2022-03-18_19-24-40-gc.log.0.current
-rw-r--r--.  1 root root  73619 Mar 18 20:36 topology-middleware.log
-rw-r--r--.  1 root root 155759 Mar 18 20:36 performancemanagement-statscollector-2022-03-18_19-24-33-
gc.log.0.current
.
.
.
[root@sannav logs]# cat patch-deploy-portal_2.1.1.7_2022_03_18_2038.log
DCM_HOME:/data/Portal_2.1.1_bld184
Build Number is bld007
FOLDER_NAME is Portal_2.1.1.7_bld007
Stopping the server:/data/Portal_2.1.1_bld184
be-faultmanagement:sann2.1.1.7
be-faultmanagement
Docker service ID: vfdm5zgtnxgr
Loading the DCM services images from File System
Successfully loaded the DCM services images from File System
fe-consolidated-1:sann2.1.1.7
fe-consolidated-1
Docker service ID: l3fuczo9zqli
Loading the DCM services images from File System
Successfully loaded the DCM services images from File System
fe-troubleshooting:sann2.1.1.7
fe-troubleshooting
Docker service ID: plx5ow9918fs
Loading the DCM services images from File System
Successfully loaded the DCM services images from File System
mw-switch-filetransfer:sann2.1.1.7
mw-switch-filetransfer
Docker service ID: 9oq5w8uj857h
Loading the DCM services images from File System
Successfully loaded the DCM services images from File System
proxy-local:sann2.1.1.7
proxy-local
Docker service ID: j0kkxhpuf3o3
Loading the DCM services images from File System
Successfully loaded the DCM services images from File System
switch-asset-collectors-mw:sann2.1.1.7
switch-asset-collectors-mw
Docker service ID: qqfhmug8x51v
Loading the DCM services images from File System
Successfully loaded the DCM services images from File System
Executing fix_log4j_vulnerability.sh script
[root@sannav logs]#
```

And that's about it. Good work if you made it till here!

Summary

Whilst this document might be trivial to those that have the necessary platform experience.. For those that don't I am sure that it will provide a useful reference. We have covered the base product installation as well as applying the latest patch. Brocade offer great FREE product training on ALL their products including SANnav. Register for a Brocade ID and then enroll on any of these courses available at the following URL: https://www.broadcom.com/support/education/brocade/courses#sannav-management-applications.