D. Delmar Davis

Notes on Systems

None

Table of contents

1. Digithink.com.

I'll be 72 when unix time ends.

After cutting my teeth on Dec VMS systems in college, I worked on my first unix systems in the late 80s.
I am still working in linux.
Collected here are some notes past and present.

And here is my resume

Did you think you'd be doing this for more than 30 years?
Maybe its time to rethink everything!

- Download site as a pdf

2. Buildnotes

2.1 Install motion on raspberry pi running debian bookworm

Install and check the status of the motion service

```
wget https://github.com/Motion-Project/motion/releases/download/release-4.6.0/pi_bookworm_motion_4.6.0-1_arm64.deb
dpkg -i pi_bookworm_motion_4.6.0-1_arm64.deb
apt --fix-broken install
systemctl status motion
```

Make sure the camera works and adjust firmware config based on what libcamera has to say about it.

```
libcamera-vid -t 20000 libcamera-vid --codec libav -o test.mp4
rpicam-still -v -o test.jpg
rpicam-vid -t 10s --codec libav -o test.mp4
nano /boot/firmware/config.txt
...
dtoverlay=ov5647
...
reboot
```

Check that the camera still works (also check motion service)

```
rpicam-vid -t 10s --codec libav -o test.mp4
rpicam-hello
systemctl status motion
```

Install libcamerify and edit the motion.service definition.

```
apt install libcamera-tools libcamera-dev libcamera-v4l2 libcamerify
nano /lib/systemd/system/motion.service
...
[Service]
User=motion
UMask=002
EnvironmentFile=-/etc/default/motion
#ExecStart=/usr/bin/motion
ExecStart=/usr/bin/libcamerify /usr/bin/motion
...
systemctl restart motion
systemctl daemon-reload
systemctl restart motion
systemctl status motion
```

Make the system allow remote connections and fix location for output.

```
nano /etc/motion/motion.conf
...
# Target directory for pictures, snapshots and movies
target_dir /var/lib/motion
...
# Restrict webcontrol connections to the localhost.
webcontrol_localhost off
...
# Restrict stream connections to the localhost.
stream_localhost off
...
^X
systemctl restart motion
systemctl status motion
netstat -tunlp
```

Probably want to put mount /var/lib/motion on a fileserver.

```
mount utah:/tank/motion/buster /var/lib/motion
grep motion /etc/mtab>>/etc/fstab
mount -a
```

2.1.1 References

- https://forums.raspberrypi.com/viewtopic.php?t=359023
- https://pimylifeup.com/raspberry-pi-webcam-server/

- https://github.com/Motion-Project/motion/discussions/1753
- https://www.raspberrypi.com/documentation/computers/camera_software.html#getting-started

2.2 Need to move the colocated systems to a different network. ASAP.

WORK IN PROGRESS -- Documenting as we go....

2.2.1 Basic Process

- get the new ip addresses
- put dns zone files into a repo (done)
- set dns ttls to be small. (600=10m) (done)
- add new dns server to ns records for digithink.com (done)
- stand up dns server and connect it to the new ip address range. (done)
- add the new server to the upstream (web.com) (done)
- set up remaining dns nodes to pull from new server (done)
- add new servers to upstream (web.com) (done)
- move static websites first (done)
- move dns (wait 10 minutes)
- move ip configurations to new ips
- repeat above to move remaining servers ending with the mail server. (done)
- move the mail (done)
- test the mail (done)
- clean up all references to 198.202.31

Put the dns zone files into a repo

- Create a blank repo in bitbucket and create an access token
- Clone the repo into /etc/bind/zones using the access token
- Move the .git folder into place and clean up.
- Add contents of /etc/bind/zones to the repo.
- Commit and push it.

```
cd /etc/bind/zones
git clone https://x-token-auth:REDACTED@bitbucket.org/suspectdevicesadmin/susdev-dns.git
ls -lsa susdev-dns/
mv susdev-dns/.git .
rmdir susdev-dns/
ls
git add *
git status
git commit -a -m"first checkin"
git push
cp named.conf.local zones
git add zones/named.conf.local
git commit -a -m"add the named.conf.local"
git push
```

Stand up dns new server (dns.digithink.com) and connect it to the new ip address range. (done)

Start a new bookworm container.

```
incus init bookworm piage -p default -p susdev24
incus start piage
incus exec piage bash
```

Configure the network

```
ip a
apt install nano
cd /etc/
grep -ri 198.202.31.200
nano systemd/network/10-cloud-init-eth0.network
reboot
ip a
ping digithink.com
```

Pull down zone files from git and configure bind9 to use them.

```
cd /etc/bind
git clone https://x-token-auth:REDACTED@bitbucket.org/suspectdevicesadmin/susdev-dns.git zones
mv named.conf.local /tmp/
ln -s zones/named.conf.local .
nano master.conf
systemctl reload named
systemctl status named.service
git status
git commit -a -m "delete unused domains"
git config user.email REDACTED@bots.bitbucket.org
git commit -a -m "delete unused domains"
git push
```

add the new server to the upstream

Use the web.com site to update the addresses of your authoritative dns servers

set up remaining dns nodes to pull from new server

on tk

```
incus exec teddy bash
```

then adjust the secondary dns server.

```
nano /etc/systemd/network/10-cloud-init-eth0.network
reboot
cd /etc/named/zones
sed -i s/198.202.31.141/69.41.138.98/ slave.conf
systemctl reload named
systemctl status named
```

While you're there clean up the rest of the 198.202.31.

```
grep -r 198.202.31 /etc
sed -i s/198.202.31.98/69.41.138.98/ /etc/resolv.conf.static
grep -r 198.202.31 /etc
exit
```

Migrate vpn nodes.

VIRGIL.

Set dns entries for wireguard hosts then adjust their ips.

```
grep -r 198.202.31. /etc/
nano /etc/netplan/50-cloud-init.yaml
network:
    version: 2
    ethernets:
        eth0:
            addresses:
            - 69.41.138.125/27
            nameservers:
                addresses:
                - 69.41.138.99
                - 8.8.8.8
                search: []
            routes:
            -   to: default
                via: 69.41.138.97
        eth1:
            addresses:
            - 192.168.31.228/24
root@virgil:~#
^x
```

```
netplan apply
ip a
sed -i s/198.202.31.132/69.41.138.99/ /etc/resolv.conf
root@virgil:~# sed -i s/198.202.31.132/69.41.138.99/ /etc/resolv.conf.static
reboot
```

SITKA

Adjust the ip address of the public interface and the name servers

```
nano /etc/rc.conf
hostname="sitka"

ifconfig_igb4="69.41.138.126 netmask 255.255.255.224"
defaultrouter="69.41.138.97"
...
^X y
cp resolv.conf /tmp/
sed s/198.202.31.141/69.41.138.98/ /tmp/resolv.conf|sed s/198.202.31.132/69.41.138.99/>/etc/resolv.conf

reboot
```

Adjust the ip address of master dns server.

```
cd /usr/local/etc/namedb/zones
cp slave.conf /tmp/
sed s/198.202.31.141/69.41.138.98/ /tmp/slave.conf >slave.conf
service named restart
tail /var/log/messages
```

move static websites

- move dns (wait 10 minutes)

YOU ARE HERE Giving the cliffnotes version

- move ip configurations to new ips

YOU ARE HERE Giving the cliffnotes version

repeat above to move remaining servers ending with the mail server. (done)

move the mail (done)

YOU ARE HERE Giving the cliffnotes version

test the mail (done)

clean up all references to 198.202.31

```
for c in `incus list -cn -f compact|grep -v NAME`; do echo $c ;incus exec $c -- grep -r 198.202.31. /etc/; done ; echo `hostname`; grep -r 198.202.31. /etc/
... clean up all container references ...
... then clean up tk ...
cd /etc/ansible/
mkdir profiles
incus profile show susdev23>profiles/susdev23.yaml
incus profile show susdev24>profiles/susdev24.yaml
sed -i s/198.202.31.200/69.41.138.113/ profiles/susdev23.yaml
sed -i s/198.202.31.129/69.41.138.97/ profiles/susdev23.yaml
sed -i s/198.202.31.141/69.41.138.99/ profiles/susdev23.yaml
sed -i s/198.202.31.132/69.41.138.98/ profiles/susdev23.yaml
sed -i s/255.255.255.128/255.255.255.244/ profiles/susdev23.yaml
cat profiles/susdev23.yaml |incus profile edit susdev23
incus profile show susdev24>profiles/susdev24.yaml
sed -i s/198.202.31.200/69.41.138.113/ profiles/susdev24.yaml
sed -i s/198.202.31.129/69.41.138.97/ profiles/susdev24.yaml
sed -i s/198.202.31.141/69.41.138.98/ profiles/susdev24.yaml
sed -i s/198.202.31.132/69.41.138.99/ profiles/susdev24.yaml
sed -i s/255.255.255.128/255.255.255.244/ profiles/susdev24.yaml
cat profiles/susdev24.yaml |incus profile edit susdev24
grep -r 198.202.31. .
mv files/merlot.profile.yaml profiles/merlot.yaml
sed -i s/198.202.31.160/69.41.138.120/ profiles/merlot.yaml
sed -i s/198.202.31.141/69.41.138.98/ profiles/merlot.yaml
grep -r 198.202.31. .
```

```
sed -i s/198.202.31.160/69.41.138.120/ README.md
git status
git add files README.md playbooks roles
git add profiles/
git commit -a -m migration
git push
grep -r 198.202.31. .
```

2.3 Guthrie rebuild -- Rinse Lather Repeat.

Guthrie is Utah's partner in home service. It is also the last ubuntu server and since it does my girlfriends backup its probably the most important.

Install incus and convert the lxd containers to incus.

```
curl  -fsSL https://pkgs.zabbly.com/key.asc -o /etc/apt/keyrings/zabbly.asc
sh -c 'cat <<EOF > /etc/apt/sources.list.d/zabbly-incus-stable.sources
Enabled: yes
Types: deb
URIs: https://pkgs.zabbly.com/incus/stable
Suites: $(. /etc/os-release && echo ${VERSION_CODENAME})
Components: main
Architectures: $(dpkg --print-architecture)
Signed-By: /etc/apt/keyrings/zabbly.asc

EOF'
apt update
apt install incus
systemctl start incus
lxd-to-incus --ignore-version-check --yes
systemctl restart incus
incus storage list
```

establishing remotes and trust between nodes

GUTHRIE TO UTAH

Generate trust token

```
root@utah:~# incus config trust add guthrie
Client guthrie certificate add token:
eyJjbGllbnRfbmFtZSI6Imd1dGhyaWUiLCJmaW5nZXJwcmludCI6IjgwMDVmNDliNTdkMTUyMDZlNjI4MDE3M2YzNTljZjI1NjVhNDU2OWQ5MzkxMjExZWRhNDllNWMxNDFkNWU0MTIiLCJhZGRyZXNzZXMiOlsi
```

Use it to add remote

```
root@guthrie:~# incus remote add utah
Generating a client certificate. This may take a minute...
Certificate fingerprint: 8005f49b57d15206e6280173f359cf2565a4569d9391211eda49e5c141d5e412
ok (y/n/[fingerprint])? y
Trust token for utah: <paste token from above>
Client certificate now trusted by server: utah
root@guthrie:~# incus list utah:
+-------------+---------+------------------------------+------+-----------+-----------+
|    NAME     |  STATE  |             IPV4             | IPV6 |   TYPE    | SNAPSHOTS |
+-------------+---------+------------------------------+------+-----------+-----------+
| bunnyfoofoo | RUNNING | 192.168.128.152 (eth0)       |      | CONTAINER | 0         |
+-------------+---------+------------------------------+------+-----------+-----------+
| haley       | RUNNING | 192.168.129.198 (eth0)       |      | CONTAINER | 0         |
|             |         | 172.18.0.1 (br-a4e161520842) |      |           |           |
|             |         | 172.17.0.1 (docker0)         |      |           |           |
+-------------+---------+------------------------------+------+-----------+-----------+
```

UTAH TO GUTHRIE

Generate trust certificate

```
root@guthrie:~# incus config trust add utah
Client utah certificate add token:
eyJjbGllbnRfbmFtZSI6InV0YWgiLCJmaW5nZXJwcmludCI6ImJmZTkxMmFkODgzNmE3NDU3NDIwNTA2ZmM4ZTEzMzY5NTM0MDc3ZmU2NDcxODQzMDkzNjMxMTdjYTU4MDFhZDQiLCJhZGRyZXNzZXMiOlsiMTky
```

Use it to add remote

```
root@utah:~# incus remote add guthrie
Certificate fingerprint: bfe912ad8836a7457420506fc8e13369534077fe647184309363117ca5801ad4
ok (y/n/[fingerprint])? yes
Trust token for guthrie: <Paste token from above>
Client certificate now trusted by server: guthrie
root@utah:~# incus list guthrie:
+-----------+---------+------------------------------+------+-----------------+-----------+
|   NAME    |  STATE  |             IPV4             | IPV6 |      TYPE       | SNAPSHOTS |
+-----------+---------+------------------------------+------+-----------------+-----------+
| annie     | RUNNING | 192.168.129.110 (enp5s0)     |      | VIRTUAL-MACHINE | 0         |
+-----------+---------+------------------------------+------+-----------------+-----------+
| dick      | RUNNING | 192.168.129.183 (enp5s0)     |      | VIRTUAL-MACHINE | 1         |
+-----------+---------+------------------------------+------+-----------------+-----------+
```

```
| gail      | RUNNING | 192.168.129.192 (eth0)       |      | CONTAINER       | 3         |
|           |         | 172.18.0.1 (br-a4e161520842) |      |                 |           |
|           |         | 172.17.0.1 (docker0)         |      |                 |           |
+-----------+---------+------------------------------+------+-----------------+-----------+
| jobs      | RUNNING | 192.168.129.187 (eth0)       |      | CONTAINER       | 0         |
+-----------+---------+------------------------------+------+-----------------+-----------+
| katherine | RUNNING | 192.168.129.188 (eth0)       |      | CONTAINER       | 0         |
+-----------+---------+------------------------------+------+-----------------+-----------+
| luigi     | RUNNING | 192.168.129.250 (eth0)       |      | CONTAINER       | 0         |
+-----------+---------+------------------------------+------+-----------------+-----------+
```

Temporarily transfer containers to Utah

luigi should be started since unlike the other containers it isnt providing a service tied to guthries storage.

```
root@utah:~# incus stop guthrie:luigi\
 && incus move guthrie:luigi luigi\
 && incus start luigi
```

Stop remaining containers and move them.

```
root@utah:~# incus stop guthrie:annie
root@utah:~# incus move guthrie:annie annie
root@utah:~# incus stop guthrie:dick
root@utah:~# incus move guthrie:dick dick
... continue for all containers on guthrie...
```

2.3.1 Install debian on guthrie.

Prep

FREE UP THE TARGET DISK (CURRENTLY MIRRORING THE UBUNTU ZFSROOT)

```
parted /dev/nvme0n1 print
zpool status -L
zfs status local
zpool status local
zpool detach local nvme-TEAM_TM8FP6002T_TPBF2306190020902551-part5
zpool status -L rpool
zpool status rpool
zpool detach nvme-TEAM_TM8FP6002T_TPBF2306190020902551-part4
zpool detach rpool zpool status bpool -L
zpool detach bpool zpool detach bpool 1f1027f1-c44d-7f42-a71b-8bdff56b3a51
```

get rid of any zfs stuff. The debian installer (for bookworm anyways) tends to freak out and not want to install on anything with zfs on it.

```
wipefs -fa /dev/nvme0n1
```

save some stuff for later

```
df -k
parted /dev/nvme0n1 print
mkdir /tank/oldguthrie/
cp /etc/netatalk/afp.conf /tank/oldguthrie/
cp -rpv /etc/ssh /tank/oldguthrie/
mkdir /tank/oldguthrie/root
cp -rpv .bash_history .bashrc .config .local .profile .selected_editor .ssh zplan go /tank/oldguthrie/root/
```

Install stuff you will want installed.

```
apt install openssh-server
apt install ca-certificates
apt install curl
apt install gpg
apt install sudo
apt install parted
apt install htop
apt install bridge-utils
```

start the sshd and go upstairs to finish this.

```
systemctl enable sshd
systemctl start sshd
```

get rid of the graphical runtime before the system goes to sleep.

```
systemctl set-default multi-user.target
reboot
```

set up sudo

```
visudo
.... add feurig ....
```

Install zfs from trixie

```
apt -t bookworm-backports install zfs-dkms zfs-zed zfsutils-linux
modprobe zfs
```

Import the data pools.

```
zpool import
zpool import -f archive
zpool import -f home
zpool import -f tank
zpool import -f filebox
```

Set up networking as it was before

Forgot a few things mount the old ubuntu root

```
mkdir /tmp/mnt
zpool import
zpool import -f rpool orpool
zfs set mountpoint=/tmp/mnt orpool/ROOT/ubuntu_lrmg80
zfs mount orpool/ROOT/ubuntu_lrmg80
cp /tmp/mnt/etc/netplan/00-merlot.yaml /tank/oldguthrie/
```

Set /etc/network/interfaces

```
cat /etc/network/interfaces|sed
ip a|sed 's/^/# /'
ip a|sed 's/^/# />/etc/network/interfaces
nano /etc/network/interfaces
auto lo
iface lo inet loopback

allow-hotplug enp9s0
auto enp9s0
#iface enp10s0 inet dhcp
iface enp9s0 inet manual

auto br0
iface br0 inet static
     address 192.168.129.182
     network 192.168.128.0
     netmask 255.255.128.0
     broadcast 192.168.255.255
     gateway 192.168.128.1
     mtu 9000
     bridge_ports enp10s0
     bridge_stp off       # disable Spanning Tree Protocol
        bridge_waitport 0    # no delay before a port becomes available
        bridge_fd 0          # no forwarding delay
^X
reboot
```

Install incus from zabbly repo.

```
curl  -fsSL https://pkgs.zabbly.com/key.asc -o /etc/apt/keyrings/zabbly.asc
sh -c 'cat <<EOF > /etc/apt/sources.list.d/zabbly-incus-stable.sources
Enabled: yes
Types: deb
URIs: https://pkgs.zabbly.com/incus/stable
Suites: $(. /etc/os-release && echo ${VERSION_CODENAME})
Components: main
Architectures: $(dpkg --print-architecture)
Signed-By: /etc/apt/keyrings/zabbly.asc
EOF'
apt update
apt install incus
```

Clear all of the zfs data from ubuntu disk otherwise incus will bitch about the old local pool

```
ls -ls /dev/disk/by-id|grep nvme1n1
zpool export orpool
wipefs -af /dev/disk/by-id/nvme-T-CREATE_TM8FPE002T_112108250100368
```

Initialize incus - create new local pool (with above device) - use existing bridge (br0)

```
incus admin init
systemctl enable incus
systemctl start incus
```

Add utah to /etc/hosts and generate trust key as above.

```
nano /etc/hosts
incus config trust add utah
```

on utah remove old remote, generate trust key, and re add guthrie as a remote

```
incus remote remove guthrie
incus config trust add guthrie
incus remote add guthrie
incus list guthrie:
```

Add utah as a remote.

```
incus remote add utah
incus list utah:
```

Move containers back as above.

Install netatalk 4 from netatalk.io

```
wget https://github.com/Netatalk/netatalk/releases/download/netatalk-4-0-0/netatalk_4.0.0.ds-1_amd64.deb
apt install ./netatalk_4.0.0.ds-1_amd64.deb
cp /tank/oldguthrie/afp.conf /etc/netatalk/afp.conf
systemctl enable netatalk
systemctl start netatalk
systemctl status netatalk
apt install avahi-daemon
apt enable avahi-daemon
systemctl enable avahi-daemon
systemctl restart avahi-daemon
systemctl status avahi-daemon
systemctl restart netatalk
```

Wrapup

- test backups on girlfriends system
- test containers

2.4 Craig Johnson -- rebuild static http sites on Debian.

2.4.1 ORGANIZE THIS PILE

Go old school on the static network configuration.

Systemd/networkd is coming but I want something that works right now.

/etc/network/interfaces aint broken.

So tear out all the new and replace it with the old.

```
systemctl stop systemd-networkd
systemctl disable systemd-networkd
systemctl stop systemd-networkd.socket
systemctl disable systemd-networkd.socket
apt install ifupdown
```

THEN CONFIGURE IT LIKE IT WAS A DECADE AGO.

```
cat /etc/network/interfaces
auto lo
iface lo inet loopback


auto eth0
iface eth0 inet static
     address 198.202.31.221
     network 198.202.31.128
     netmask 255.255.255.128
     broadcast 198.202.31.255
     gateway 198.202.31.129
     mtu 9000

auto eth0:1
iface eth0:1 inet static
     address 198.202.31.230
     network 198.202.31.128
     netmask 255.255.255.128
     broadcast 198.202.31.255
     gateway 198.202.31.129
     mtu 9000

auto eth0:2
iface eth0:2 inet static
     address 198.202.31.231
     network 198.202.31.128
     netmask 255.255.255.128
     broadcast 198.202.31.255
     gateway 198.202.31.129
     mtu 9000


auto eth0:3
iface eth0:3 inet static
     address 198.202.31.232
     network 198.202.31.128
     netmask 255.255.255.128
     broadcast 198.202.31.255
     gateway 198.202.31.129
     mtu 9000
```

2.4.2

nginx configuration

NEW CONFIG FOR WWW.3DANGST.COM (DEFAULT)

```
server {

    #listen 443 ssl 198.202.31.221;
    root /var/www/3dangst/site;
    index index.html;

    server_name www.3dangst.com;

    location / {
        try_files $uri $uri/ =404;
    }
```

```
    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/www.3dangst.com/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/www.3dangst.com/privkey.pem; # managed by Certbot
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

}

server {
    if ($host = www.3dangst.com) {
        return 301 https://$host$request_uri;
    } # managed by Certbot


    listen 80 default_server;

    server_name www.3dangst.com;
    return 404; # managed by Certbot


}
```

COPY NGINX CONFIG, /ETC/LETSENCRYPT AND CONTENT (/VAR/WWW/*) FROM THE OLD SERVER

We copied the old servers default to /etc/nginx/sites-avaliable/digithink and then linked it into sites-enabled.

```
cat /etc/nginx/sites-avaliable/digithink
server {

    listen 198.202.31.230:80;
    server_name www.digithink.com;

    if ($host = www.digithink.com) {
        return 301 https://$host$request_uri;
    } # managed by Certbot

    if ($host = 198.202.31.230) {
        return 444;
    } # managed by Certbot

    return 404; # managed by Certbot
}


server {
    listen 198.202.31.230:80;
    server_name  www.digithink.com;
    root /var/www/digithink/site;
    index index.html;

    listen 198.202.31.230:443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/www.digithink.com/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/www.digithink.com/privkey.pem; # managed by Certbot
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

    if ($host = 198.202.31.230) {
       return 444;
    } # managed by Certbot

   error_page 404 /404.html;
   location  /404.html {
      internal;
   }

}

upstream bartender {
    server 127.0.0.1:5000;
}

server {
    server_name bartender.digithink.com;

    listen 198.202.31.232:443 ssl;
    server_name bartender.digithink.com;
    ssl_certificate /etc/letsencrypt/live/bartender.digithink.com/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/bartender.digithink.com/privkey.pem; # managed by Certbot

    root /var/www/digithink/whiskey/bartender;
    index index.html;
    location /whiskey {
        include proxy_params;
        proxy_pass http://bartender/whiskey;
    }

    error_page 404 /404.html;
    location  /404.html {
          internal;
    }

    location /lacuenta {
```

```
        root /var/www/digithink/whiskey/logs;
    }


}

server {
    if ($host = bartender.digithink.com) {
        return 301 https://$host$request_uri;
    } # managed by Certbot

    listen 198.202.31.232:80;
    server_name bartender.digithink.com;
    return 404; # managed by Certbot
}


server {
        listen 198.202.31.231:80;
        server_name busholini.org w.busholini.org www.busholini.org;
        if ($host = www.busholini.org) {
           return 301 https://$host$request_uri;
        } # managed by Certbot
        if ($host = git.suspectdevices.com) {
           return 444;
        }
        if ($host = 198.202.31.231) {
           return 444;
        }
    return 404; # managed by Certbot


}

server {
        listen 198.202.31.231:443 ssl; # managed by Certbot
        ssl_certificate /etc/letsencrypt/live/www.busholini.org/fullchain.pem; # managed by Certbot
        ssl_certificate_key /etc/letsencrypt/live/www.busholini.org/privkey.pem; # managed by Certbot
        include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
        ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

        server_name busholini.org w.busholini.org www.busholini.org;

        if ($host = git.suspectdevices.com) {
           return 444;
        }
        if ($host = 198.202.31.231) {
           return 444;
        }

        root /var/www/busholini/www;
        index index.html;

}
cd /etc/nginx/sites-enabled/
ln -s /etc/nginx/sites-available/digithink .
nginx -t
```

INSTALL THE PARTS THAT THE BARTENDER NEEDS

```
apt install python3-flask
apt install python3-gunicorn

apt install at
echo www-data |tee /etc/at.allow
```

2.5 Nick Cave -- My personal datacenter

When I went to new york 2 summers ago I took a mac mini and a pair of 14T disks, which had a sort pile of pictures, my music library and which backed up my laptop. At home it still does the backups. Recently I came apon a 2013 24 core mac pro with 64G of memory and 1T of internal ssd. With that kind of processing I can actually run vms as well as getting my backups served. So we are retiring Costello and building out Nick.

Start by installing debian trixie from a usb stick. This worked flawlessly. Once you have that baseline set up the stuff we use.

```
apt update
# DONT MODERNIZE THE LIST TRIXIE TRASHES IT
apt install sudo nano wget curl
apt install avahi-daemon
apt install avahi-daemon avahi-utils
apt install libnss-mdns
apt install htop
apt install openssh-server
apt install ca-certificates
apt install gpg
apt install sudo
apt install parted
apt install htop
apt install bridge-utils
```

Install the zfs stuff.

```
nano /etc/apt/sources.list
apt update
apt install zfs-dkms
apt install zfsutils-linux
modprobe zfs
zfs list
apt -o Acquire::Check-Valid-Until=false update
apt policy linux-headers-*
apt install linux-headers-$(uname -r)
apt install --reinstall zfsutils-linux
zpool import
zpool import -f tank
zpool import -f local
zpool import -f reddisk
zpool clear tank ; zpool clear reddisk ; zpool status -v
```

replace the bad disk and replace the bad zfs partitions

```
sgdisk --replicate=/dev/sde /dev/sdb
sgdisk -G /dev/sde
ls -ls /dev/disk/by-id|grep sde
zpool status tank
zpool replace tank -o ashift=12 ata-ST12000NM000J-2TY103_ZRT0GC3A-part3 ata-WDC_WD120EFBX-68B0EN0_D7JVX9MN-part3
zpool replace reddisk -o ashift=12 9814339977262587103 ata-WDC_WD120EFBX-68B0EN0_D7JVX9MN-part4
```

Install netatalk

```
wget https://github.com/Netatalk/netatalk/releases/download/netatalk-4-0-0/netatalk_4.0.0.ds-1_amd64.deb
dpkg --install ./netatalk_4.0.0.ds-1_amd64.deb
apt install avahi-daemon
apt --fix-broken install
apt install avahi-daemon avahi-utils
apt install libnss-mdns
systemctl status avahi-daemon
nano /etc/netatalk/afp.conf
systemctl restart netatalk
systemctl status netatalk
```

Add personal keys to ssh authorized keys.

```
nano .ssh/authorized_keys
cp .ssh/authorized_keys ~feurig/.ssh/authorized_keys
chown -R feurig:feurig ~feurig/.ssh
```

attempt to use tbolt 2 enclosure.

```
apt update
apt install thunderbolt-tools

apt install bolt
boltctl
```

```
boltctl list
boltctl enroll
boltctl enroll 00000000-0000-0018-80fa-1c134c238952
boltctl list
fdisk -l
boltctl info 00000000-0000-0018-80fa-1c134c238952
boltctl list
boltctl list --all
```

FAIL

Move on to initial incus fail. Newest incus creates a bridge and expects it to be there.

```
reboot
fdisk -l
nano .ssh/authorized_keys
apt install incus incus-tools qemu-system
curl  -fsSL https://pkgs.zabbly.com/key.asc -o /etc/apt/keyrings/zabbly.asc
sh -c 'cat <<EOF > /etc/apt/sources.list.d/zabbly-incus-stable.sources
Enabled: yes
Types: deb
URIs: https://pkgs.zabbly.com/incus/stable
Suites: $(. /etc/os-release && echo ${VERSION_CODENAME})
Components: main
Architectures: $(dpkg --print-architecture)
Signed-By: /etc/apt/keyrings/zabbly.asc
EOF'
apt update
apt install incus
systemctl start incus
incus admin init
incus --version
```

At this point incus was foobarred and had set up a default lxdbr0 and was not the way we like things. So we....

Delete the lxd bridge and set up the normal bridge.

```
ip link set incusbr0 down
ip link delete incusbr0
cat > /etc/network/interfaces<<EOD
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug enp11s0
iface enp11s0 inet dhcp
auto br0
iface br0 inet static
     address 192.168.129.189
     network 192.168.128.0
     netmask 255.255.128.0
     broadcast 192.168.255.255
     gateway 192.168.128.1
     mtu 9000
     bridge_ports enp12s0
     bridge_stp off       # disable Spanning Tree Protocol
        bridge_waitport 0    # no delay before a port becomes available
        bridge_fd 0          # no forwarding delay
EOD
systemctl restart networking
```

reinstall incus (purge doesnt work correctly here)

```
apt purge incus
apt install incus
incus admin init
incus profile edit default
incus profile show default
incus init ubuntu/focal -c limits.cpu=4 -c limits.memory=8GiB -d root,size=20GiB --vm gru
incus init images:ubuntu/focal -c limits.cpu=4 -c limits.memory=8GiB -d root,size=20GiB --vm gru
incus profile show default
incus profile edit default
incus init images:ubuntu/focal -c limits.cpu=4 -c limits.memory=8GiB -d root,size=20GiB --vm gru
incus start gru
incus init images:ubuntu/focal -c limits.cpu=4 -c limits.memory=8GiB -d root,size=20GiB --vm minion1
incus start minion1
ip link set incusbr0 down
ip link delete incusbr0
nano /etc/resolv.conf
```

So since we have a hot swappable disk container (usb3 since the tbolt is worthless) we wanna mount apfs volumes that we find in our disk collection.

```
apt install apfs-dkms --fix-missing
```

2.6 tk2022 -- Rebuild kb2018 using debian bookworm.

The process for installing debian on the old dl380 is about the same as the dell excep that its bios and not uefi and the disks have to be set up by the controller. (flesh this in a bit)

2.6.1 Prep

For reference see Build notes for guthrie

- back up all containers to /tank
- convert all lxd containers to incus with lxd-to-incus
- migrate all incus containers to temporary server

2.6.2 Rebuild

Partition the disk

```
parted /dev/sdg
GNU Parted 3.6
Using /dev/sdg
...mkpart until you get the stuff below....
(parted) print
Model: HP LOGICAL VOLUME (scsi)
Disk /dev/sdg: 1024GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags: pmbr_boot

Number  Start   End     Size    File system  Name   Flags
 1      1049kB  2097kB  1049kB  fat32                bios_grub
 2      2097kB  250GB   250GB   ext4
 3      250GB   500GB   250GB   ext4         incus

(parted)disk_set pmbr_boot on
(parted)set 1 bios_grub on
(parted)quit
```

Create the root filesystem. You can save yourself some time by copying down the UUID for later.

```
mkfs.ext4 -j /dev/sdg2
mount /dev/sdg2 /mnt/tktest/
```

apt install debootstrap

```
### Debootstrap with proxy.

```sh
export http_proxy=http://192.168.31.2:3128/
debootstrap --arch amd64 bookworm /mnt/debinst http://ftp.us.debian.org/debian
```

Grab a few things from the old server.

```
mkdir /mnt/tktest
mount /dev/sdj /mnt/tktest
incus admin init --dump>/mnt/tktest/root/incusinit.yml
cp -rpv /etc/ssh /mnt/tktest/etc/
cp -rpv /root/.ssh /mnt/tktest/root/
```

Mount chroot environment.

```
mount -t sysfs /proc /mnt/tktest/proc
mount -t sysfs /sys /mnt/tktest/sys
mount --bind /dev /mnt/tktest/dev
mount --bind /dev/pts /mnt/tktest/dev/pts
LANG=C.UTF-8 chroot /mnt/tktest /bin/bash
```

ALTERNATE WAY TO MOUNT
```

```
mkdir /mnt/tktest
mount /dev/sdj /mnt/tktest
mount --make-rslave --rbind /proc /mnt/tktest/proc
mount --make-rslave --rbind /sys /mnt/tktest/sys
mount --make-rslave --rbind /dev /mnt/tktest/dev
mount --make-rslave --rbind /dev/pts /mnt/tktest/dev/pts
LANG=C.UTF-8 chroot /mnt/tktest /bin/bash
PS1='TKTEST\w\$ '
```

Set up apt (with proxy)

```
cat >/etc/apt/sources.list<<EOD
#deb http://ftp.us.debian.org/debian bookworm main
deb http://deb.debian.org/debian bookworm main non-free non-free-firmware contrib
deb http://deb.debian.org/debian bookworm-updates main non-free non-free-firmware contrib
deb http://deb.debian.org/debian-security/ bookworm-security main non-free non-free-firmware contrib
deb http://deb.debian.org/debian bookworm-backports main contrib non-free non-free-firmware
deb [trusted=yes] http://downloads.linux.hpe.com/SDR/downloads/MCP/debian bookworm/current non-free # disabled on upgrade to focal

EOD
TKTEST/# cat > /etc/apt/apt.conf.d/99proxy <<EOD
> Acquire::http::Proxy "http://192.168.31.2:3128/";
> EOD
```

Set up the fstab.

We want to use the uuid for the mounts. The hp raid controller shuffles the /dev/sdx quite a bit.

```
blkid|grep sdb|sed 's/^/# /' >>/etc/fstab
nano /etc/fstab
UUID=c51cb56b-9da4-479b-ba11-dfaac580df64 / ext4 rw,relatime 0 0
UUID=5456b1ce-999f-43a1-b13f-d507321f3ed8 /var/lib/incus ext4 rw,relatime 0 0
# /dev/sdb2: UUID="c51cb56b-9da4-479b-ba11-dfaac580df64" BLOCK_SIZE="4096" TYPE="ext4" PARTUUID="ad01a32f-edc1-4f85-a8e3-b27b2e92fd03"
# /dev/sdb3: UUID="5456b1ce-999f-43a1-b13f-d507321f3ed8" BLOCK_SIZE="4096" TYPE="ext4" PARTLABEL="incus" PARTUUID="998a853e-da55-453a-a936-65d559454ef7"
# /dev/sdb1: PARTUUID="ce1294c5-8fb4-4c82-a3ea-40b6f9872efd"
```

Install stuff you will want installed.

```
apt install openssh-server
apt install ca-certificates
apt install curl
apt install gpg
apt install sudo
apt install parted
apt install htop
apt install git
```

Make devices

```
TKTEST/# apt install makedev
cd /dev
MAKEDEV generic
```

Set up time

There should be a way to preseed the time zone.

```
cat> /etc/adjtime<<EOD
0.0 0 0.0
0
UTC
EOD
dpkg-reconfigure tzdata
```

Set up networking

Make sure you install bridge-utils otherwise the bridges wont come up.

```
apt install bridge-utils
cat >/etc/network/interfaces<<EOD
#------------------------------------------------------------------------/etc/network/interfaces
# 2: enp3s0f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq master br0 state UP group default qlen 1000
# 3: enp3s0f1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq master br1 state UP group default qlen 1000
```

```
# 4: enp4s0f0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
# 5: enp4s0f1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9000 qdisc mq master br3 state UP group default qlen 1000
source /etc/network/interfaces.d/*

auto lo
iface lo inet loopback

iface enp3s0f0 inet manual
iface enp3s0f1 inet manual
iface enp4s0f0 inet manual
iface enp4s0f1 inet manual

auto br0
iface br0 inet manual
     bridge_ports enp3s0f0
     bridge_stp off       # disable Spanning Tree Protocol
        bridge_waitport 0    # no delay before a port becomes available
        bridge_fd 0          # no forwarding delay


auto br1
iface br1 inet static
     address 192.168.31.159
     network 192.168.31.0
     netmask 255.255.255.0
     broadcast 192.168.31.255
     bridge_ports enp4s0f1
     bridge_stp off       # disable Spanning Tree Protocol
        bridge_waitport 0    # no delay before a port becomes available
        bridge_fd 0          # no forwarding delay
EOD
```

Set up resolution.

This is kind of silly since you need to proxy to get anywhere and the proxies do dns. However we do want resolution for the admin land so we add sitka and naomis internal address.

```
cat >/etc/resolv.conf<<EOD
192.168.31.2            # sitka (dnsmasq)
192.168.31.141          # naomi's internal address
search admin.suspectdevices.com merlot.suspectdevices.com suspectdevices.com digithink.com fromhell.com
EOD
```

Install the gigabyte nic drivers.

A linux box without network is secure but useless.

```
apt update
apt install firmware-bnx2
```

Update grub

Since the hp is bios based we install grub-pc rather than an efi based solution.

```
apt install grub-pc
nano /etc/default/grub
GRUB_TERMINAL=console serial
GRUB_GFXPAYLOAD_LINUX=text
GRUB_DEFAULT=0
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_SERIAL_COMMAND="serial --speed=115200"
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="console=tty1 console=ttyS1,115200n8 ipv6.disable=1 iommu=pt"
GRUB_CMDLINE_LINUX="console=tty1 console=ttyS1,115200n8 ipv6.disable=1 iommu=pt"
GRUB_DISABLE_OS_PROBER=false
grub-install /dev/sdj
update-grub2
```

Add update script.

```
nano /usr/local/bin/update.sh
#!/bin/bash
# update.sh for debian/ubuntu/centos/suse  (copyleft) don@suspecdevices.com
echo ---------------------- begin updating `uname -n` ----------------------
if [ -x "$(command -v apt-get)" ]; then
   apt-get update
   apt-get -y dist-upgrade
   apt-get -y autoremove
```

```
fi
if  [ -x "$(command -v yum)" ]; then
   echo yum upgrade.
   yum -y upgrade
fi
if  [ -x "$(command -v zypper)" ]; then
   echo zypper dist-upgrade.
   zypper -y dist-upgrade
fi
echo ========================= done ==============================
^X
chmod +x /usr/local/bin/update.sh
```

Install ssacli

```
apt install gpg
apt install curl
curl -x http://192.168.31.2:3128/ -fsSL https://downloads.linux.hpe.com/SDR/hpPublicKey2048.pub | gpg --dearmor -o /etc/apt/trusted.gpg.d/hpPublicKey2048.gpg
curl -x http://192.168.31.2:3128/ -fsSL https://downloads.linux.hpe.com/SDR/hpePublicKey2048_key1.pub | gpg --dearmor -o /etc/apt/trusted.gpg.d/
hpePublicKey2048_key1.gpg
curl -x http://192.168.31.2:3128/-fsSL https://downloads.linux.hpe.com/SDR/hpPublicKey2048_key1.pub | gpg --dearmor -o /etc/apt/trusted.gpg.d/
hpPublicKey2048_key1.gpg
apt update
apt install ssacli
```

Using ssacli to set the primary boot disk.

```
=> set target ctrl slot=0

   "controller slot=0"

=> show config detail
... find the drive that coresponds to what you want

=> ld 10 modify bootvolume=primary
=>
```

To recover if the selected drive does not boot log into the ilo.

```
</>hpiLO-> power reset

status=0
status_tag=COMMAND COMPLETED
Thu Nov 28 17:04:30 2024

Server resetting .......

</>hpiLO-> vsp
```

Wait for the eternity it takes to run through the hardware and memory on the hp. Once it gets to the actual bios change to the text console.

```
<ESC>(
</>hpiLO-> textcons
```

The text console is nice because (inspite of char set differences) the function keys work. Press f8 when you get to the raid controller (after it searches for the disks)

Text console will not work until it actually gets to the bios and you can switch back to the VSP by escaping out

```
<ESC>(
</>hpiLO-> vsp

Virtual Serial Port Active: COM2
```

Install zfs from trixie

```
apt -t bookworm-backports install zfs-dkms zfs-zed zfsutils-linux
```

Install incus from zabbly (with proxy)

```
apt install curl
curl  -x http://192.168.31.2:3128/ -fsSL https://pkgs.zabbly.com/key.asc -o /etc/apt/keyrings/zabbly.asc
```

```
sh -c 'cat <<EOF > /etc/apt/sources.list.d/zabbly-incus-stable.sources
Enabled: yes
Types: deb
URIs: https://pkgs.zabbly.com/incus/stable
Suites: $(. /etc/os-release && echo ${VERSION_CODENAME})
Components: main
Architectures: $(dpkg --print-architecture)
Signed-By: /etc/apt/keyrings/zabbly.asc

EOF'
apt update
apt install incus
incus admin init

incus storage create devel zfs source=/dev/disk/by-id/wwn-0x600508b1001cfe22c14c918541d42c3a-part1 zfs.pool_name=devel
ls -ls /dev/disk/by-id|grep sda
zpool status devel
zpool attach devel wwn-0x600508b1001cfe22c14c918541d42c3a-part1 wwn-0x600508b1001c2ad6bd48a76e9aee8e03-part1
zpool status infra
zpool attach infra wwn-0x600508b1001cfe22c14c918541d42c3a-part2 wwn-0x600508b1001c2ad6bd48a76e9aee8e03-part2
```

Migrate containers back from spare server.

Again see Build notes for guthrie

Install ansible and set up bitbicket repository

Create an access key on bitbucket with write access to the SusdevAdmin/ansible repo.

Copy the key somewhere safe.

```
git config --global http.proxy http://192.168.31.2:3128
git clone https://x-token-auth:<Token from above>@bitbucket.org/suspectdevicesadmin/ansible.git
ls
cd ansible/
ls
nano ansible.cfg
nano hosts
git config user.email <username provided above>@bots.bitbucket.org
git commit -a -m"test through proxy"
git push
```

2.6.3 References.

- https://www.debian.org/releases/stable/amd64/apds03.en.html
- https://downloads.linux.hpe.com/SDR/downloads/MCP/debian/dists/bookworm/
- https://sleeplessbeastie.eu/2017/06/26/how-to-fix-the-missing-hpes-public-keys/
- https://serverfault.com/questions/1142235/-debian-12-live-grub-installerror-boot-efi-doesnt-look-like-an-efi-partition
- https://linuxopsys.com/mount-partitions-using-uuid-in-linux -https://www.cyberciti.biz/faq/linux-finding-using-uuids-to-update-fstab/

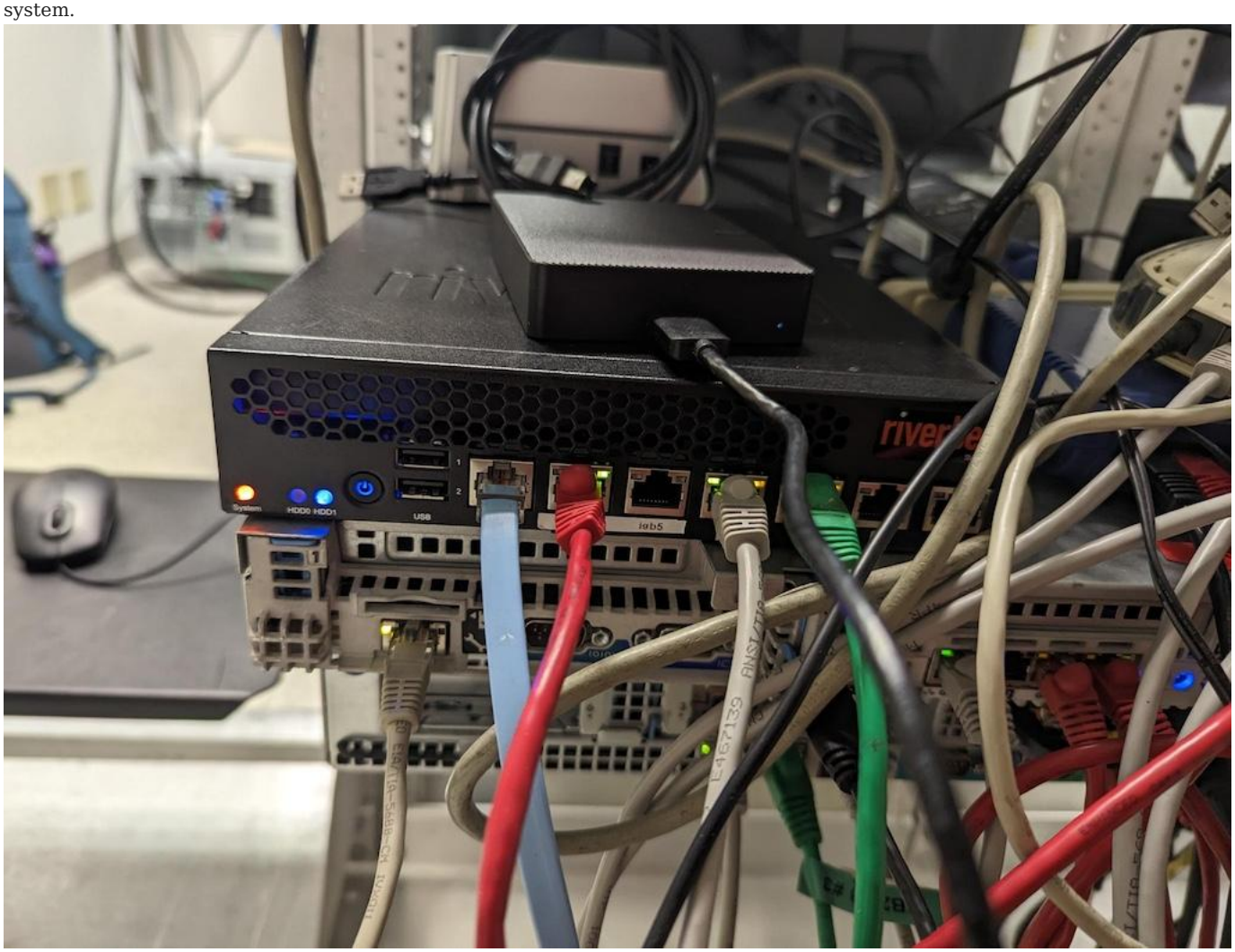
2.7 Sitka -- Using a Tank for Crowd Control

2.7.1 Overview

Before retiring our openwrt router we used a container as a proof of concept We are going to reimpliment it using physical hardware and harden it. The idea is to access the Admin lan without giving it any more access than it needs. The admin land has the servers lights out interfaces (ilo and drac) and allows direct communication between servers. The router will also provide a secondary dns server.

Hardware

Our router was originally designed to be used with pfsense, a comercial product built around freebsd and its packet filtering system.



At home we run opnsense which is an open source replacement. At the colo we are going to strip it down to its underlying operating system and open source compontents.

Components

WIREGUARD

(insert short description of wg) We originally set out to use several complicated vpns until we realized they were overkill.The configuration for wireguard is described in our staging setup

TINYPROXY

The only reason the servers would need to directly connect to anything is to get updates. For this a simple http proxy is all that we need. The configuration for tinyproxy is described in our staging setup.

BIND 9

When the main server is being worked on we completely lose DNS. So we provide the secondary.

DNSMASQ

When talking to isolated internal machines its nice to have local dns. (also a dhcp server for the admin lan)

PF

Pf is bsd's packet filter system.

Redundancy and remote control

2.7.2 Configuration / setup

Initial setup

```
pkg upgrade
pkg install bind918-9.18.30
pkg install dnsmasq
pkg wireguard-tools-1.0.20210914_3
pkg install wireguard wireguard-tools
pkg install tinyproxy
nano /etc/rc.conf
hostname="sitka"
#ifconfig_igb4="DHCP"
ifconfig_igb4="inet 198.202.31.141 netmask 255.255.255.128"
defaultrouter="198.202.31.129"
ifconfig_igb0="inet 192.168.31.2 netmask 255.255.255.0"
sshd_enable="YES"
moused_nondefault_enable="NO"
# Set dumpdev to "AUTO" to enable crash dumps, "NO" to disable
dumpdev="AUTO"
zfs_enable="YES"
ipv6_enable="NO"
ipv6_network_interfaces="none"
ip6addrctl_enable="NO"
dnsmasq_enable="YES"
gateway_enable="YES"
```

Creating a bridge network for the admin lan.

Note: The initial configuration assumed that an external hub would bridge between the admin facing interfaces and the lights out cards on both servers. With the reduction of the colo footprint to a single server this is handled by bridging ibg0 and igb1. The above /etc/rc.conf is changed by replacing ifconfig_igb0 with the following.

```
#ifconfig_igb0="inet 192.168.31.2 netmask 255.255.255.0"
cloned_interfaces="bridge0"
ifconfig_bridge0="inet 192.168.31.2 netmask 255.255.255.0 addm igb0 addm igb1 up"
ifconfig_igb0="up"
ifconfig_igb1="up"
```

Wireguard setup

Wireguard on freebsd is much like wireguard on linux except that instead of iptables the work is done with freebsds packet filter pf.

USE PF TO PASS NETWORK TRAFFIC

```
service wireguard enable
sysrc wireguard_interfaces="wg0"
sysrc gateway_enable=YES
sysctl -w net.inet.ip.forwarding=1
service pf enable
service pflog enable
nano /etc/pf.conf
internal_if="bridge0"
wg_net="10.0.0.0/24"
```

```
scrub in all
nat on $internal_if from $wg_net to any -> ($internal_if)
pass log all
service pf start
service pflog start
```

CONFIGURE WIREGUARD

Wireguard configuration comes in two pieces the local interface and peers that connect to it.

```
nano /usr/local/etc/wireguard/wg0
[interface]
Address = 10.0.0.11/32
ListenPort = 1194
PrivateKey = REDACTED =
#public key Biw53AZ3wWp4mr/iWfuZWi4eFPfFIYjOLT3weE7mFmI=

# note the peers public key will have to come from the client.
[peer]
PublicKey = mxU1WAMJGg3Da5D47rP5OWVYOe4+dwQQum3IFVZHAFY=
AllowedIPs = 10.0.0.16/32
PreSharedKey= REDACTED =
^X
service wireguard start
```

TinyProxy setup

CONFIGURATION

```
nano /usr/local/etc/tinyproxy.conf
User nobody
Group nobody
Port 3128
Listen 192.168.31.2
Timeout 600
Allow 192.168.31.1/24
ViaProxyName "tinyproxy"
DefaultErrorFile "/usr/local/share/tinyproxy/default.html"
StatFile "/usr/local/share/tinyproxy/stats.html"
LogFile "/var/log/tinyproxy.log"
LogLevel Info
PidFile "/var/run/tinyproxy.pid"
MaxClients 50
^X
service tinyproxy enable
service tinyproxy start
```

TEST THE PROXY

Note that there is only the internal interface on this box. The bridge to the outside is anonymous and only the containers have access to it.

```
root@kh2024:~# nano /etc/apt/apt.conf.d/99proxy
Acquire::http::Proxy "http://192.168.31.2:3128/";
^X
root@kh2024:~# apt update
Hit:1 http://deb.debian.org/debian bookworm InRelease
Hit:2 http://deb.debian.org/debian bookworm-updates InRelease
```

Bind 9 / Secondary DNS server

There are several versions of bind avaliable with Freebsd 14.1 but we are using bind9 on the primary so we install bind918.

```
pkg install bind918
```

Everything not in freebsd is off of /usr/local/ so instead of /etc/named the configuration for for bind9 is under /usr/local/etc/namedb the default configuration file (/usr/local/etc/namedb/named.conf) only listens to localhost so the first change to make is to change.

```
    listen-on       { 127.0.0.1; };
```

to

```
    listen-on       { 198.202.31.132; };
```

Then copy the zone directory from the old linux slave server and add the following to the end of the named.conf file.

```
include "/usr/local/etc/namedb/zones/slave.conf";
```

Enable and start the service.

```
sysrc named_enable=YES
service named start
```

2.7.3 Todo

- dnsmasq for internal network.
- hardening.

2.7.4 References

- https://www.digithink.com/rethinkeverything/norouter/wireguard-and-tinyproxy/
- https://forums.freebsd.org/threads/wireguard-network-setup.94793/
- https://forums.freebsd.org/threads/wireguard-setup-with-pf-problems.72623/
- https://vlads.me/post/create-a-wireguard-server-on-freebsd-in-15-minutes/
- https://freebsdsoftware.org/www/tinyproxy.html

Wireguard references

- https://herrbischoff.com/2023/04/freebsd-how-to-set-up-a-simple-and-actually-working-wireguard-server/
- https://forums.freebsd.org/threads/simple-and-secure-vpn-in-freebsd-introducing-wireguard.78628/
- https://www.zenarmor.com/docs/network-security-tutorials/how-to-install-wireguard-on-freebsd

2.8 Utah - Replace Ubuntu with Debian on home server.

Utah (Phillips) is my secondary home file server. It is a cheeze grater style mac with 3 2t ssds on a pci card that are the boot disks and then mirrored 8T and 14T zfs disks. Like the other 2 home servers it is an appletalk server as well as providing lxc containers and running docker processes.

I attempted to repeat the process used at the colo on utah but since there is no ipmi interface on the old mac pro it was not possible.

2.8.1 Initial setup.

I started with a fresh install on one of the 3 nvme disks and then referenced/copied some things around from the old os boot disk and the first attempt to install debian trixie. I won't bother you with the details. I will however reuse /dev/nvme0n1.

```
fdisk -l|grep Disk
mkdir /mnt/ubuntu
mkdir /mnt/wtfdebian
mount /dev/sde2 /mnt/ubuntu/
mount /dev/nvme0n1p2 /mnt/wtfdebian/
```

get rid of the graphical runtime before the system goes to sleep.

```
systemctl set-default multi-user.target
reboot
```

2.8.2 Convert from netplan back to /etc/network/interfaces

Somewhere along the last few ubuntu updates ubuntus netplan started becoming unusable. Sometimes the updates would overwrite the existing configuration without backing it up. When I tried to set up a fresh bridge configuration it refused. So stick a fork in it and turn it over we are done.

```
apt install bridge-utils
cat /mnt/ubuntu/etc/network/interfaces
cat /mnt/ubuntu/etc/netplan/
cat /mnt/ubuntu/etc/netplan/50-cloud-init.yaml
...
nano /etc/network/interfaces
...
#source /etc/network/interfaces.d/*

auto lo
iface lo inet loopback

allow-hotplug enp7s0f1
auto enp7s0f1
iface enp7s0f1 inet dhcp

auto br0
iface br0 inet static
     address 192.168.129.100
     network 192.168.128.0
     netmask 255.255.128.0
     broadcast 192.168.255.255
     gateway 192.168.128.1
     bridge_ports enp7s0f0
     bridge_stp off       # disable Spanning Tree Protocol
        bridge_waitport 0    # no delay before a port becomes available
        bridge_fd 0          # no forwarding delay
^X
systemctl restart networking
```

2.8.3 Install zfs and appletalk

Add repositories from testing(trixie)

The zfs supported by bookworm will not import the filesystems created by ubuntu24.04 but the zfs packages backported from trixie will.

```
nano /etc/apt/sources.list
deb-src http://security.debian.org/debian-security bookworm-security main non-free-firmware
```

```
deb-src http://debian.osuosl.org/debian/ bookworm-updates main non-free-firmware
deb-src http://debian.osuosl.org/debian/ bookworm main non-free-firmware
deb http://security.debian.org/debian-security bookworm-security main non-free-firmware
deb http://debian.osuosl.org/debian/ bookworm-updates main non-free-firmware
deb http://debian.osuosl.org/debian/ bookworm main non-free-firmware
deb http://deb.debian.org/debian-security/ bookworm-security main non-free non-free-firmware contrib
deb http://deb.debian.org/debian bookworm-updates main non-free non-free-firmware contrib
deb http://deb.debian.org/debian bookworm main non-free non-free-firmware contrib
deb http://deb.debian.org/debian bookworm-backports main contrib non-free non-free-firmware
^X
```

Install Netatalk 4 debian provided by netatalk

There was no debian maintainer for a year or so which meant that debian dropped appletalk support on bookworm. This has been resolved and should be supported when trixie is released. In the mean time the fine folk at netatalk have provided a packaged version for both netatalk3 and netatalk4 for bookworm.

```
wget https://github.com/Netatalk/netatalk/releases/download/netatalk-4-0-0/netatalk_4.0.0.ds-1_amd64.deb
dpkg --install ./netatalk_4.0.0.ds-1_amd64.deb
apt --fix-broken install
dpkg --install ./netatalk_4.0.0.ds-1_amd64.deb
apt install avahi-daemon
```

Install zfs from bookworm-backports and import existing pools.

```
apt -t bookworm-backports install zfs-dkms zfs-zed zfsutils-linux
zpool import
zpool import -f tank
zpool import -f reddisk
```

Configure appletalk to serve the zfs pools

```
nano /etc/netatalk/afp.conf
...
[Global]
; Global server settings
mimic model = RackMac
afp listen = 192.168.129.100

; pretty sure this one stays.
map acls = mode
; Not sure about the next two
;aclinherit = passthrough
;aclmode = passthrough

[Homes]
basedir regex = /home

[reddisk]
path = /reddisk
^X
systemctl enable avahi-daemon
systemctl start avahi-daemon
systemctl status avahi-daemon
systemctl enable netatalk
systemctl start netatalk
systemctl status netatalk
```

2.8.4 Install and initialize incus

Make some space

Unlike lxd incus does most of its work under /var/lib/incus as apposed to in the storage pools. You need to make some room there.

```
cat /etc/fstab
# / was on /dev/sdf2 during installation
UUID=1403cbd4-a187-4cd2-8d83-c7c036b3e589 /               ext4    errors=remount-ro 0       1
# /boot/efi was on /dev/sdf1 during installation
UUID=0E04-F115  /boot/efi       vfat    umask=0077      0       1
# /home was on /dev/sdf4 during installation
UUID=22bbd39c-e3b8-402a-bcc2-cd05a889cecb /var/lib/incus           ext4    defaults        0       2
# swap was on /dev/sdf3 during installation
UUID=4f2aa57f-219d-4355-815c-3a89a613a8cb none            swap    sw              0       0
```

Install from bookworm-backports (trixie)

```
apt update
apt -t bookworm-backports install incus incus-tools qemu-system

root@utah:~# ls -ls /dev/disk/by-id|grep nvme0
0 lrwxrwxrwx 1 root root 13 Nov 17 16:31 nvme-eui.0000000001000000e4d25c757b395601 -> ../../nvme0n1
0 lrwxrwxrwx 1 root root 13 Nov 17 16:31 nvme-INTEL_SSDPEKNU020TZ_PHKA314002UT2P0C -> ../../nvme0n1
0 lrwxrwxrwx 1 root root 13 Nov 17 16:31 nvme-INTEL_SSDPEKNU020TZ_PHKA314002UT2P0C_1 -> ../../nvme0n1
root@utah:~# ls -ls /dev/disk/by-id/nvme-INTEL_SSDPEKNU020TZ_PHKA314002UT2P0C
0 lrwxrwxrwx 1 root root 13 Nov 17 16:31 /dev/disk/by-id/nvme-INTEL_SSDPEKNU020TZ_PHKA314002UT2P0C -> ../../nvme0n1
root@utah:~# incus admin init
Would you like to use clustering? (yes/no) [default=no]:
Do you want to configure a new storage pool? (yes/no) [default=yes]:
Name of the new storage pool [default=default]: local
Name of the storage backend to use (dir, zfs) [default=zfs]:
Create a new ZFS pool? (yes/no) [default=yes]:
Would you like to use an existing empty block device (e.g. a disk or partition)? (yes/no) [default=no]: yes
Path to the existing block device: /dev/disk/by-id/nvme-INTEL_SSDPEKNU020TZ_PHKA314002UT2P0C
Would you like to create a new local network bridge? (yes/no) [default=yes]: no
Would you like to use an existing bridge or host interface? (yes/no) [default=no]: yes
Name of the existing bridge or host interface: br0
Would you like the server to be available over the network? (yes/no) [default=no]: yes
Address to bind to (not including port) [default=all]:
Port to bind to [default=8443]:
Would you like stale cached images to be updated automatically? (yes/no) [default=yes]:
Would you like a YAML "init" preseed to be printed? (yes/no) [default=no]: yes
config:
  core.https_address: '[::]:8443'
networks: []
storage_pools:
- config:
    source: /dev/disk/by-id/nvme-INTEL_SSDPEKNU020TZ_PHKA314002UT2P0C
  description: ""
  name: local
  driver: zfs
profiles:
- config: {}
  description: ""
  devices:
    eth0:
      name: eth0
      nictype: bridged
      parent: br0
      type: nic
    root:
      path: /
      pool: local
      type: disk
  name: default
projects: []
cluster: null
```

3. Resume

3.1 D Delmar Davis

Portland, Oregon, ddelmardavis@gmail.com (503) 284-2945

3.1.1 Summary

(I'll be 78 when Unix(tm) time ends)....

I have been administering Unix systems for more than 30 years, in addition to deploying and maintaining Web and other Internet services for 25. The systems have ranged from stand-alone, completely exposed servers to services with separate Web, application, and database layers, clustered and load balanced for redundancy and scalability, and placed behind firewalls, I have done work in the ever amorphous cloud infrastructure provided by Amazon but I am more interested in locally owned and operated LXD based containers.

I have extensive experience with Linux deployments (Rocky/RH/Fedora/Centos, SUSE, Debian/Ubuntu, Tizen, OpenWRT) I have professional experience with HP-UX, True64, Linux, Solaris, Freebsd, OS X, and AIX and I have installed and configured databases such as Oracle, Sybase, and SQL Server. Against my better judgment, I have installed, maintained and configured Microsoft systems, and even made them play well with others (SSO, SMB, etc). Additionally, I have a strong background and proven success in providing instruction and documentation for work to be maintained by others.

Throughout my career I have learned new systems quickly. I follow issues and problems through to solution, be they social or technical. Security is the key to surviving on both the World Wide Web and large Intranets. For this reason, I am a practitioner and proponent and of well defined policies, best practices, regular updates, and common sense. My reputation for getting things done and team participation make me an ideal candidate for positions where honest work is valued. Specialties: Thinking outside the box.

Also: Springfield and Waltham are not Boston in the same way Hillsboro and Beaverton are not Portand and I am not interested in your urgent requirement for Intel's permanently contingent workforce at Jones Farm so don't bother asking....

3.1.2 Experience

Systems Engineer

Laika Jul 2021 - Present.

Contiguous Online Server Presence

Fromhell.com Nov 1996 - Present (28 years 2 months +)

I have built and maintained servers for the domains digithink.com and fromhell.com (email only) since 1996. They started as a sparc IPC running SunOs 4.1.3 and have had many different varients of bsd and linux since then. They are currently on incus containers running Debian. My current toolset includes ansible, python and zfs. I also document my work even when it's just mine. (See:https://www.digithink.com)

Package Handler

UPS Oct 2017 - Jul 2021 (3 years 9 months)

Its funny, If I went to the gym I would never build the muscles I have from sorting ~6000 packages a day for close to 4 years. I had a 12 minute bicycle commute. Another 10 minutes to get to the other side of the channel to PCCs campus in the shipyards (Before covid, UPS reimbursed my tuition as I updated my welding skills). I got to see first hand the advantages that unionized workers have over similar non union positions (looking at you Amazon). Through the teamsters I recieved better health insurance than I ever received doing tech work. Also nice to not have a digital leash (going on 3 years cell phone free). Life is good.

Applications Engineer

Suspect Devices Apr 2008 - Jan 2017 (8 years 10 months)

For roughly a decade I worked through Tempus Dictum (DBA Suspect Devices) building hardware and software solutions around the Arduino and other open source hardware and software. My primary focus was on tools for artists and musicians. Client projects ranged from medical equipment, to bicycle racing, to music, to heavy industrial motor controllers. In addition to my embedded work I coded applications for Macintosh and iPhone. I developed hardware platform for teaching micro-controllers to artists and hobbyists. I produced and continue to present workshops focused on introducing micro-controllers to the community. As a systems administrator I maintained multiple internet servers for this company and clients. Upgraded and maintaining a linux based scientific computing cluster at UCSF. Test-deployed cluster implementation to the Amazon Elastic Cloud to benchmark cloud against old hardware. Recent client work involved moving a legacy FreeBSD System to the cloud and exploring cloud based archival options.

IT Administration Engineer

Jaguar Land Rover Sep 2013 - Aug 2015 (2 years)

I was contracted to provide primary IT support for the servers and workstations at the JLRNA's Open Software Technology Center in Portland. Supported the build servers and developers tools for the tizen platform as prescribed by Intel (OpenSuse 12.1-12.3). Later I was hired. I migrated servers to new Open Source Technology Center in the Pearl and argued successfully to move to Debian as primary linux platform. I was unsuccessful in convincing management that the position should be outsourced.. Supported 56 servers running Debian and OpenSUSE in addition to roughly the same number of Windows 7 based systems and users.

Continuing Education Instructor

Pacific Northwest College of Art Sep 2012 - Dec 2014 (2 years 4 months)

After three years of giving one day classes to the community through Dorkbotpdx I was asked by PNCA's extensions school to develop curriculum for and teach an 8 week survey course introducing Artists to microcontrollers using the Arduino platform.

Unix Systems Engineer

Adecco Dec 2012 - Mar 2013 (4 months)

I fulfilled a 3 month contract with Integra focused on covering staff shortfalls (2 people covering ~170 business critical systems). This work included a security audit of key systems, ongoing Oracle upgrade support, as well as replacing much of the expensive and complicated CA Spectrum suit with Nagios and other open source monitoring tools. Worked with senior administrator to streamline and clean up administration. Helped to evaluate free version of puppet versus CF engine and hand rolled scripting. And in general kept the rubber side down.

Senior Network Analyst

Washington County Mar 2007 - Nov 2007 (9 months)

Built and configured systems for Oracle RAC cluster using SLES on generic SAN connected blades to replace expensive L Class HP Servers. Moved all production data from HP-UX file server to Novell OES server on SAN connected blade. (Reducing Costs / Increasing Performance).

Unix System Administrator

Washington County May 2006 - Dec 2006 (8 months)

Tested and documented file restoration on business critical systems as well as creating process and media for bare bones AIX recovery. Developed transition plan from aging HP-UX servers to san attached generic blade architecture. Deployed first workstation in this process. Worked with IT Services (ITS) and one of its vendors to resolve several support issues. Interviewed, helped select, and trained permanent Unix System Administrator. Created operations and troubleshooting guides for all ITS Unix systems. Worked toward better integration between Unix SA position and ITS support team. Upgraded mediawiki server and

trained ITS staff to use it for its internal documentation. Worked with application and ITS teams to provide additional san space, file restoration, additional printers and modems as needed.

Unix Consultant

SolutionsIQ May 2005 - May 2006 (1 year 1 month)

Prepared Unix production and development systems for maintenance by windows based skeleton crew. Systems were comprised of Linux and Solaris based Oracle servers along with several Debian- based infrastructure servers. Cloned Solaris 8/Oracle 8 server for disaster recovery. Spec'd out and installed additional disk for all Solaris Oracle servers. Built serial console server for sun systems. Trained operations staff in general maintenance tasks for all Unix and Linux platforms.

Unix System Administrator

SolutionsIQ Sep 2005 - Dec 2005 (4 months)

Researched, redesigned and deployed web server and content management framework for internal content and documentation. Integrated existing documentation into framework. Trained team to perform ongoing maintenance. Deployed 11 production and development servers running Solaris 9 and 10. Worked with other engineers to refine Jumpstart install process, in particular with regards to JASS and postinstall scripts.

Network Engineer

Hewlett Packard Enterprise Sep 2002 - May 2005 (2 years 9 months)

Maintained all aspects of 150 system proprietary development environment for HP's internal web development at remote data center. Managed migration of initial environment to more stable data center. Coordinated the addition 70 servers to meet expanded capacity needs. Worked with other administrators to keep all systems patched in response to constant security updates (over 400 systems, HP-UX/Linux/W2K). Built 4 Terabyte HPUX /Samba solution for Windows cluster suffering unplanned exponential storage growth. Built and maintained HP-UX build server for Linux distributions. Helped develop organizational security policy. Provided 24x7 tier-1 support for live applications as member of 2 teams spanning 5 data centers. Worked with development teams to prepare troubleshooting guides clear enough to outsource support. Initiated password audit to bring production systems in line with security policy. Maintained professional level of service and support to organization which suffered 7 re- organizations over a two and a half year period.

UNIX System Administrator

Rogue Wave Sep 2001 - Apr 2002 (8 months)

Worked as a member of 5-person team maintaining over 90 UNIX servers used to develop, build, and troubleshoot Rogue Wave's C++ library products. Installed and configured 30 systems running Solaris, AIX, HP-UX, True64 and Linux. Primarily responsible for new Solaris and AIX build servers. Installed and configured Oracle and Sybase databases. Worked with Senior Admin to develop centralized, scripted system setup to allow rapid deployment/ recovery of system configurations for a given software release. Implemented / maintained configuration scripts for Solaris (2.6 - 9beta) and AIX (4.33 - 5.1) systems. Migrated production NIS and license servers from arcane and dying systems to supportable hardware and OS levels. Set up demo server for new web-based technology. Installed compilers, databases, patches, and other software required for development.

Fabricator

GIBSON STEEL FABRICATING, INC. Jun 2001 - Sep 2001 (4 months)

I told my employer that if they called me at 11am on my days off, as they had for months, I was walking. You have to mean that stuff. To be fair it took well over a month before I got the call, gave my notice, and went back to welding. I stood and welded out catch basins (GMAW and OXY/Acetylene work) until I realized how short I was financially. I asked for $1.75 more an hour at a shop that hadn't raised any of its employees wages in close to a year. They raised the entire shop floor by 50c and offered me a 75c raise. I took the first tech job offered. Like I said. You have to mean it.

UNIX System Administrator / Database Programmer

Modern Medium Aug 2000 - Jul 2001 (1 year)

Systems/Database Administrator and Programmer for www.buymusichere.com, a stocked 250,000 product virtual storefronts for 15 clients with 100 projected. Previous contractor was unable to produce more than 4 working storefronts in 1 year; in 3 months we produced 15 storefronts. Eliminated redundant data, reducing processing time and storage needs by ~70%. Established backup of databases and system. Reconfigured raid system to use existing resources effectively. Installed additional memory, disk and processor to allow for growth. Set up development environment on smaller Sun. Wrote import scripts in Transact SQL/PHP for the automatic updating of databases.

President

Digithink Jun 1996 - Dec 2000 (4 years 7 months)

Set up a small ISP while attending college. Consulted on various jobs, resolved Sun hardware/DNS issues for Ordata.com (now Willamette.net). Net presence provided the basis for contract work and fiscal stability between major contracts/

UNIX System Administrator / Programmer

Northwest Media May 2000 - Aug 2000 (4 months)

Created development environment for web site geared towards post care tracking of youth from programs such as foster care, jobs plus, and job core. Set up UNIX (FreeBSD) based development environment using Staging / File server behind a firewall. Ported web-server based data entry to user-friendly firewall protected Access/VBA application, wrote DLL's to publish data to server.

UNIX Systems Specialist/Database Programmer

Oregon Public Education Network Feb 1997 - Jun 1999 (2 years 5 months)

Provided system administration, programming, and technical consulting to the OPEN-C website http:// www.open.k12.or.us. Deployed three web servers with systems running Solaris and HP-UX (everything from boxes of parts to web sites). Developed SQL/WWW scripting language in Perl, consolidating most of sites cgi-scripts. Later contracted to redesign system as an Apache Module. Provided technical consulting, diagnosing and resolving all UNIX Network and World Wide Web related issues.

System Administrator and Security Consultant

DNSI Sep 1996 - Jul 1997 (11 months) Instituted security audit of systems and made recommendations to improve operational security. Proposed and implemented plan to restructure LLC company with crippling financial debt. This incorporated upgrading Internet connectivity while reducing costs, and restructuring company to provide financial solvency and stability. Served as liaison with US West, guiding company through complex series of modem line shortages. Arranged transfer of hardware and software for 700 clients to new servers and location. Daily operations included setup and administration of FreeBSD and Linux servers from building/ installing hardware, OS, services, and virtual hosts all the way through client relations.

SHOP HAND/FABRICATOR

GIBSON STEEL FABRICATING, INC. Jul 1995 - Sep 1996 (1 year 3 months)

When I moved to Eugene the wage base was so low that Symatec moved its customer support there. So I went to work using the skills I learned in High School (Welding). I performed all aspects of storm water catch basin assembly except for seam welding. Operated hydraulic sheers, torch ,band saw, and fork lift. Assembled basins using SMAW.

Technician

Eli Hefron and Sons Jan 1993 - May 1993 (5 months)

Tested setup and configured surplus sun systems, including installation of SunOS 4.1.x through Solaris 1.x [sic] for shipment to clients.

UNIX System Administrator/CAD Support Specialist

Badger Engineers Feb 1989 - Sep 1992 (2 years 7 months)

Supported department's expansion from one Vax and 4 un-networked PCs to two Vax's, 40 Unix Stations, and 100 networked PCs. UNIX administration included direct user support, adding software, user accounts and scripting. Served as project leader; responsible for development of a method of high volume batch translation between different CAD formats. Piloted the use of several (then) new technologies for better cross platform integration such as PCNFS and sendmail based problem logging. Initiated cadre based help/support groups. Trained senior operators and engineers in computer fundamentals.

Engineering Programmer

Bovay Northwest, Inc. 1986 - 1988 (3 years) Automated drafting and design processes; programming in AutoLisp, MuLisp, and DBase.

3.1.3 Education

University of Oregon

Bachelor of Arts, History 1995 - 2004

At UO I studied History with an emphasis on revolution. While there I actively promoted punk and other local music as a DJ at KWVA radio. I was invited to display my artwork in several solo and group exhibitions.

Portland Community College

Carreer Pathways Certificate, Welding Technology/Welder 2017 - 2017

Spokane Falls Community College

Associate of Arts, Science; Software Engineering Technology 1983 - 1991 At SFCC I split my studies between fine arts, core studies and software engineering.

Kellogg Sr High

High School Diploma, High School 1982 - 1983

Cross country, debate, yawn.

Secondary Diplomas and Certificates

ITP Summer Camp 2013

3.1.4 Licenses & Certifications

Linux Foundation Certified Systems Administrator (LFCS-1500-0198-0100)

The Linux Foundation Issued Feb 2015 - Expired Feb 2017

Osha 10

US Department of Labor Issued Jul 2017

3.1.5 Skills

Unix • Linux • System Administration • Disaster Recovery • Computing • Troubleshooting • Data Center • Firewalls • Security • Cloud • GTAW • SMAW • FCAW • GMAW

3.2 Abovethenorm

Name: Delmar Davis

Role: Systems Engineer

“Delmar takes on big, complicated projects that require developing new skills along the way, and he always delivers. This is a testament to his decades of experience in this field! An example of this pattern is his commitment to deploying the VMware environment in order to retire the legacy KVM hosts. Delmar dove in headfirst and was able to wrap up the entire project by the end of the year. That gave us the ability to start 2025 on the right foot in terms of environment and virtualization platform. Our ability to work using those virtual machines is invisible to us due to the large effort that Delmar did behind the scenes.

Delmar always puts himself forward to take on these big projects rather than being nominated to do them, and this is due to his can-do attitude and dedication to pushing things forward. I absolutely love seeing Delmar in action, and I know that the team and I can blindly rely on his dedication, expertise, and intelligence to get the job done. Delmar often posts links to songs in the Core Mattermost channel, often by old punk bands none of us have ever heard of, when the song matches the sentiment of what we're discussing. We find it funny yet engaging in a very strange way that gets us all connected on a different level. Delmar is just truly genuine and amazing.”

4. Rethinkeverything

4.1 Things that need to be rethought

- canonical
- ansible
- git mirroring
- remote backups
- vpn/tailscale
- file shares
- ticketing

4.1.1 Linkdump

- https://www.openproject.org/
- https://www.howtogeek.com/devops/how-to-set-up-a-personal-gitlab-server/
- https://duplicity.gitlab.io
- https://rdiff-backup.net
- https://github.com/librsync/librsync?tab=readme-ov-file
- https://librsync.github.io
- https://www.maketecheasier.com/rclone-sync-multiple-cloud-storage-providers-linux/

4.2 TwentyTwentyTwo

4.2.1 There is no bullet list like MY Bullet list (2022 version)

Notes to myself #rethinkeverything

Switch hands

- Move the pain
- Rewire the brain

IT'S YOUR DATA

- Hand Copy it in Triplicate.
- If its social then scrape it and automate it. God knows they do.

IT'S YOUR WORK

- They can't own what you learn.
- Redact and copy your notes in Triplicate.
- Create/test and share open source gists/solutions
- Make work pathways to give back to the community
- If you have to learn it you best use it at home (lxd5/ansible/jellyfish/usw)

WORK ON ONE LESS THING (SIMPLIFY)

- Every convenience is a point of failure or an attack surface.
- Git does not need to look good to be usefull. (--gitea, --gitlab, ++bare-git+hooks/mirrors)
- -- Twitter
- with or without musk
- also #fuckthatguy.
- If your content is usefull it will recieve the appropriate tweets, links, usw.
- and if it doesnt the internet is fundamentally broken.
- If your wysywig is so unusable you don't "blog" Throw it away. (--wordpress)

HOME IS WHERE THE HEART IS

- Don't let pi/routers do server/container work.
- PiHole (filtering dns)
- dhcp
- look at virtualized routing
- If you can't netboot off of it is it really your network.
- Same goes for centralized management (ldap).
- dhcp
- tftp
- iscsi
- All active work behind at least one firewall.
- Automate pushes (including this site).
- Streamline/cleanup html generation
- It should also be replicated in at least one other location

LET'S GO TO YOUR PLACE

- Ticketing systems should be more like distributed ~~punch~~notecards. --trac
- You shouldnt be giving out XXX bucks a month to post your public images so they can be "distributed"
- flicker free
- multi homed

JUST BECAUSE YOU HAVE SOURCE CONTROL DOESNT MEAN IT'S ALL CODE

- use markdown/git for most things.
- but focus on the english.

WORK IMITATES LIFE

(What problems are we trying to solve)

- Minimize technical debt both past and future.
- Disentangle the various interconnected pieces and dependencies
- Automate as much as possible
- Document what is to be done. (Specification and sample implimentation)
- Practice experience based stepwise refinement.

START MAKING SENSE.

Consolidate home network using OPNSense.

- REMOVE openwrt router
- REMOVE dedicated caching server
- REMOVE dedicated dnsmasq server.
- REMOVE (that fucking qwest router)
- KEEP Pihole-FTL dns based blacklisting
- ADD Better firewall rules
- ADD VPN acces to home network
- ADD Isolated Wireless network for solar array controller.

Sarcasms (link them later)

1. See: Tufte's critiq of power point.
2. "as code" is only as good as managements understanding of the job of the people who actually write and maintain it minus corporate whims, the abuse of executive privilege, and cultural constraints..

4.2.2 Wordpress in 2024

After 3+ years of trying, I still havn't managed to export my wordpress site into something static and maria free and I seem to need to keep using it even though the ui has become impossible to blog in. So I am readopting wordpress or at least trying to get it running on this years ubuntu and not hating it too much. Also upgrading the existing server from focal to jammy screws this site into something approaching disfunction usually only found at work work without the coresponding paycheck.

OS ubuntu using apt install wordpress. WOOOT! We can install wordpress as a package!!!

```
apt update
apt install wordpress
apt install certbot python3-certbot-apache
cd /etc/apache2/sites-available/
```

Unfortunately its not that simple. The example from ubuntu sucks ass in terms of detail and the details are on the sites that hand roll the software.

```
apt update
apt install wordpress
mysql
mysql -u root
apt install mysql-server
mysql -u root
apachectl start
cd /etc/wordpress/
ls
nano config-localhost.php
mysql -u www-data
mysql -u root
ls
nano /etc/apache2/sites-enabled/000-default.conf
apache2ctl restart
nano /etc/apache2/sites-enabled/000-default.conf
apache2ctl restart
ls /etc/apache2/mods-available/
a2enmod socach*
a2enmod ssl
systemctl restart apache2
ln -s config-localhost.php config-suspectdevices.com.php
apache2ctl restart
nano config-localhost.php
reboot
ip a
cd
ls .ssh/
ls .ssh/authorized_keys
cat .ssh/authorized_keys
cat ~joe/.ssh/authorized_keys >>.ssh/authorized_keys
cat ~feurig/.ssh/authorized_keys >>.ssh/authorized_keys
cat .ssh/authorized_keys
exit
cat /etc/apache2/sites-enabled/000-default.conf
chown -R www-data:www-data /usr/share/wordpress
nano /etc/wordpress/htaccess
nano /etc/wordpress/config-localhost.php
nano /usr/share/wordpress/wp-config.php
reboot
find / -name php.ini -print 2>/dev/null
nano /etc/php/8.1/apache2/php.ini
nano /etc/php/8.1/cli/php.ini
reboot
history
history|cut -c8-200
```

4.2.3 the mother obscening issues (based on 2023s attempt)

1. Why the obscenity do you move the location of the software when you package it.
2. Why the obscenity don't you install mysql-server
3. Why the obscenity does your default install not deal with the fact that NO ONE USES FTP FOR ANYTHING.
4. WHO THE OBSCENITY BOTHERS TO EXPORT AND IMPORT A SITE USING LESS THAN 2M of data.
5. Man the result of exporting a site and importing it creates some goddamned ugly site.

References

- https://www.hostinger.com/tutorials/fix-the-uploaded-file-exceeds-the-upload-max-filesize-directive-in-php-ini-wordpress
- https://stackoverflow.com/questions/37157264/wordpress-plugin-install-could-not-create-directory
- THIS https://www.linode.com/docs/guides/how-to-install-wordpress-ubuntu-22-04/ -https://ubuntu.com/server/docs/how-to-install-and-configure-wordpress

THIS

(Optional) WordPress typically uses FTP credentials to install new themes and plug-ins. Add the following lines to the wp_config.php file to remove this restriction. This file is located in the root directory for the domain inside the public_html subdirectory.

File: /var/www/html/example.com/public_html/wp-config.php

```
/** Bypass FTP */
define('FS_METHOD', 'direct');
```

4.2.4 Costello, an Ubuntu 22.04 lxd 5 home server.

Costello is a portable server. I set it up to take to nyu for backing up my laptop, serving music and I wanted originally to use it to sort through my pics/data.

YOU ARE HERE. This needs to be updated for its post debian installed state.

```
# snap install lxd
# apt install htop  openssh-server install netatalk  zfsutils-linux
# apt remove --purge network-manager network-manager-gnome network-manager-pptp network-manager-pptp-gnome
```

```
# ip a |sed 's/^/# /'>> /etc/netplan/01-network-manager-all.yaml
# nano /etc/netplan/01-network-manager-all.yaml
#----------------------------------- /etc/netplan/01-network-manager-all.yaml
#
# Dont Let NetworkManager manage *ANY* devices on this system
#
# enp3s0f0  68:fe:f7:09:3c:4c
# Dont Let NetworkManager manage *ANY* devices on this system
#2: enp3s0f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
#    link/ether 68:fe:f7:09:3c:4c brd ff:ff:ff:ff:ff:ff
#    inet 192.168.128.229/17 brd 192.168.255.255 scope global dynamic noprefixroute enp3s0f0
#3: wlp2s0: <NO-CARRIER,BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state DORMANT group default qlen 1000
#    link/ether 18:81:0e:ee:7c:88 brd ff:ff:ff:ff:ff:ff

network:
  version: 2
  renderer: networkd
  ethernets:
    eth0:
        match:
          macaddress:  68:fe:f7:09:3c:4c
        mtu: 7000
        dhcp4: no
        dhcp6: no
        set-name: eth0
    wlp2s0:
        dhcp4: no
        dhcp6: no
  bridges:
    br0:
        dhcp4: no
        dhcp6: no
        mtu: 7000
        addresses:
            - 192.168.129.45/17
        #gateway4: 192.168.129.1
        routes:
          - to: default
            via: 192.168.129.1
        nameservers:
            addresses:
                - 192.168.129.1
                - 198.202.31.132
        interfaces:
            - eth0
# netplan apply
# reboot
```

```
fdisk -l
# fdisk -l
...
Disk /dev/nvme0n1: 1.82 TiB, 2000398934016 bytes, 3907029168 sectors
Disk model: CT2000P2SSD8
...
Device           Start        End    Sectors  Size Type
/dev/nvme0n1p1    2048    1050623    1048576  512M EFI System
/dev/nvme0n1p2 1050624 3907028991 3905978368  1.8T Linux filesystem
...
Disk /dev/sda: 12.73 TiB, 14000519643136 bytes, 27344764928 sectors
Disk model: M001G-2KJ103
...
Device           Start         End    Sectors  Size Type
/dev/sda1         2048  6442452991 6442450944    3T Linux filesystem
/dev/sda2   6442452992 12884903935 6442450944    3T Linux filesystem
/dev/sda3  12884903936 21474838527 8589934592    4T Linux filesystem
/dev/sda4  21474838528 27344764894 5869926367  2.7T Linux filesystem
...
Disk /dev/sdb: 1.86 TiB, 2048408248320 bytes, 4000797360 sectors
Disk model: JAJS600M2TB
...
Device      Start        End    Sectors  Size Type
/dev/sdb1      40     409639     409600  200M EFI System
/dev/sdb2  409640 4000797319 4000387680  1.9T Apple APFS
```

```
# ls -lsa /dev/disk/by-id/|grep sda
# zpool create tank wwn-0x5000c500dc29d6c5-part4
```

```
lxd init
# lxd init
Would you like to use LXD clustering? (yes/no) [default=no]: yes
What IP address or DNS name should be used to reach this node? [default=192.168.129.45]:
Are you joining an existing cluster? (yes/no) [default=no]:
What name should be used to identify this node in the cluster? [default=costello]:
Setup password authentication on the cluster? (yes/no) [default=no]: yes
Trust password for new clients:
Again:
Do you want to configure a new local storage pool? (yes/no) [default=yes]:
Name of the storage backend to use (btrfs, dir, lvm, zfs) [default=zfs]:
Create a new ZFS pool? (yes/no) [default=yes]:
Would you like to use an existing empty block device (e.g. a disk or partition)? (yes/no) [default=no]: yes
Path to the existing block device: /dev/disk/by-id/wwn-0x5000c500dc29d6c5-part1
Do you want to configure a new remote storage pool? (yes/no) [default=no]:
Would you like to connect to a MAAS server? (yes/no) [default=no]:
Would you like to configure LXD to use an existing bridge or host interface? (yes/no) [default=no]: yes
Name of the existing bridge or host interface: br0
Would you like stale cached images to be updated automatically? (yes/no) [default=yes]:
Would you like a YAML "lxd init" preseed to be printed? (yes/no) [default=no]: yes
config:
  core.https_address: 192.168.129.45:8443
  core.trust_password: ON.TACOCAT.NO
networks: []
storage_pools:
- config:
    source: /dev/disk/by-id/wwn-0x5000c500dc29d6c5-part1
  description: ""
  name: local
  driver: zfs
profiles:
- config: {}
  description: ""
  devices:
    eth0:
      name: eth0
      nictype: bridged
      parent: br0
      type: nic
    root:
      path: /
      pool: local
      type: disk
  name: default
projects: []
cluster:
  server_name: costello
  enabled: true
  member_config: []
  cluster_address: ""
  cluster_certificate: ""
  server_address: ""
  cluster_password: ""
  cluster_certificate_path: ""
  cluster_token: ""
```

```
nano /etc/systemd/resolved.conf
[Resolve]
DNS=192.168.129.250
#FallbackDNS=
Domains=lan suspetdevices.com local
ln -sf /run/systemd/resolve/resolv.conf /etc/resolv.conf
nano /etc/nsswitch.conf
...
hosts:          files mdns4_minimal dns [NOTFOUND=return] dns
...
```

```
# nano /etc/netatalk/afp.conf
;------------------------------------------- /etc/netatalk/afp.conf
; Netatalk 3.x configuration file
;

[Global]
; Global server settings

; pretty sure this one stays.
map acls = mode
; Not sure about the next two
aclinherit = passthrough
aclmode = passthrough

[tank]
path = /tank
ea=none
# service netatalk restart
```

```
# chown feurig /tank/
# su - feurig
```

DOCUMENTATION.

```
ip3 install mkdocs                                  # Install mkdocs
pip3 install mkdocs-bootswatch                       # Themes
pip3 install mkdocs-multirepo-plugin                 # Multi-repo support
pip3 install mkdocs-mermaid2-plugin                  # Mermaid.js support
pip3 install autolink-references-mkdocs-plugin       # Autolink tickets inserted into docs
#mkdocs serve
```

4.3 Ansible

4.3.1 Anisible vs Puppet vs pylxd vs incus.

THIS IS VERY MUCH A WORK IN PROGRESS

Moving to incus pretty much broke the cloud.init+ansible pattern I developed when I started working with lxd. On the otherhand these patterns are pretty disfunctional (unusable by others, tempermental, usw)

Get to the point.

Looking at this issue provides the opportunity to move forward by learning from our mistakes, trying new tools and putting them into practice.

Fix the pattern.

(we look at the problem but we're part of the problem)

- Define what we are trying to achieve.
  - Update containers in our environment centrally.
  - Maintain the best security model.
    - for the colo
    - for the house lan
  - Look at alternative toolsets
  - test each of tool sets
    - Initialize and update containers with the tools/configurations we want universaly (python3, nano, ssh, usw/ssh-keys,admin user, usw)
    - Update the containers
- Impliment the desired outcomes using each of the tools based on the goals above.
- Select the best toolset and patterns and start using them.

Getting to it

ANSIBLE, CLOUD-INIT, AND LXD/INCUS

USE CASES

Initialization

Ad Hoc Updates (ssh-keys root passwds usw)

SECURITY MODEL

4.4 Kubernetes

4.4.1 If I have to kubernetes I am gonna certify

hash tag just kill me already.

Installing k8s on incus

It turns out I cant install vms on my home servers because they do not have sufficient resources. So we are installing our test cluster (for the Certified Kubernetes Administrator test) at the colo The course work is based on ubuntu 24.04 but we use RockyLinux at work so my original intention was to build a R9 cluster at home and Ubuntu/focal at work. However the instruction is so specific to the platform I am going to start on Ubuntu at the colo and at work and then rebuild the cluster using R9 on the cluster.

ATTEMPT #1

Notes on Ubuntu Install

linkdump

- https://rudimartinsen.com/2023/12/29/kubernetes-cluster-on-vms-2024/

4.4.2 YOU ARE HERE FLESHING THIS IN

```
apt update && apt upgrade -y
apt install apt-transport-https software-properties-common ca-certificates socat wget curl nano -y
swapoff -a
modprobe overlay
modprobe br_netfilter
cat << EOF | tee /etc/sysctl.d/kubernetes.conf
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
net.ipv4.ip_forward = 1
EOF
sysctl --system
mkdir -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
chmod a+r /etc/apt/keyrings/docker.gpg
cat >/etc/apt/sources.list.d/docker.list<<EOD
deb [arch=amd64 signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu focal stable
EOD
apt update && apt install containerd.io -y
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
chmod a+r /etc/apt/keyrings/kubernetes-apt-keyring.gpg
cat >/etc/apt/sources.list.d/kubernetes.list  << EOD
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /
EOD
apt update
apt-get install -y kubeadm=1.31.1-1.1 kubelet=1.31.1-1.1 kubectl=1.31.1-1.1
apt-mark hold kubelet kubeadm kubectl
```

4.5 No canonical

4.5.1 Stick a fork in them and turn them over they're done

Recently we had to rebuild one of our servers because the hardware raided boot disk that had behaved perfectly for years shat on the boot disk. I tried to remotely (via ipmi/virtualized serial)install a zfs rooted ubuntu (24.04) I waisted 2 man weeks. Then I spent an entire weekend and more money than a surplus dell r7xx driving to the colo (6 hrs away) and installing it (desktop install converted to zfs/server because thats the only pathway provided only to have it implode when implimenting the mirrors. We finally (with remote hands) installed 24.04 server on a single disk but at this point I was very much done. After a over a decade of using ubuntu as my primary server operating system Canonical has driven it into the dirt.

There are no polite words for the behaviour of Canonical with regards to sane headless installs with redundant root volumes.

There are other issues to take up not the least of which is their destruction of LXD.

I think the last straw was how unfriendly and unhelpfull the #ubuntu irc channel was. They used to be wrong but friendly.

No really (queue the intro to freaky styley)

We are in the process of reducing our colo presence from 2 servers to one server and replacing our firewall. While there we will migrate from lxd to incus and then from Ubuntu to the underlying Debian. From there we will continue to migrate the containers running Ubuntu to Debian and other well behaved operating systems.

DEBIAN

In this particular use case, Debian not only allowed me to install a system over a virtual serial console but allowed me to install from the original ubuntu. So, I replaced ubuntu server with debian on the secondary lxd server (with a temporary dual boot pathway back). Then I migrated the containers on the main server to it and repeated the process on our primary server.

INCUS

Canonical really pooched the LXD service. First with its insistance of using snaps for everything, second by alienating its lead developers, and finally changing the licensing from an open source platform to a less permissive licence. There are no polite words.

So I am migrating all of my lxd implimentations to the open source incus project.

The plan

SERVERS

Test.(done)

- Migrate non critical home lxd servers (utah/costello) to Debian. (done)
- Migrate lxd to incus on first two home servers. (done)
- Test incus to incus container migration.(done)

Repeat process at colocation (done)

- Migrate secondary server to debian/incus. (done)
- Migrate primary server to incus. (done)
- Move containers temporarily to secondary server. (done)
- Migrate primary server to debian. (done)
- Migrate containers back to primary (done).
- Down secondary server (done).

FIREWALL/ROUTER.

This work is described at https://www.digithink.com/buildnotes/using-a-tank-for-crowd-control/

- Set up wireguard (done)
- Set up secondary dns (done)
- Set up tinyproxy (done)
- Set up dnsmasq for admin lan.
- Harden the whole mess with pf.

CONTAINERS

- Rebuild containers using debian (or other well behaved operating systems).

FIN.

4.5.2 Debian install on kh2024

Debian allows us to install an operating system on to a set of partitions on a running linux system. It also does not require a fucking vga monitor. (hey canonical you suck!)

https://www.debian.org/releases/stable/amd64/apds03.en.html

```
parted /dev/sdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) p
Model: ATA TEAM T2532TB (scsi)
Disk /dev/sdb: 2048GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name  Flags
 1      1049kB  200GB   200GB   ext4         /
 2      200GB   201GB   1000MB  fat32        2     boot, esp
 3      201GB   2048GB  1847GB  ext4         3

(parted) rm 3
(parted) mkpart 3 ext3 201GB 501GB
(parted) p
Model: ATA TEAM T2532TB (scsi)
Disk /dev/sdb: 2048GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End    Size    File system  Name  Flags
 1      1049kB  200GB  200GB   ext4         /
 2      200GB   201GB  1000MB  fat32        2     boot, esp
 3      201GB   501GB  300GB   ext2         3

(parted) mkpart 4 linux-swap 501GB 600GB
(parted) mkpart 5 ext4 600GB 100%
(parted) print
Model: ATA TEAM T2532TB (scsi)
Disk /dev/sdb: 2048GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name  Flags
 1      1049kB  200GB   200GB   ext4         /
 2      200GB   201GB   1000MB  fat32        2     boot, esp
 3      201GB   501GB   300GB   ext4         3
 4      501GB   600GB   99.0GB               4     swap
 5      600GB   2048GB  1448GB  ext4         5

(parted) q
Information: You may need to update /etc/fstab.
```

If you are on ubuntu or anoter debian based system you can install debootstrap using apt. Otherwise follow the bouncing prompt from the link above.

```
apt install debootstrap
```

set up target root filesystem

```
mke2fs -j /dev/sdb3
```

bootstrap the root file system

```
mkdir /mnt/debinst
mount /dev/sdb3 /mnt/debinst
debootstrap --arch amd64 bookworm /mnt/debinst http://ftp.us.debian.org/debian
```

set up mounts for chroot.

```
mount -t proc proc /mnt/debinst/proc
mount -t sysfs /sys /mnt/debinst/sys
mount --bind /dev /mnt/debinst/dev
mount --bind /dev/pts /mnt/debinst/dev/pts
LANG=C.UTF-8 chroot /mnt/debinst /bin/bash
```

BACK ON THE CHROOT

```
.... second stage may not be needed ....
/debootstrap/debootstrap --second-stage
.... if you did the above this shouldnt be needed either ....
apt install makedev
cd /dev
MAKEDEV generic
nano /etc/fstab
mount -a
nano /etc/adjtime
...
0.0 0 0.0
0
UTC
...
dpkg-reconfigure tzdata
ip a
apt-install bridge-utils
nano /etc/network/interfaces
nano /etc/resolv.conf
echo kh2024>/etc/hostname
nano /etc/hosts
apt install locales
dpkg-reconfigure locales
apt install grub2
... not sure about this one ...
grub install /dev/sdb
mount -a
apt install linux-image-amd64
apt install grub2
grub-update
grub-install /sdb
grub2-install /sdb
nano /etc/default/grub

update-grub2
```

make sure we can log into the box

```
apt install openssh-server
nano /etc/ssh/sshd_config
... disable root ssh login with password ...
adduser joe
adduser feurig
vigr
apt install sudo
visudo
passwd -u root
passwd root
nano ~root/.ssh/authorized_keys
```

copy the entire /etc/ssh/ directory so the host keys dont change

```
exit
cp -rpv /etc/ssh /mnt/debinst/etc/
LANG=C.UTF-8 chroot /mnt/debinst /bin/bash
```

Install gb ethernet firmware

```
nano /etc/apt/sources.list
...
root@kh2024:~# cat /etc/apt/sources.list
#deb http://ftp.us.debian.org/debian bookworm main
deb http://deb.debian.org/debian bookworm main non-free non-free-firmware contrib
deb http://deb.debian.org/debian bookworm-updates main non-free non-free-firmware contrib
deb http://deb.debian.org/debian-security/ bookworm-security main non-free non-free-firmware contrib
deb http://deb.debian.org/debian bookworm-backports main contrib non-free non-free-firmware
...
apt update
apt install firmware-bnx2
update-grub
update-grub2
nano /etc/default/grub
reboot
```

Install zfs 2.2 from bookworm-backports

```
nano /etc/apt/sources.list
...
#deb http://ftp.us.debian.org/debian bookworm main
deb http://deb.debian.org/debian bookworm main non-free non-free-firmware contrib
deb http://deb.debian.org/debian bookworm-updates main non-free non-free-firmware contrib
deb http://deb.debian.org/debian-security/ bookworm-security main non-free non-free-firmware contrib
deb http://deb.debian.org/debian bookworm-backports main contrib non-free non-free-firmware
...
apt update
apt-get install linux-headers-$(uname -r)
```

```
apt -t bookworm-backports install zfs-dkms zfs-zed zfsutils-linux
zpool import -f tank
apt install parted
```

clean zfs info off of old devel and infra disk.

```
wipefs -a /dev/sde
```

```
nano /etc/sysctl.conf
...
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1
...

sysctl net.ipv6.conf.all.disable_ipv6
```

4.5.3 incus install on kh2024

Currently just the bones. You are here fleshing them in.

This is on a system that has been installed as described in the debian link.

This requires that you modify the /etc/apt/sources.list to include the backports.

```
root@kh2024:~# cat /etc/apt/sources.list
#deb http://ftp.us.debian.org/debian bookworm main
deb http://deb.debian.org/debian bookworm main non-free non-free-firmware contrib
deb http://deb.debian.org/debian bookworm-updates main non-free non-free-firmware contrib
deb http://deb.debian.org/debian-security/ bookworm-security main non-free non-free-firmware contrib
deb http://deb.debian.org/debian bookworm-backports main contrib non-free non-free-firmware
```

Set up the network

```
apt -t bookworm-backports install incus incus-tools
apt install bridge-utils
nano /etc/network/interfaces
....
# https://ip4calculator.com
source /etc/network/interfaces.d/*

auto lo
iface lo inet loopback

iface eno2 inet manual
iface eno3 inet manual
iface eno4 inet manual

auto br0
iface br0 inet static
     address 198.202.31.158
     network 198.202.31.128
     netmask 255.255.255.128
     broadcast 198.202.31.255
     gateway 198.202.31.129
     bridge_ports eno4
     bridge_stp off       # disable Spanning Tree Protocol
        bridge_waitport 0    # no delay before a port becomes available
        bridge_fd 0          # no forwarding delay

auto eno1
iface eno1 inet static
     address 192.168.31.158
     network 192.168.31.0
     netmask 255.255.255.0
     broadcast 192.168.31.255
     #gateway 192.168.31.2
...
systemctl restart networking
ip a
```

Set up a partition for containers.

```
fdisk -l |grep -v loop|grep Disk
df -k
fdisk -l /dev/sdc
fdisk -l /dev/sdd
fdisk -l /dev/sdb
parted /dev/sdb
(parted) rm 5
(parted) mkpart 600GB 900GB
(parted) mkpart 900GB 100%
(parted) print
Model: ATA TEAM T2532TB (scsi)
Disk /dev/sdb: 2048GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system     Name  Flags
 1      1049kB  200GB   200GB   ext4            /
 2      200GB   201GB   1000MB  fat32           2     boot, esp
 3      201GB   501GB   300GB   ext3            3
 4      501GB   600GB   99.0GB  linux-swap(v1)  4     swap
 5      600GB   900GB   300GB                   5
 6      900GB   2048GB  1148GB                  6

(parted) quit
ls -lsa /dev/disk/by-id/|grep sdb
...
```

```
0 lrwxrwxrwx 1 root root  10 Oct 31 09:30 ata-TEAM_T2532TB_TPBF2402200040201609-part5 -> ../../sdb5
...
```

Install and initialize incus. - existing bridge is br0 - zfs pool is on /dev/disk/by-id/ata-TEAM_T2532TB_TPBF2402200040201609-part5

To install 6.0 install from bookworm-backports.

```
apt -t bookworm-backports install incus incus-tools
incus admin init
```

To install the latest you need to follow the directions at https://github.com/zabbly/incus

```
apt install curl
curl -fsSL https://pkgs.zabbly.com/key.asc -o /etc/apt/keyrings/zabbly.asc
sh -c 'cat <<EOF > /etc/apt/sources.list.d/zabbly-incus-stable.sources
Enabled: yes
Types: deb
URIs: https://pkgs.zabbly.com/incus/stable
Suites: $(. /etc/os-release && echo ${VERSION_CODENAME})
Components: main
Architectures: $(dpkg --print-architecture)
Signed-By: /etc/apt/keyrings/zabbly.asc

EOF'
apt update
apt install incus incus-tools
```

Pull down the images you know you are going to use.

```
incus image list images: bookworm
incus image copy images:debian/12/cloud local:
incus image list images: trixie
incus image copy images:debian/trixie/cloud local:
incus image alias create bookworm 4ed6d8b34c84
incus image alias create trixie b0104c654d3d
incus image list
```

Set up a profile. This should be edited for things that no longer matter.

```
incus profile create susdev24<<EOD
name: susdev24
description: Try to create a sane environment for cloud-init based operating systems
config:
  user.network-config: |
    version: 1
    config:
      - type: physical
        name: eth0
        subnets:
          - type: static
            ipv4: true
            address: 198.202.31.200
            netmask: 255.255.255.128
            gateway: 198.202.31.129
            control: auto
      - type: nameserver
        address:
          - 198.202.31.132
          - 8.8.8.8
  user.user-data: |
    #cloud-config
    timezone: America/Vancouver
    users:
      - name: feurig
        passwd: "REDACTED"
        gecos: Donald Delmar Davis
        ssh-authorized-keys:
          - REDACTED
        groups: sudo,root,wheel
        shell: /bin/bash
      - name: joe
        passwd: "REDACTED"
        gecos: Joseph Wayne Dumoulin
        ssh-authorized-keys:
          - REDACTED
        groups: sudo,root,wheel
        shell: /bin/bash
    manage_resolv_conf: true
    packages:
    - python3
    - python-is-python3
    - python2
    - nano
    - openssh-server
```

```
    - less
    package_update: true
    package_upgrade: true
    write_files:
    - path: /etc/resolv.conf.static
      permissions: '0644'
      owner: root:root
      content: |
        nameserver 198.202.31.141
        nameserver 8.8.4.4
        search suspectdevices.com fromhell.com vpn
    - path: /usr/local/bin/update.sh
      permissions: '0774'
      owner: root:root
      content: |
        #!/bin/bash
        # update.sh for debian/ubuntu/centos  (copyleft) don@suspecdevices.com
        echo ---------------------- begin updating `uname -n` ----------------------
        if [ -x "$(command -v apt-get)" ]; then
          apt-get update
          apt-get -y dist-upgrade
    # and the users are locked by default
    # cloud cart blanch accounts are inexcusable
    - sed -i "s/^127.0.0.1/#127.0.0.1/" /etc/hosts
    - echo 127.0.0.1 `hostname` localhost >>/etc/hosts
    - passwd joe -u
    - passwd feurig -u
    - userdel -f ubuntu
    - userdel -f centos
    - userdel -f opensuse
    - mv /etc/resolv.conf /etc/resolv.conf.foobarred
    - ln -s /etc/resolv.conf.static /etc/resolv.conf
    - netplan apply
    - apt-get install -y openssh-server nano less
    - apt-get install -y python-is-python3
    - apt-get install -y python
    power_state:
       mode: reboot
       message: See You Soon...
       condition: True
EOD
```

Launch a container

```
incus launch local:bookworm teddy -p default -p susdev24
incus list
incus exec teddy bash
```

YOU ARE HERE ADDING A SECTION ON INCUS TO INCUS TRUST....

4.6 Norouter

4.6.1 Agent forwarding example

This is an example of using agent forwarding to access hosts on the admin lan.

```
feurig@Amyl ~ % ssh-add
Identity added: /Users/feurig/.ssh/id_rsa (feurig@nix.lan)
feurig@Amyl ~ % ssh -A sitka.suspectdevices.com
Last login: Sat Oct 12 20:39:27 2024 from 209.66.79.150
FreeBSD 14.1-RELEASE (GENERIC) releng/14.1-n267679-10e31f0946d8

Welcome to FreeBSD!
...
feurig@sitka:~ $ ssh root@192.168.31.158
The authenticity of host '192.168.31.158 (192.168.31.158)' can't be established.
ED25519 key fingerprint is SHA256:GFIX+l6M/OOI5BJVWK1U1H5lKzDCkzbXNEgX1Sn7rK0.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.31.158' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-45-generic  x86_64)
...
Last login: Sat Oct 12 20:29:44 2024 from 192.168.31.228
root@kh2024:~#
```

```
Connection to virgil.suspectdevices.com closed.
feurig@Amyl ~ % ssh -A feurig@virgil
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-45-generic x86_64)
...
feurig@virgil:~$ ssh 192.168.31.2
The authenticity of host '192.168.31.2 (192.168.31.2)' can't be established.
ED25519 key fingerprint is SHA256:pEu5SYscD/+jLLcwzoyPcemXS2AyuOkF9zkC5r5WjDg.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.31.2' (ED25519) to the list of known hosts.
Last login: Sat Oct 12 20:40:09 2024 from 209.66.79.150
FreeBSD 14.1-RELEASE (GENERIC) releng/14.1-n267679-10e31f0946d8
...
feurig@sitka:~ $
```

4.6.2 Wireguard and Tinyproxy

(add the freebsd configuration sections)

After working through the complexities of using headscale/tailscale I realized that I really only needed the colo router to do 2 things.

1. Provide us access to the admin lan (the servers and their remote consoles).

```
graph LR
D([192.168.31.0/24])<-->A[Host interface]
D<-->E[Host Drac/ILO]
C[laptop] <-- Wireguard --> B(sitka/virgil);
B <-- Wireguard -->D;
```

2. Allow the servers to reach the update repositories.

```
graph LR
B --> I([internet])
A[Host] -- Apt Via Proxy --> B(sitka/virgil);
```

To do this and to provide redundant routes to the admin lan we take two approaches.

1. Replace the router with a container.
2. Replace the router with a better one.

Replacing the colo router with a container.

By using a container with access to both the external lan and the admin lan we can set up wireguard and tinyproxy. Wireguard allows us to securely connect to the admin lan while tinyproxy allows the servers a mechanism to recieve software updates. This will become a staging/test setup for the colo firewall.

SETTING UP THE CONTAINER

To be able to do its job the container needed to be privilaged and it also would not run on 22.04. Its ok 22.04 still has a few years of support left.

```
root@aoc2024:~# lxc init ubuntu:22.04 homer -c security.privileged=true -p susdev23 -p infra
root@aoc2024:~# lxc config edit homer
name: homer
description: "wireguard/squid host"
...
devices:
  eth1:
    name: eth1
    nictype: bridged
    parent: br3
    type: nic
^x
root@aoc2024:~# lxc start homer
root@aoc2024:~# lxc exec homer bash
root@homer:~# nano /etc/netplan/50-cloud-init.yaml
network:
    version: 2
    ethernets:
        eth0:
            addresses:
            - 198.202.31.227/25
            nameservers:
                addresses:
                - 198.202.31.132
                - 8.8.8.8
                search:
                - suspectdevices.com
                - styx.suspectdevices.com
            routes:
            -   to: default
                via: 198.202.31.129
        eth1:
            addresses:
            - 192.168.31.227/24
^x
root@homer:~# netplan apply
```

Backcheck the interfaces

```
root@homer:~# 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
45: eth0@if46: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 00:16:3e:ba:f0:be brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 198.202.31.228/25 brd 198.202.31.255 scope global eth0
       valid_lft forever preferred_lft forever
47: eth1@if48: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 00:16:3e:2e:6f:d8 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 192.168.31.228/24 brd 192.168.31.255 scope global eth1
       valid_lft forever preferred_lft forever
```

INSTALL PREREQUISITES AND ENABLE IP FORWARDING

The next few sections are done on the gateway container (virgil)

```
apt install wireguard
apt install resolvconf
sysctl -w net.ipv4.ip_forward=1
```

Wireguard

SET UP WIREGUARD

Server Setup

```
cd /etc/wireguard/
wg genkey | sudo tee private.key
chmod go= private.key
cat private.key | wg pubkey | sudo tee public.key
wg genpsk |tee preshared.psk
nano /etc/wireguard/wg0.conf
```

```
# wg0.conf
[Interface]
Address = 10.0.0.1/32
ListenPort = 1194
PrivateKey = <<contents of private.key>>
PostUp = iptables -A FORWARD -i %i -j ACCEPT; iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
PostDown = iptables -D FORWARD -i %i -j ACCEPT; iptables -t nat -D POSTROUTING -o eth1 -j MASQUERADE

# merlot
[Peer]
PublicKey = <<contents of public.key>>
AllowedIPs =  10.0.0.2/32,192.168.128.0/17
PresharedKey = <<contents of preshared.key>>

# amyl dons laptop
[Peer]
PublicKey = <<key from wireguard client>>
AllowedIPs = 10.0.0.6/32
PresharedKey = <<contents of preshared.key>>
```

Enable it

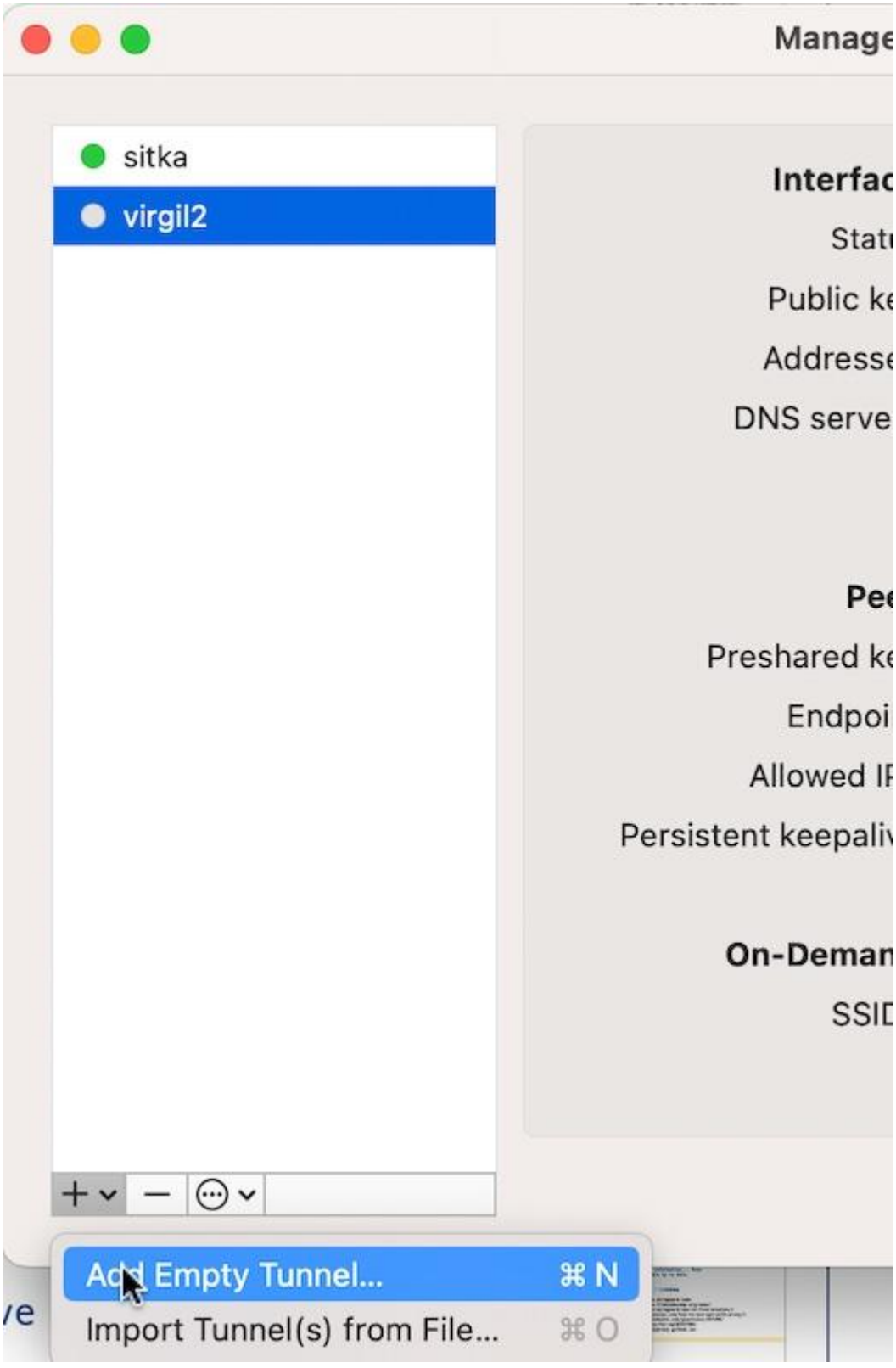
```
wg-quick up wg0
systemctl enable wg-quick@wg0
```

Client Configuration.

To avoid contention please reference the ips spreadsheet under the 10.0.0.x tab.

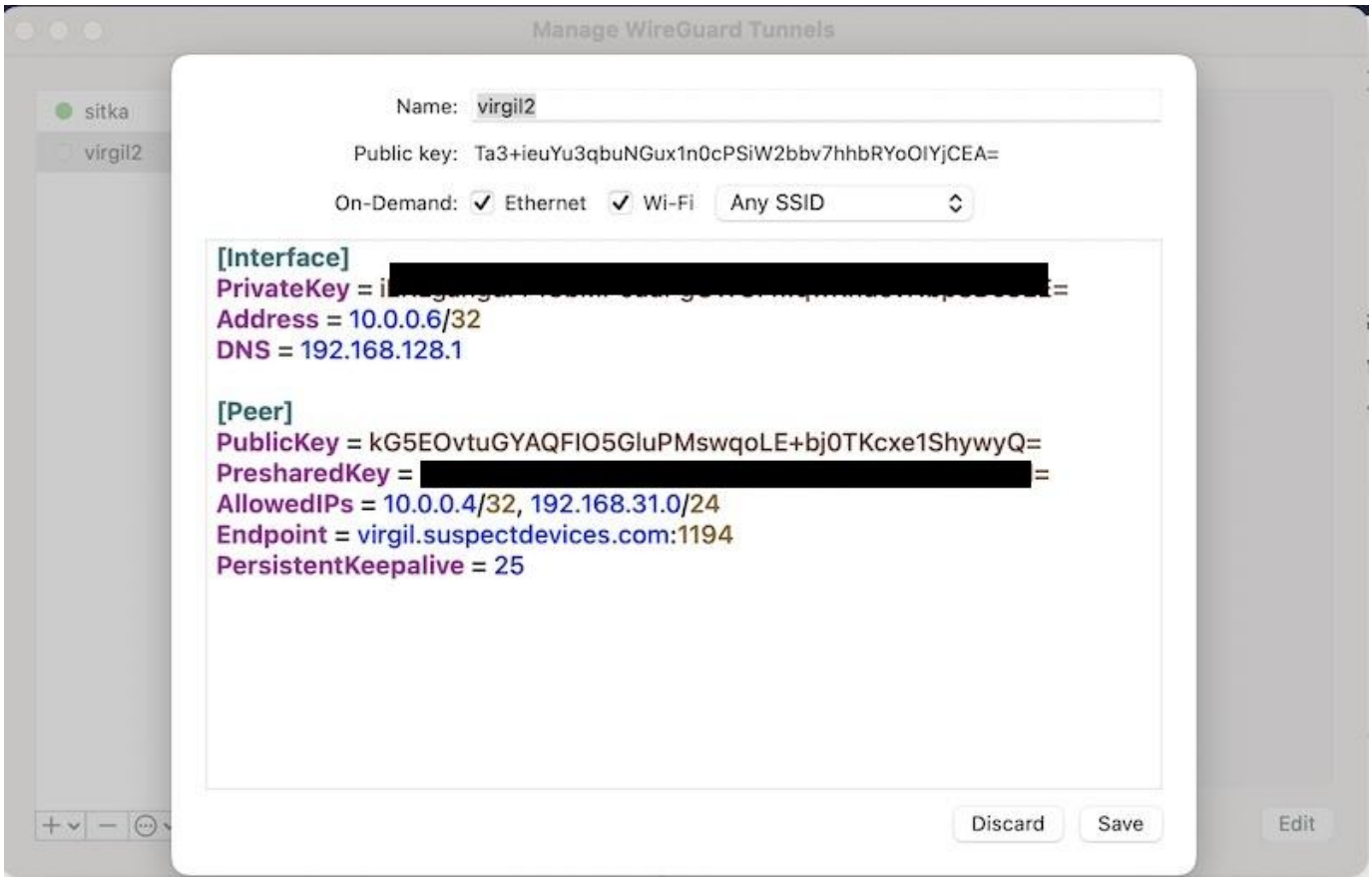
MacOs client

To add the wireguard server to macos go to manage-tunnels and hit the + ->Add empty tunnel.... \



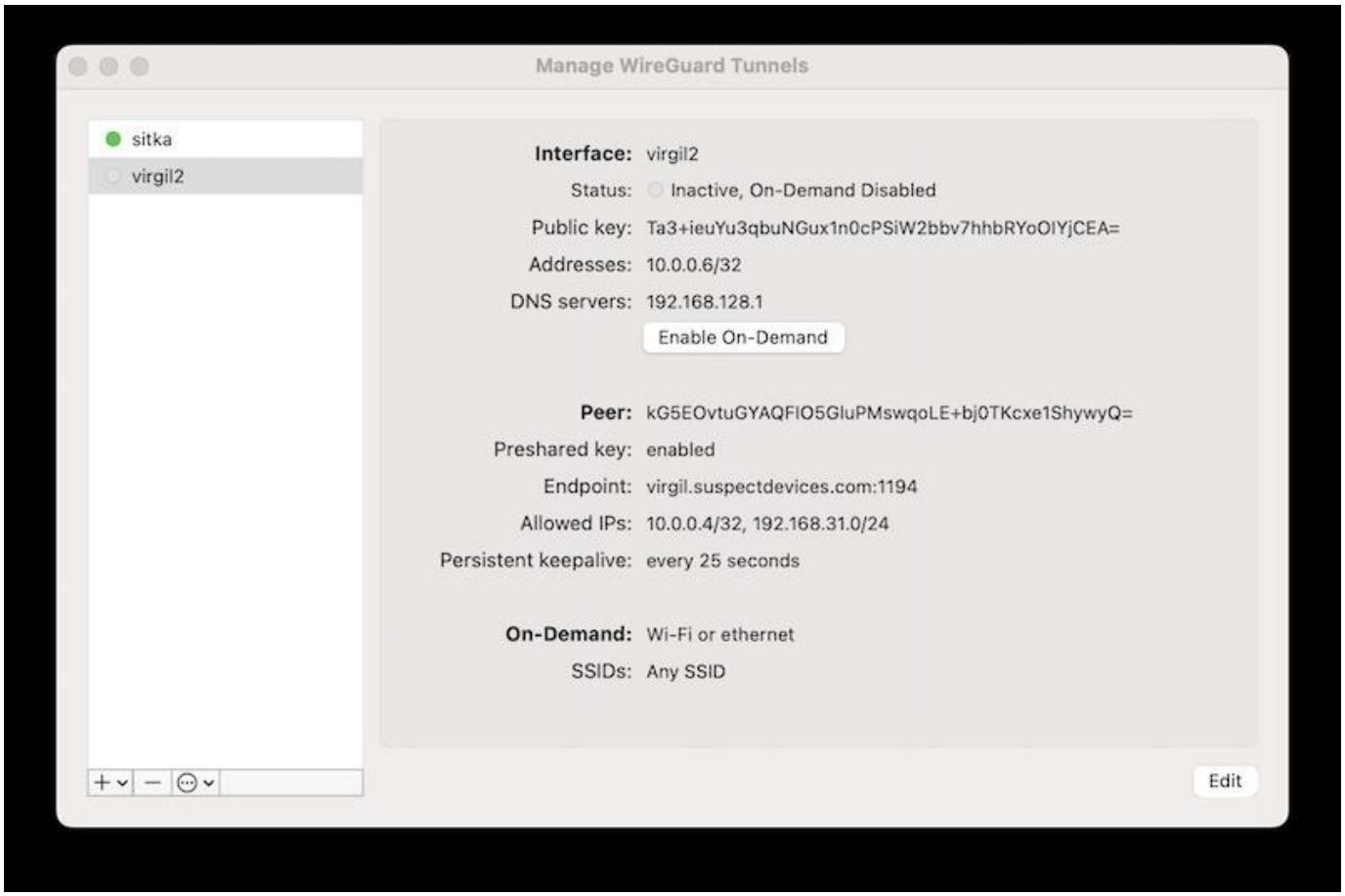
You will get a form which includes the clients public key, its private key and a lot of white space.

Use the public key to fill in the peer section on the server (as pre done above) and then flesh in the local interface and peer details.



The interface address is the unique address of the peer on the wireguard network. The interface dns will be the local dns server.

The peer data can be found in the files cited above. When finished press save. The AllowedIPs should include the wireguard servers wg address and any ips routed through it (in the above case the admin lan at the colo).



To select the connection double click on the tunnel and press the Enable On-Demand button.

Then test it.

```
feurig@Amyl ~ % ssh root@192.168.31.159
...
root@tk2022:~#
```

Linux client

Example setup using virgil as server....

On the client generate the pub/private keys.

```
apt install wireguard resolvconf
cd /etc/wireguard/
wg genkey > private.key
wg pubkey < private.key > public.key
chmod o-rwx *
```

on the server add the client as a peer.

```
nano /etc/wireguiard/wg0.conf
...
# otto
[Peer]
PublicKey = <public key from above>
AllowedIPs = 10.0.0.8/32
PresharedKey = <preshared key from server setup>
...
^X
systemctl restart wg-quick@wg0
```

On the client create configuration with server as peer.

```
cd /etc/wireguard
nano wg0.conf
[Interface]
```

```
PrivateKey = <private key from above>
Address = 10.0.0.8/32
DNS = 192.168.31.141

[Peer]
PublicKey = <public key from server>
PresharedKey = <preshared key from server setup>
AllowedIPs = 10.0.0.4/32, 192.168.31.0/24
Endpoint = virgil.suspectdevices.com:1194
PersistentKeepalive = 25
^X
wg-quick up wg0
ping 192.168.31.2
```

No ~~Squid~~

The last update to squid completely overwrote its working configuration file without even making a backup copy. Can you say exposure and disfunction? FRACK THAT. IT'S GONE.

```
root@virgil:/etc/squid# apt remove --purge squid
```

TinyProxy -- proxy for main server and router.

SETTING UP TINYPROXY

The example below is on virgil (x.x.x.228) sitka is described in her build notes (x.x.x.2)

```
apt install tinyproxy -y
systemctl enable tinyproxy
cd /etc/tinyproxy/
cp tinyproxy.conf tinyproxy.conf.noisy
grep -v "^\#" tinyproxy.conf.noisy |grep -v "^$" >tinyproxy.conf
nano tinyproxy.conf
```

```
User tinyproxy
Group tinyproxy
Port 3128
Listen 192.168.31.228
Timeout 600
DefaultErrorFile "/usr/share/tinyproxy/default.html"
StatFile "/usr/share/tinyproxy/stats.html"
LogLevel Info
PidFile "/run/tinyproxy/tinyproxy.pid"
MaxClients 10
Allow 192.168.31.1/24
ViaProxyName "tinyproxy"
```

```
systemctl enable tinyproxy
systemctl start tinyproxy
```

TEST THE PROXY

```
root@kb2018:~# curl -x 192.168.31.228:3128 http://archive.ubuntu.com/ubuntu
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>301 Moved Permanently</title>
</head><body>
<h1>Moved Permanently</h1>
<p>The document has moved <a href="http://archive.ubuntu.com/ubuntu/">here</a>.</p>
<hr>
<address>Apache/2.4.52 (Ubuntu) Server at archive.ubuntu.com Port 80</address>
</body></html>
```

SET UP APT TO USE PROXY

```
nano /etc/apt/apt.conf.d/80proxy.conf
```

```
Acquire::http::Proxy "http://192.168.31.227:3128/";
```

TEST APT THROUGH PROXY

```
root@aoc2024:/etc/apt/apt.conf.d# ip route delete default
root@aoc2024:/etc/apt/apt.conf.d# ip route
192.168.31.0/24 dev br3 proto kernel scope link src 192.168.31.158
root@aoc2024:/etc/apt/apt.conf.d# apt update
Hit:1 http://us.archive.ubuntu.com/ubuntu noble InRelease
Get:2 https://pkgs.tailscale.com/stable/ubuntu noble InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu noble-updates InRelease
```

```
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:5 http://us.archive.ubuntu.com/ubuntu noble-backports InRelease
Fetched 6575 B in 1s (9357 B/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
```

References / Linkdump

- https://www.wireguard.com
- https://www.freecodecamp.org/news/build-your-own-wireguard-vpn-in-five-minutes/
- https://linuxiac.com/how-to-use-apt-with-proxy/
- https://askubuntu.com/questions/257290/configure-proxy-for-apt#257296
- https://tinyproxy.github.io

4.7 Sense

4.7.1 Start making sense

At work we use pf on freebsd for our firewalls. I have been re-learning it as my freebsd-firewall experience is over 2 decades old. At home and in the colo we have been using openwrt which is great but a real pain to keep updated and deploy. I have been looking at pfsense and in the process, I discovered opnsense. If my BSD/PF chops ever get good enough I may go to straight freebsd but the convenience of guided configuration of a secure system is hard to ignore [1]

The hardware

As I was considering looking at pfsense I scored a pair of routers with 6x1G ports, and room for a pair of ssds.

First thing I did was to pull the os disk and replace it with a 1T ssd and upgrade the memory. The second thing I did was to replace the fans with quieter ones and print a pair of noise reducing mufflers. (I should blog about this on suspect devices at some point)

After that I installed opensense and started working on my list of things to do.

THE GOAL: CONSOLIDATE HOME NETWORK USING OPNSENSE

- [x] REMOVE openwrt router
- [x] REMOVE dedicated caching server (Done)
- [x] REMOVE dedicated dnsmasq server
- [ ] REMOVE (that f**king centurylink router)
- [x] KEEP Pihole-FTL dns based blacklisting
- [ ] ADD Better firewall rules
- [ ] ADD VPN acces to home network
- [ ] ADD Isolated Wireless network for solar array controller.

FOOTNOTES/SARCASMS

1). On the other hand having a gui make things easier makes it easy to break things and less easy to debug them. (having managed to brick the home network trying to add an isolated wireless network)

4.7.2 Centurylink fiber

Linkdump

- https://gist.github.com/matracey/12cc7c51297561f49b4d1a95b68abc45
- https://forum.netgate.com/topic/83139/pppoe-on-wan-link-for-centurylink-gigabit-service/23
- https://www.centurylink.com/home/help/internet/modems-and-routers/third-party-modem-support-and-settings.html
- https://www.tp-link.com/us/support/faq/2709/

4.7.3 Keeping Pihole-ftl while moving to opnsense.

Not sure this is the best way.

Linkpile

- https://discourse.pi-hole.net/t/opnsense-pihole/54818
- https://pi-hole.net/blog/2021/09/30/pi-hole-and-opnsense/
- https://discourse.pi-hole.net/t/first-timer-using-opnsense-and-pi-hole-guide/61694
- https://github.com/pi-hole/FTL
- TODO: look at Adguard Home >>>https://www.reddit.com/r/OPNsenseFirewall/comments/tqzijy/want_to_have_a_pihole_plugin_for_opnsense_express/

4.7.4 Initial impression of opnsense

.

This is mostly a note about freebsd audit and why I went with opnsense. One of my coworkers didnt like some of the coding last time he looked at opnsense, but I am willing to ignore this while I work on being able to do most of this stuff by hand.

pkg audits and updates.

Out of the box pfsense-ce (2.6..) had over 20 vulnerabilities most of them in the core parts of the system. With an older version of freebsd and no real upgrade path I thought "well obscene me, this obscenes". This was really a deal breaker.

OPNsense on the other hand came out of the box with around a dozen which after a pgk update && pkg upgrade dropped down to one. This is recent, not critical and consistent with the upgrades I have been doing at work. Bodes well.

```
root@OPNsense:~ # pkg audit -F
vulnxml file up-to-date
py39-setuptools-63.1.0 is vulnerable:
  py39-setuptools -- denial of service vulnerability
  CVE: CVE-2022-40897
  WWW: https://vuxml.FreeBSD.org/freebsd/1b38aec4-4149-4c7d-851c-3c4de3a1fbd0.html

1 problem(s) in 1 installed package(s) found.
root@OPNsense:~ # freebsd-version
13.1-RELEASE-p5
```

Link pile.

- https://forum.opnsense.org/index.php?topic=18274.0
- https://connortumbleson.com/2022/06/06/opnsense-wireguard-pihole/
- https://homegrowntechie.com/discovering-migrating-to-opnsense/

4.7.5 Reconsidering OpnSense

My initial tact was to take opnsense and use it as a prototype for the underlying freebsd based software (pf,dnsmasq,usw) That is the configuration at the colo and it works well in that environment. However recently I started looking at what opnsense out of the box brings to the table. In particular I started looking at the total pile of shit that my centurylink provided router was letting into my network. And I decided that if I was hand rolling pf I would not have caught half of it.

I was in the middle of converting everything to /etc/ethers+/etc/hosts+dnsmasq and I said heck, lets just do the same in opnsense. Then we can look at getting rid of the pile of hot garbage that centurylink is charging me $15 a month for.

Well. That didn't work

So I turned on the dnsmasq dns and added all of the hosts in my network to /etc/hosts and /etc/ethers It seemed to work but the next time I did an update it overwrote both files, and stopped resolving the hosts I used most. So the main takaways were.

1. It seems to work at first.
2. It overwrites your files.
3. You have to manually add everything using the gui.
4. It's not automatable or scriptable.
5. It doesn't work.

So I turned unbound back on and looked at the alternatives.

You can't configure the software/services but you can run a jail.

So this is going to be a longer process than I would have liked but I have a test system to build the jail on.

Linkdump.

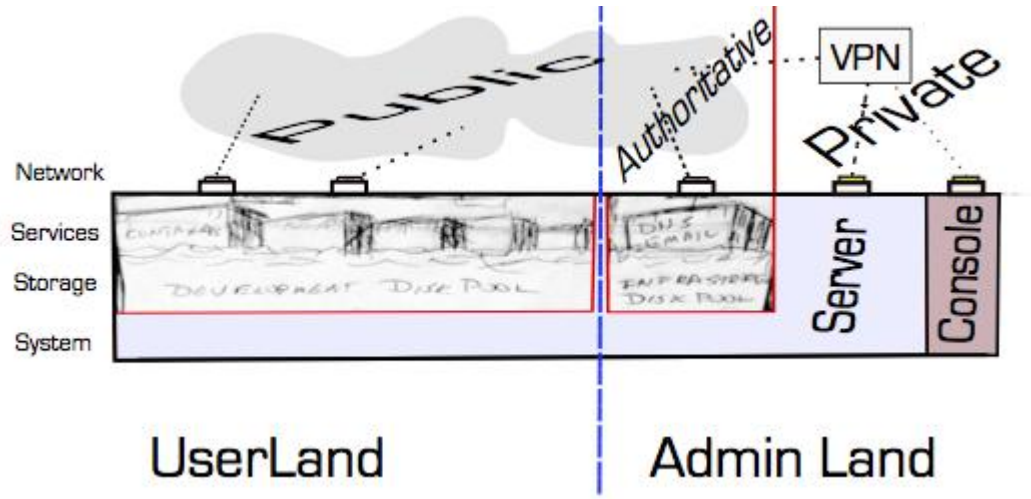
- https://forum.opnsense.org/index.php?topic=26975.0
- https://www.reddit.com/r/opnsense/comments/sjewa4/jails_under_opnsense_221/?rdt=59758
- https://forum.opnsense.org/index.php?topic=26724.0

5. Serverdocs

5.1 Server Modernization

This is an ongoing project that hosts several domains. (please refer to the operations guide for current setup and maintainence)

5.1.1 Overview



Phase I (2017-2018)

Phase one of the server modernization shifted away from multipurposed servers and kvms to lxc/lxd based containers.

- Moving all legacy system functions onto separate linux containers isolated from each other.
- Use mirrored disk systems to insure that disk corruption does not lead to data corruption.
- Start giving a shit about the systems, code, and sites on them.
- Own your code/data. (If your free code hosting system is shutdown or taken over by Microsoft is it really free?)

Server Modernization Phase II (2019-2021)

Phase two extends on this by integrate Ansible into system maintenance tasks.

- Integrate Ansible into system maintenance tasks
- Reevaluate Centos and other RPM based containers built using playbooks vs profiles/scripts/cloud-init while maintaining current security model
- Clean up the cruft (If it doesn't bring you joy DTMFA)

SMP III Own Your Shit (2022-2023)

- Work on secure and efficient traffic in and out of home lans (Privoxy,DNS based ad blocking,squid etc)
- Continue to refine server operation/maintanance.
- Build out content.
- git to markdown automation
- Rethink openwrt based routing.
- explore opnsense
- Document original home server/network setup

- Rethink everything

SMP IV Keep going (2024-)

- Reduce colo footprint.
- remove the dell.
- Dump Canonical
- debian
- incus
- Adapt Freebsd based firewall/router
- wireguard
- dnsmasq
- pf
- Make shit happen
- Build out content.
- Start new projects.
- Distribute data and backups over the network to home servers.

Goals

- Security
- Flexibility
- Simplification

Isolation

- network
- performance
- disk

5.2 ILO Command Line Notes

5.3 Linkdump

- https://serverfault.com/questions/489865/batch-reset-all-ilo-passwords-via-command-line
- https://download.ni.com/support/manuals/377263a.pdf

5.4 Dl380 Raid Bios notes

YOU ARE HERE Flesh this out with new stuff we learned...

5.4.1 Configuring the disks using the raid controller bios

```
steve:~ don$ ssh kates-ilo.suspectdevices.com
User:feurig logged-in to kb2018.suspectdevices.com(192.168.31.119 / FE80::9E8E:99FF:FE0C:BAD8)
iLO 3 Advanced for BladeSystem 1.88 at  Jul 13 2016
Server Name: kb2018
Server Power: On

</>hpiLO-> vsp

Virtual Serial Port Active: COM2

Starting virtual serial port.
Press 'ESC (' to return to the CLI Session.

root@kb2018:~# fdisk -l|grep Disk\ \/
Disk /dev/loop0: 86.9 MiB, 91099136 bytes, 177928 sectors
Disk /dev/loop1: 87.9 MiB, 92164096 bytes, 180008 sectors
Disk /dev/loop2: 63.4 MiB, 66486272 bytes, 129856 sectors
Disk /dev/sda: 136.7 GiB, 146778685440 bytes, 286677120 sectors

root@kb2018:~# reboot
[  OK  ] Stopped Stop ureadahead data collection 45s after completed          Stopping Session 98 of user feurig.
            Stopping Availability of block devices...
    ...

[  OK  ] Reached target Shutdown.
[  OK  ] Reached target Final Step.
            Starting Reboot...
[292357.910620] reboot: Restarting system



After several seconds you will see a text based bios screen
[[Image(CaptiveRaidController:ILo3SSHConsoleBooting.png)]]
After the network controller is started the raid controller will give you a chance to configure it.
_PRESS F8 NOW!! _ \\
[[Image(CaptiveRaidController:PressF8NOW.png)]]

If you miss it you will have to escape back to the ILO3 and power cycle the machine. _(This is ok because the disks are not active until the machine actually
boots)_

Booting from Hard Drive C:
<ESC> (
</>hpiLO-> power off hard

status=0
status_tag=COMMAND COMPLETED
Wed Sep 26 15:31:57 2018



Forcing server power off .......
Please wait 6 seconds for this operation to complete.



</>hpiLO-> power

status=0
status_tag=COMMAND COMPLETED
Wed Sep 26 15:32:04 2018



power: server power is currently: Off


</>hpiLO-> power on

status=0
status_tag=COMMAND COMPLETED
Wed Sep 26 15:32:21 2018



Server powering on .......



</>hpiLO-> vsp

Virtual Serial Port Active: COM2
```

```
Starting virtual serial port.
Press 'ESC (' to return to the CLI Session.
```

Once in the raid controller bios you will get a main menu.

[[Image(CaptiveRaidController:ViewLogicalDrive.png)]]

If you select view logical drives will see that the first two disks are combined into a mirrored pair and that there are no other drives defined.

So we select "Create Logical Drive". Which gives us the following screen.

[[Image(CaptiveRaidController:CreateLogicalDriveDefaults.png)]]

Notice that the defaults are to create a raid 1+0 array with the first two matching disks. Deselecting either disk (down arrow, spacebar) will cause the raid configuration to automatically drop to RAID 0

Press Enter when finished. The next screen will ask you to verify the creation

Repeat this for each remaining disk.

When you are finished you can view the logical drives. [[Image(CaptiveRaidController:RaidConfFinished.png)]]

The key will walk you back out so you can continue to boot.

5.4.2 success

```
root@kb2018:~# fdisk -l|grep Disk\ \/
Disk /dev/loop0: 86.9 MiB, 91099136 bytes, 177928 sectors
Disk /dev/loop1: 87.9 MiB, 92164096 bytes, 180008 sectors
Disk /dev/loop2: 63.4 MiB, 66486272 bytes, 129856 sectors
Disk /dev/sda: 136.7 GiB, 146778685440 bytes, 286677120 sectors
Disk /dev/sdb: 223.6 GiB, 240021504000 bytes, 468792000 sectors
Disk /dev/sdc: 223.6 GiB, 240021504000 bytes, 468792000 sectors
Disk /dev/sdd: 279.4 GiB, 299966445568 bytes, 585871964 sectors
Disk /dev/sde: 279.4 GiB, 299966445568 bytes, 585871964 sectors
root@kb2018:~#
```

References

- https://www.n0tes.fr/2024/03/04/CLI-HPE-ssacli-and-hpssacli-tools/
- https://gist.github.com/ameiji/bfb738ec5edd6ab701b2095ed05e138e

5.5 ILO3 Notes

The ILO 3 card on the HP Prolient DL380 allows us complete remote control of the server for this reason the same security precautions which are used on the idrac6 need to be implemented.

Securing the ILO3

The ilo3 is not directly accessible accept through the admin lan firewall. Eventually this will require vpn access however in the mean time it is accessed through port redirection. The ilo3s main access is through https. The port number for this is configurable along with the other ports used. (ssh + 2 ports for console redirection)

[[Image(ILO3Notes:ilo3NetworkPorts.png)]] Unless you are working in a MAAS environment the ipv6 should be disabled and the ipv4 address should be made static. This will require resetting the ILO3 itself. [[Image(ILO3Notes:ILO3ResetILO.png)]]

MANAGE ADMIN ACCOUNTS

Create user and management accounts as soon as possible and demote or remove any existing accounts. [[Image(ILO3Notes:ilo3UserAdmin.png)]] While there you should add your ssh keys for ssh connections. Note that only dsa keys are supported so you my need to create a separate public key.

```
steve:~ don$ ssh-keygen -t dsa
Generating public/private dsa key pair.
...
```

Java Console

The ILO 3 provides a java console similar to the one provided by the Dell idrac. It requires the remote console port (17990) as well as the Virtual Medea Port (17988) to function properly. [[Image(ILO3Notes:HPBootSplash.png)]]

Remote Media

Attaching an iso is straight forward. [[Image(ILO3Notes:ilo3RemovableMedia.png)]] Using the Ubuntu 18.04 Live Server over a DSL connection is pokey and complains a lot but it does not fail. [[Image(ILO3Notes:ilo3NetworkMountsAndLag.png)]]

Enabling bios and console accèss via ssh.

Once you have administrative access to the ILO3 and you have an os install you can do everything vial ssh. Much like the idrac you need access to the f9 key. [[Image(wiki:Idrac6:fnkeys.png)]] * Enter bios * Select Serial settings. * set console redirection to com2 _ you will have to do this in the advanced settings as well _ [[Image(ILO3Notes:ILO Bios Virtual Serial Port.jpg)]]

5.5.1 Connecting to the console

Once the bios is set up you can ssh to the console using your iso credentials and ssh key.

```
steve:~ don$ ssh -p22222 feurig@vpn.suspectdevices.com
User:feurig logged-in to kb2018.suspectdevices.com(192.168.31.119 / FE80::9E8E:99FF:FE0C:BAD8)
iLO 3 Advanced for BladeSystem 1.88 at  Jul 13 2016
Server Name: kb2018
Server Power: On

hpiLO-> help

status=0
status_tag=COMMAND COMPLETED
Sat Sep 22 20:20:42 2018
...
DMTF SMASH CLP Commands:
...
HP CLI Commands:

POWER    : Control server power.
UID      : Control Unit-ID light.
NMI      : Generate an NMI.
VM       : Virtual media commands.
LANGUAGE : Command to set or get default language
```

```
VSP      : Invoke virtual serial port.
TEXTCONS : Invoke Remote Text Console.
```

Then you can connect to the console

```
hpiLO-> vsp

Virtual Serial Port Active: COM2

Starting virtual serial port.
Press 'ESC (' to return to the CLI Session.


Ubuntu 18.04.1 LTS kb2018 ttyS1

kb2018 login:
```

If the session is preoccupied use the following (stop /system1/oemhp_vsp1)

```
steve:~ don$ ssh -p 22222 feurig@vpn.suspectdevices.com
User:feurig logged-in to kb2018.suspectdevices.com(192.168.31.119 / FE80::9E8E:99FF:FE0C:BAD8)
iLO 3 Advanced for BladeSystem 1.88 at  Jul 13 2016
Server Name: kb2018
Server Power: On

hpiLO-> vsp
...
Virtual Serial Port is currently in use by another session.
hpiLO-> stop /system1/oemhp_vsp1
...
hpiLO-> hpiLO-> vsp

Virtual Serial Port Active: COM2
```

5.5.2 fixing grub (identical to the process for idrac 6)

You need set the console to ttyS1 by adding a console=ttyS1,115200n8 to the end of the kernel line

```
root@bs2020:~# nano /boot/grub/menu.list
...
kernel          /boot/vmlinuz-4.4.0-96-generic root=UUID=8cafbdf6-441e-4f76-b89c-017fc22253f9 ro console=hvc0 console=ttyS1,115200n8
```

Add the changes to /etc/default/grub so that it will survive updates to the kernel.

```
root@bs2020:~# nano /etc/default/grub
...
GRUB_TERMINAL='serial console'
GRUB_CMDLINE_LINUX="console=hvc0 console=ttyS1,115200n8"
GRUB_SERIAL_COMMAND="serial --speed=115200 --unit=1 --word=8 --parity=no --stop=1"
root@bs2020:~# update-grub
```

Reboot the server and attach to the console. [[Image(ILO3Notes:ILo3SerialBootScreen.png)]]
[[Image(ILO3Notes:ILO3SerialConsoleBootFinish.png)]]

5.5.3 virtual serial port in action

In order to make the dl380 expose the disks we added required jumping into the raid controllers bios during boot and configuring it. This is documented [[wiki:CaptiveRaidController|here]]

HP Documents

- ILO3 Users Guide (https://support.hpe.com/hpsc/doc/public/display?sp4ts.oid=5294355&docLocale=en_US&docId=emr_na-c02774507)
- ILO3 Scripting Guide (https://support.hpe.com/hpsc/doc/public/display?sp4ts.oid=5294355&docLocale=en_US&docId=emr_na-c02774508)
- ILO3 Serial Port Guide (https://support.hpe.com/hpsc/doc/public/display?sp4ts.oid=5294355&docLocale=en_US&docId=emr_na-c00263709)
- ILO3 Security Brief (https://support.hpe.com/hpsc/doc/public/display?sp4ts.oid=5294355&docLocale=en_US&docId=emr_na-a00026171en_us)

Link Dump

- using the VSP features of the ilo3 to configure the raid controller (http://trac.suspectdevices.com/trac/wiki/CaptiveRaidController)
- Using IPMI to configure ILO card (http://dev-random.net/configuring-hp-ilo-through-linux-automatically/)
- https://sysadmin.compxtreme.ro/access-hps-ilo-remote-console-via-ssh/
- bonus link on how to kill outstanding connections (https://stivesso.blogspot.com/2012/02/hp-ilolinux-output-to-vsp-for-linux.html)

5.6 Hot swapping disks on live zfs pools

HOLY FUCKING AWESOME!!!! Watch while I add a fresh disk as a mirror, resliver the pool and remove and repartition the original disk while the container using the pool is still running!!!

YOU ARE HERE - making this into a structured document

```
root@bs2020:~# zpool status
    pool: lxd4dev
    state: ONLINE
    scan: scrub repaired 0 in 0h8m with 0 errors on Sun Aug 12 00:32:48 2018
config:

    NAME        STATE     READ WRITE CKSUM
    lxd4dev     ONLINE       0     0     0
        sdd1      ONLINE       0     0     0
        sdf       ONLINE       0     0     0
        sde       ONLINE       0     0     0

errors: No known data errors

    pool: lxd4infra
    state: ONLINE
    scan: scrub repaired 0 in 0h2m with 0 errors on Sun Aug 12 00:27:02 2018
config:

    NAME        STATE     READ WRITE CKSUM
    lxd4infra   ONLINE       0     0     0
        sda1      ONLINE       0     0     0

errors: No known data errors
root@bs2020:~# zpool add -n lxd4infra mirror sdb
invalid vdev specification: mirror requires at least 2 devices
root@bs2020:~# zpool add -n lxd4infra mirror sda1 sdb
invalid vdev specification
use '-f' to override the following errors:
/dev/sda1 is part of active pool 'lxd4infra'
/dev/sdb does not contain an EFI label but it may contain partition
information in the MBR.
root@bs2020:~# mklabel GPT /dev/sdb
bash: mklabel: command not found
root@bs2020:~# parted /dev/sdb
bash: parted: command not found
root@bs2020:~# gparted /dev/sdb
bash: gparted: command not found
root@bs2020:~# zpool add -nf lxd4infra mirror sda1 sdb
invalid vdev specification
the following errors must be manually repaired:
/dev/sda1 is part of active pool 'lxd4infra'
root@bs2020:~# zpool add -nf lxd4infra mirror sdb sda1
invalid vdev specification
the following errors must be manually repaired:
/dev/sda1 is part of active pool 'lxd4infra'
root@bs2020:~# zpool add -nf lxd4infra mirror sdb
invalid vdev specification: mirror requires at least 2 devices
root@bs2020:~# zpool add -nf lxd4infra sda1 mirror sdb
invalid vdev specification: mirror requires at least 2 devices
root@bs2020:~# zpool attach  -n  sda1 sdb
invalid option 'n'
usage:
    attach [-f] [-o property=value] <pool> <device> <new-device>
root@bs2020:~# zpool attach   sda1 sdb
missing <new_device> specification
usage:
    attach [-f] [-o property=value] <pool> <device> <new-device>
root@bs2020:~# zpool attach lxd4infra   sda1 sdb
invalid vdev specification
use '-f' to override the following errors:
/dev/sdb does not contain an EFI label but it may contain partition
information in the MBR.
root@bs2020:~# gparted
bash: gparted: command not found
root@bs2020:~# parted
bash: parted: command not found
root@bs2020:~# apt-get install parted
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
    libparted2
Suggested packages:
    libparted-dev libparted-i18n parted-doc
The following NEW packages will be installed:
    libparted2 parted
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 158 kB of archives.
After this operation, 520 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

```
Get:1 http://us.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libparted2 amd64 3.2-15ubuntu0.1 [115 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu xenial-updates/main amd64 parted amd64 3.2-15ubuntu0.1 [42.4 kB]
Fetched 158 kB in 0s (277 kB/s)
Selecting previously unselected package libparted2:amd64.
(Reading database ... 32152 files and directories currently installed.)
Preparing to unpack .../libparted2_3.2-15ubuntu0.1_amd64.deb ...
Unpacking libparted2:amd64 (3.2-15ubuntu0.1) ...
Selecting previously unselected package parted.
Preparing to unpack .../parted_3.2-15ubuntu0.1_amd64.deb ...
Unpacking parted (3.2-15ubuntu0.1) ...
Processing triggers for libc-bin (2.23-0ubuntu10) ...
Processing triggers for man-db (2.7.5-1) ...
Setting up libparted2:amd64 (3.2-15ubuntu0.1) ...
Setting up parted (3.2-15ubuntu0.1) ...
Processing triggers for libc-bin (2.23-0ubuntu10) ...
root@bs2020:~# parted /dev/sdb
GNU Parted 3.2
Using /dev/sdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) mklabel GPT
(parted) w
    align-check TYPE N                        check partition N for TYPE(min|opt) alignment
    help [COMMAND]                            print general help, or help on COMMAND
    mklabel,mktable LABEL-TYPE                create a new disklabel (partition table)
    mkpart PART-TYPE [FS-TYPE] START END      make a partition
    name NUMBER NAME                          name partition NUMBER as NAME
    print [devices|free|list,all|NUMBER]      display the partition table, available devices, free space, all found partitions, or a particular
        partition
    quit                                      exit program
    rescue START END                          rescue a lost partition near START and END
    resizepart NUMBER END                     resize partition NUMBER
    rm NUMBER                                 delete partition NUMBER
    select DEVICE                             choose the device to edit
    disk_set FLAG STATE                       change the FLAG on selected device
    disk_toggle [FLAG]                        toggle the state of FLAG on selected device
    set NUMBER FLAG STATE                     change the FLAG on partition NUMBER
    toggle [NUMBER [FLAG]]                    toggle the state of FLAG on partition NUMBER
    unit UNIT                                 set the default unit to UNIT
    version                                   display the version number and copyright information of GNU Parted
(parted) q
Information: You may need to update /etc/fstab.

root@bs2020:~# zpool attach lxd4infra   sda1 sdb
root@bs2020:~# zpool status
    pool: lxd4dev
    state: ONLINE
    scan: scrub repaired 0 in 0h8m with 0 errors on Sun Aug 12 00:32:48 2018
config:

    NAME        STATE     READ WRITE CKSUM
    lxd4dev     ONLINE       0     0     0
        sdd1      ONLINE       0     0     0
        sdf       ONLINE       0     0     0
        sde       ONLINE       0     0     0

errors: No known data errors

    pool: lxd4infra
    state: ONLINE
status: One or more devices is currently being resilvered.  The pool will
    continue to function, possibly in a degraded state.
action: Wait for the resilver to complete.
    scan: resilver in progress since Tue Sep  4 09:05:14 2018
    182M scanned out of 5.38G at 10.7M/s, 0h8m to go
    181M resilvered, 3.30% done
config:

    NAME        STATE     READ WRITE CKSUM
    lxd4infra   ONLINE       0     0     0
        mirror-0  ONLINE       0     0     0
        sda1    ONLINE       0     0     0
        sdb     ONLINE       0     0     0  (resilvering)

errors: No known data errors
root@bs2020:~# packet_write_wait: Connection to 198.202.31.242: Broken pipe
steve:~ don$ ssh feurig@bs2020.suspectdevices.com
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-96-generic x86_64)


    * Documentation:  https://help.ubuntu.com
    * Management:     https://landscape.canonical.com
    * Support:        https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.

New release '18.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Tue Sep  4 08:26:28 2018 from 75.164.203.77
feurig@bs2020:~$ sudo bash
[sudo] password for feurig:
root@bs2020:~# packet_write_wait: Connection to 198.202.31.242: Broken pipe
```

```
steve:~ don$ ssh feurig@bs2020.suspectdevices.com
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-96-generic x86_64)

    * Documentation:  https://help.ubuntu.com
    * Management:     https://landscape.canonical.com
    * Support:        https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.

New release '18.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Wed Sep  5 16:10:53 2018 from 75.164.203.77
feurig@bs2020:~$ sudo bash
[sudo] password for feurig:
root@bs2020:~# packet_write_wait: Connection to 198.202.31.242: Broken pipe
steve:~ don$
steve:~ don$ ssh feurig@bs2020.suspectdevices.com
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-96-generic x86_64)

     * Documentation:  https://help.ubuntu.com
     * Management:     https://landscape.canonical.com
     * Support:        https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.

New release '18.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Wed Sep  5 18:56:14 2018 from 75.164.203.77
feurig@bs2020:~$ sudo bash
[sudo] password for feurig:
root@bs2020:~# zpool status
    pool: lxd4dev
    state: ONLINE
    scan: scrub repaired 0 in 0h8m with 0 errors on Sun Aug 12 00:32:48 2018
config:

    NAME        STATE     READ WRITE CKSUM
    lxd4dev     ONLINE       0     0     0
        sdd1      ONLINE       0     0     0
        sdf       ONLINE       0     0     0
        sde       ONLINE       0     0     0

errors: No known data errors

    pool: lxd4infra
    state: ONLINE
    scan: resilvered 5.38G in 0h6m with 0 errors on Tue Sep  4 09:11:31 2018
config:

    NAME        STATE     READ WRITE CKSUM
    lxd4infra   ONLINE       0     0     0
        mirror-0  ONLINE       0     0     0
        sda1    ONLINE       0     0     0
        sdb     ONLINE       0     0     0

errors: No known data errors
root@bs2020:~# zpool detach -n lxd4infra sda1
invalid option 'n'
usage:
    detach <pool> <device>
root@bs2020:~# zpool detach  lxd4infra sda1
root@bs2020:~# zpool status
    pool: lxd4dev
    state: ONLINE
    scan: scrub repaired 0 in 0h8m with 0 errors on Sun Aug 12 00:32:48 2018
config:

    NAME        STATE     READ WRITE CKSUM
    lxd4dev     ONLINE       0     0     0
        sdd1      ONLINE       0     0     0
        sdf       ONLINE       0     0     0
        sde       ONLINE       0     0     0

errors: No known data errors

    pool: lxd4infra
    state: ONLINE
    scan: resilvered 5.38G in 0h6m with 0 errors on Tue Sep  4 09:11:31 2018
config:

    NAME        STATE     READ WRITE CKSUM
    lxd4infra   ONLINE       0     0     0
        sdb       ONLINE       0     0     0

errors: No known data errors
root@bs2020:~# gparted /dev/sda
bash: gparted: command not found
root@bs2020:~# parted /dev/sda
```

```
GNU Parted 3.2
Using /dev/sda
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) mklabel GPT
Warning: The existing disk label on /dev/sda will be destroyed and all data on this disk will be lost. Do you want to continue?
Yes/No? y
(parted) q
Information: You may need to update /etc/fstab.

root@bs2020:~# zpool status
    pool: lxd4dev
    state: ONLINE
    scan: scrub repaired 0 in 0h8m with 0 errors on Sun Aug 12 00:32:48 2018
config:

    NAME        STATE     READ WRITE CKSUM
    lxd4dev     ONLINE       0     0     0
        sdd1      ONLINE       0     0     0
        sdf       ONLINE       0     0     0
        sde       ONLINE       0     0     0

errors: No known data errors

    pool: lxd4infra
    state: ONLINE
    scan: resilvered 5.38G in 0h6m with 0 errors on Tue Sep  4 09:11:31 2018
config:

    NAME        STATE     READ WRITE CKSUM
    lxd4infra   ONLINE       0     0     0
        sdb       ONLINE       0     0     0

errors: No known data errors
root@bs2020:~# zpool attach lxd4infra   sdb sda
root@bs2020:~# zpool status
    pool: lxd4dev
    state: ONLINE
    scan: scrub repaired 0 in 0h8m with 0 errors on Sun Aug 12 00:32:48 2018
config:

    NAME        STATE     READ WRITE CKSUM
    lxd4dev     ONLINE       0     0     0
        sdd1      ONLINE       0     0     0
        sdf       ONLINE       0     0     0
        sde       ONLINE       0     0     0

errors: No known data errors

    pool: lxd4infra
    state: ONLINE
status: One or more devices is currently being resilvered.  The pool will
    continue to function, possibly in a degraded state.
action: Wait for the resilver to complete.
    scan: resilver in progress since Thu Sep  6 09:24:09 2018
    69.8M scanned out of 5.42G at 5.37M/s, 0h17m to go
    67.9M resilvered, 1.26% done
config:

    NAME        STATE     READ WRITE CKSUM
    lxd4infra   ONLINE       0     0     0
        mirror-0  ONLINE       0     0     0
        sdb     ONLINE       0     0     0
        sda     ONLINE       0     0     0  (resilvering)

errors: No known data errors
root@bs2020:~#
```

5.6.1 Hardware

Starting in December the environment will contains a freebsd based router/firwall and a single enterprise class server

- sitka -- a RiverBed Stealhead CX-770
- tk2018 -- a HP ProLiant DL380 (g7) .

5.6.2 Network

The network is divided into 3 segments

- 192.168.31.0/24 a private administrative lan
- 10.0.0.0/24 wireguard lan
- 198.202.31.129/25 A public facing lan.

The host itself does not have any public facing interfaces. It only accessible though the wireguard protected admin lan. The containers, which handle all public facing work do so via an anonymous bridge configuration, allowing them to access the internet directly without allowing external access to the underlying servers.

As we move forward the unfiltered interface used by the public facing containers will eventually be replaced by a filtered interface through the firewall.

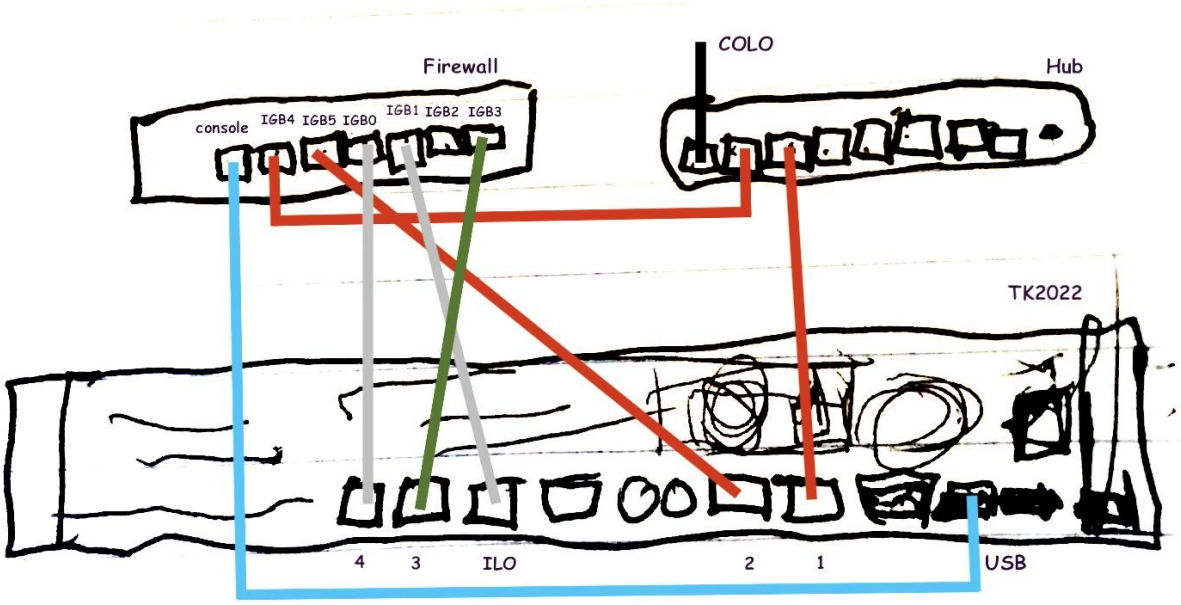
Sitka's Network Config

			sitka ports
port	Interface	IP Address/mask	purpose
igb0	bridge0	192.168.31.159/24	internal / admin lan
igb1	bridge0		
igb2	N/A	N/A	N/A
igb3	igb3	?.?.?.?/??	TBD
igb4	igb4	198.202.31.132/25	
igb5	igb5	0.0.0.0/32	firewalled public interface

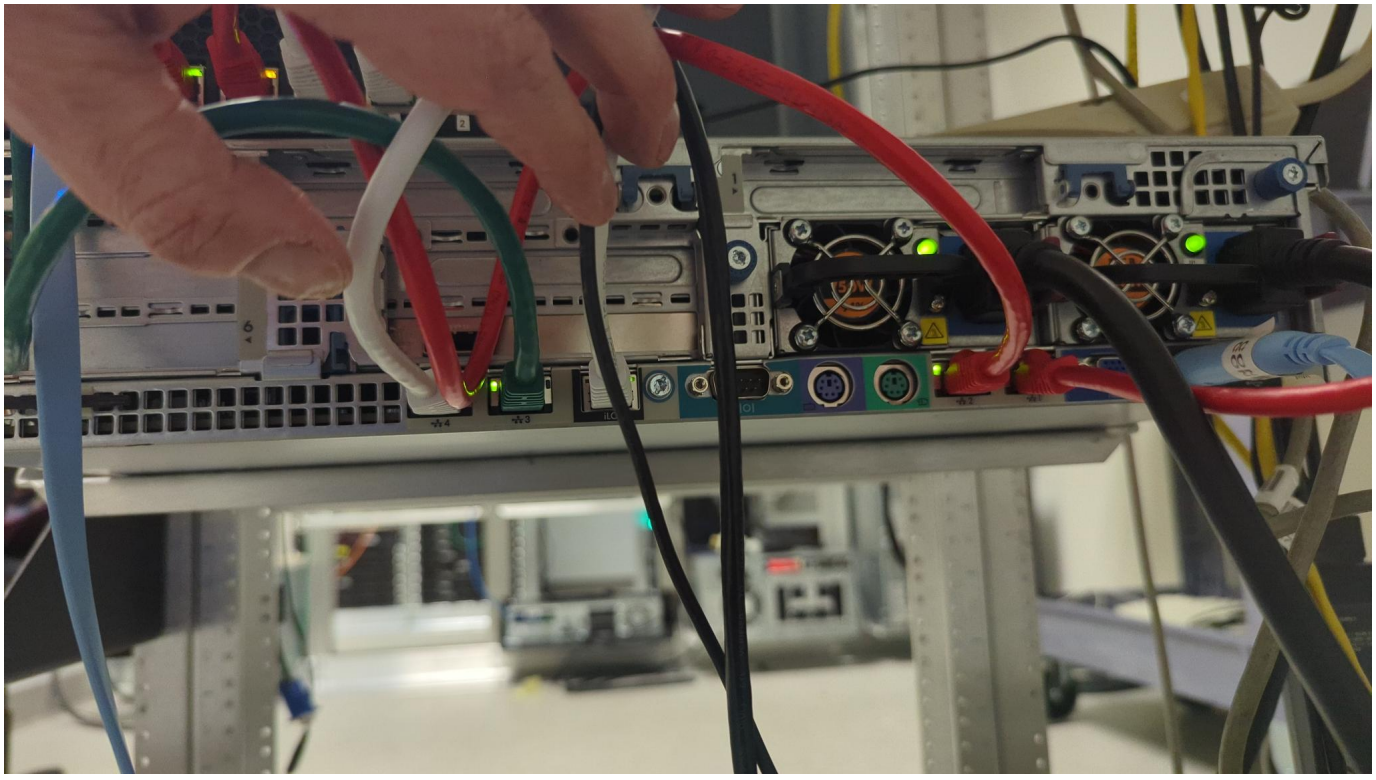
TK2022's Network Config

				tk2022 ports
port	Interface	IP Address/mask	linux device	purpose
1	br0	0.0.0.0/32	enp3s0f0	unfiltered public interface
2	br2	0.0.0.0/32	enp3s0f1	firewalled public interface
3	N/A	?.?.?.?/??	enp4s0f0	TBD
1	br1	192.168.31.159/24	enp4s0f1	internal / admin lan
ilo		192.168.31.119/24		remote console

AS DRAWN



AS DEPLOYED



AS IMPLIMENTED

in /etc/network/interfaces

```
#------------------------------------------------------/etc/network/interfaces
# 2: enp3s0f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq master br0 state UP group default qlen 1000
# 3: enp3s0f1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq master br1 state UP group default qlen 1000
# 4: enp4s0f0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
# 5: enp4s0f1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9000 qdisc mq master br3 state UP group default qlen 1000
# https://ip4calculator.com
```

```
source /etc/network/interfaces.d/*

auto lo
iface lo inet loopback

iface enp3s0f0 inet manual
iface enp3s0f1 inet manual
iface enp4s0f0 inet manual
iface enp4s0f1 inet manual

auto br0
iface br0 inet manual
     bridge_ports enp3s0f0
     bridge_stp off       # disable Spanning Tree Protocol
        bridge_waitport 0    # no delay before a port becomes available
        bridge_fd 0          # no forwarding delay


auto br1
iface br1 inet static
     address 192.168.31.159
     network 192.168.31.0
     netmask 255.255.255.0
     broadcast 192.168.31.255
     bridge_ports enp4s0f1
     bridge_stp off       # disable Spanning Tree Protocol
        bridge_waitport 0    # no delay before a port becomes available
        bridge_fd 0          # no forwarding delay
EOD
```

and in /etc/rc.conf

```
ifconfig_igb4="69.41.138.126 netmask 255.255.255.224"
defaultrouter="69.41.138.97"

#ifconfig_igb0="inet 192.168.31.2 netmask 255.255.255.0"

cloned_interfaces="bridge0"
ifconfig_bridge0="inet 192.168.31.2 netmask 255.255.255.0 addm igb0 addm igb1 up"
ifconfig_igb0="up"
ifconfig_igb1="up"
```

AS REFERENCED

See: https://bitbucket.org/suspectdevicesadmin/ansible/src/master/hosts which is built referencing a google doc with proposed allocations

5.6.3 Server OS, Filesystems and Disk layout

The server runs Debian bookworm along with zabbly supported version of incus. Outside of zfs not much is added to the stock installation. This is intentional. The real work is done by the containers the host os is considered disposable.

Disk Layout

The incus server uses hardware raid 1 for the boot disk. The containers and other data are a able to take advantage of zfs mirroring and caching.

					kb2018 disks
disk	device/pool	bay	type	mount point(s)	purpose/notes
sdb	/dev/sdb	2C:1:3	raid1+0	/, /var/lib/incus	os and incus data
		2C:1:4	raid1+0		
sda	infra, devel	3C:1:7	zfs		incus storage pools
sdg		3C:1:8	zfs mirror		
sdd	tank	3C:1:5	zfs	/tank	space for stuff
sdc		3C:1:6	zfs mirror		

Hardware raid on the DL380

The raid controller on the Dell allows a mixing of hardware raid and direct hot swappable connections. The HP 420i does only hardware raid or direct connections (HBA) but not both. Since we use the hardware raid the remaining disks need to be configured using the ssacli or the raid controllers bios.

See: Dude Where Are My Disks

5.6.4 Containers

Work previously done by standalone servers is now done though incus managed containers. An up to date list of containers is somewhat maintained at https://bitbucket.org/suspectdevicesadmin/ansible/src/master/hosts

5.6.5 Ansible

Ansible is used to make most tasks reasonable including. * creating containers * updating admin passwords and ssh keys.

5.7 Tasks: Accessing Hosts

tk2022 ssh access

The host machines for the containers can be accessed through the admin lan. This is done via wirguard on either sitka or virgil

note: as of a few updates ago you have to tell apples ssh client to use ssh-dss as below

YOU ARE HERE update the ilo settings so they report the right server.

```
steve:~ don$ ssh -p22 -oHostKeyAlgorithms=+ssh-dss feurig@tinas-ilo.suspectdevices.com
feurig@192.168.31.119's password:
User:feurig logged-in to kb2018.suspectdevices.com(192.168.31.119 / FE80::9E8E:99FF:FE0C:BAD8)
iLO 3 Advanced for BladeSystem 1.88 at  Jul 13 2016
Server Name: kb2018
Server Power: On

</>hpiLO-> vsp

Virtual Serial Port Active: COM2

Starting virtual serial port.
Press 'ESC (' to return to the CLI Session.

tk2022 login: <ESC> (
</>hpiLO-> exit

_ if the serial port is still in use do the following _

Virtual Serial Port is currently in use by another session.
</>hpiLO-> stop /system1/oemhp_vsp1
```

See: ilo 3 notes page

ssh access to containers

The susdev profile adds ssh keys and sudo passwords for admin users allowing direct ssh access to the container.

```
steve:~ don$ ssh feurig@ian.suspectdevices.com
...
feurig@ian:~$
```

The containers can be accessed directly from the incus host as root

```
root@bs2020:~# incus exec harvey bash
root@harvey:~# apt-get update&&apt-get -y dist-upgrade&& apt-get -y autoremove
```

5.7.1 Updating dns

Dns is provided by bind , The zone files have been consolidated into a single directory under /etc/bind/zones on naomi (dns.suspectdevices.com).

```
root@naomi:/etc/bind/zones# nano suspectdevices.hosts
...
@               IN      SOA  dns1.digithink.com. don.digithink.com (
                2018080300 10800 3600 3600000 86400 )
;               ^^ update ^^
; .... make some changes ....
morgan          IN      A       198.202.31.224
git             IN      CNAME   morgan
...
root@naomi:/etc/bind/zones# service bind9 reload
root@naomi:/etc/bind/zones# tail /var/log/messages
...
Sep  3 08:10:04 naomi named[178]: zone suspectdevices.com/IN: loaded serial 2018080300
Sep  3 08:10:04 naomi named[178]: zone suspectdevices.com/IN: sending notifies (serial 2018080300)
Sep  3 08:10:04 naomi named[178]: client 198.202.31.132#56120 (suspectdevices.com): transfer of 'suspectdevices.com/IN': AXFR-style IXFR started (serial
2018080300)
Sep  3 08:10:04 naomi named[178]: client 198.202.31.132#56120 (suspectdevices.com): transfer of 'suspectdevices.com/IN': AXFR-style IXFR ended
Sep  3 08:10:04 naomi named[178]: client 198.202.31.132#47381: received notify for zone 'suspectdevices.com'
```

5.7.2 Updating Hosts / Containers

When updates are available Apticron sends us an email. We prefer this to autoupdating our hosts as it helps us maintain awareness of what issues are being addressed and does not stop working when there are issues. All running containers can be updated using the following update script.

```
nano /usr/local/bin/update.sh
#!/bin/bash
# update.sh for debian/ubuntu/centos/suse  (copyleft) don@suspecdevices.com
echo -------------------- begin updating `uname -n` ----------------------
if [ -x "$(command -v apt-get)" ]; then
   apt-get update
   apt-get -y dist-upgrade
   apt-get -y autoremove
fi
if  [ -x "$(command -v yum)" ]; then
   echo yum upgrade.
   yum -y upgrade
fi
if  [ -x "$(command -v zypper)" ]; then
   echo zypper dist-upgrade.
   zypper -y dist-upgrade
fi
echo ========================== done ==============================
^X
chmod +x /usr/local/bin/update.sh
```

pushing the update script to containers.

```
incus file push /usr/local/bin/update.sh virgil/usr/local/bin/
incus exec virgil chmod +x /usr/local/bin/update.sh
```

you can run this against all running containers as follows.

```
for c in `incus list -cn -f compact|grep -v NAME`; do incus exec $c update.sh; done ; update.sh
```

This could also be used as an ansible ad hoc command.

```
ansible pets -m raw -a "update.sh"
```

https://bitbucket.org/suspectdevicesadmin/ansible/src/master/files/update.sh

5.7.3 Creating containers

```
cd /etc/ansible
nano hosts
... add new host ...
ansible-playbook playbooks/create-lxd-containers.yml
```

https://bitbucket.org/suspectdevicesadmin/ansible/src/master/roles/create_lxd_containers/tasks/main.yml .....YOU ARE HERE..... documenting the ansible script to create containers

5.7.4 Backing Up Containers

YOU ARE HERE RETHINKING THIS

5.8 links.... (tbd)

5.9 Scrapbook

Migrate Users UID/GID

YOU ARE HERE (Consolidate this into a scraps)

```
chown --from=1000:1000 999:999 /. -Rv
```

clone drives and change uuids

```
screen dd if=/dev/sda of=/dev/sdj bs=1M status=progress
apt install uuid-runtime
printf '%s\n' p x i $(uuidgen) r w | sudo fdisk /dev/sdj
e2fsck -f /dev/sdj2
tune2fs -U $(uuidgen) /dev/sdj2
blkid
```

5.10 TaskAddLxdContinerWithAnsible

5.10.1 New Container Using Ansible

YOU ARE HERE: Evaluating whether or not this is still a good way to create containers.

With Ansible added to kb2018 we expand on the profiles we use to create users and create a sane environment. There are two steps required to create a container on kb2018.

- Add the name, ip_address, and purpose to the inventory file /etc/ansible/hosts.

```
...
redshirt ip_address=198.202.31.200 purpose="Disposable Ubuntu"
...

2. Run the ansible playbook _/etc/ansible/playbooks/create-lxd-containers.yml_

root@kb2018:/etc/ansible# ansible-playbook /etc/ansible/playbooks/create-lxd-containers.yml

PLAY [localhost]
*****************************************************************************************************************************************************************************
```

If you want something other than ubuntu-lts you can: * set the image_alias. these are images that we know work in our environment

```
root@kb2018:/etc/ansible# lxc image list
+------------+--------------+--------+----------------------------------------------+--------+----------+------------------------------+
|   ALIAS    | FINGERPRINT  | PUBLIC |                 DESCRIPTION                  |  ARCH  |   SIZE   |          UPLOAD DATE         |
+------------+--------------+--------+----------------------------------------------+--------+----------+------------------------------+
| centos/7c  | 700c86f31546 | no     | Centos 7 (20190109_02:16) plus cloud         | x86_64 | 172.49MB | Mar 21, 2019 at 1:54am (UTC) |
+------------+--------------+--------+----------------------------------------------+--------+----------+------------------------------+
| debian/9c  | 38d17964647d | no     | Debian stretch (20190108_05:24) plus cloud   | x86_64 | 227.58MB | Mar 19, 2019 at 5:57am (UTC) |
+------------+--------------+--------+----------------------------------------------+--------+----------+------------------------------+
| ubuntu-lts | c395a7105278 | no     | ubuntu 18.04 LTS amd64 (release) (20180911) | x86_64 | 173.98MB | Sep 29, 2018 at 11:50pm (UTC) |
+------------+--------------+--------+----------------------------------------------+--------+----------+------------------------------+
root@kb2018:~# cd /etc/ansible/
root@kb2018:/etc/ansible# ls
ansible.cfg  files  hosts  host_vars  playbooks  README.md  roles
oot@kb2018:/etc/ansible# lxc delete redshirt --force
root@kb2018:/etc/ansible# nano hosts
...
redshirt ip_address=198.202.31.200 purpose="Disposable Debian" image_alias="debian/9c"
...

And (re)run the playbook.

root@kb2018:/etc/ansible# ansible-playbook /etc/ansible/playbooks/create-lxd-containers.yml
```

You can also create infrastructure servers by setting net_and_disk_profile to "infra".

The ansible playbook and host file are maintained in a private bitbucket repository. If you add roles or create a host that you want to keep please update the repository. Ignore the errors, I will reconfigure a user for kb2018 when bitbucket really stops supporting the organization account

```
feurig@kb2018:~$ sudo bash
[sudo] password for feurig:
root@kb2018:~# cd /etc/ansible/hosts
root@kb2018:/etc/ansible# nano hosts
...
morgan   ip_address=198.202.31.224 purpose="Infrastucture Test Machine" net_and_disk_profile="infra"
...
root@kb2018:/etc/ansible# ansible-playbook /etc/ansible/playbooks/create-lxd-containers.yml
PLAY RECAP
*****************************************************************************************************************************************************************************
localhost                  : ok=4    changed=1    unreachable=0    failed=0

root@kb2018:/etc/ansible# git commit -a -m "Recreate Morgan as Infrastructure Test Server"
[master 10d4ce0] Recreate Morgan as Infrastructure Test Server
    Committer: Root at KB2018 <root@kb2018.suspectdevices.com>
...
    1 file changed, 1 insertion(+), 1 deletion(-)
root@kb2018:/etc/ansible# git push
Counting objects: 3, done.
Delta compression using up to 16 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 378 bytes | 378.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0)
remote:
remote: Warning!
```

```
remote: You are currently connecting with your team account.
remote: This is no longer supported, so please connect using your user account.
remote:
To bitbucket.org:suspectdevicesadmin/ansible.git
    d7f5f12..10d4ce0  master -> master
root@kb2018:/etc/ansible#
```

5.11 Ubuntu LTS Email Server Setup

This document assumes that you have set up a debian 9 or ubuntu LTS server(/container) set up and that postfix/email has been set up using tasksel.

5.11.1 Dovecot (imap server) and Postfix (mail server)

configure dovecot to use self signed ssl cert created by postfix.

```
root@naomi:/etc/postfix# cd ../dovecot/conf.d/
root@naomi:/etc/dovecot/conf.d# nano 10-ssl.conf
##
## SSL settings
##

# SSL/TLS support: yes, no, required. <doc/wiki/SSL.txt>
ssl = yes

# PEM encoded X.509 SSL/TLS certificate and private key. They're opened before
# dropping root privileges, so keep the key file unreadable by anyone but
# root. Included doc/mkcert.sh can be used to easily generate self-signed
# certificate, just make sure to update the domains in dovecot-openssl.cnf
ssl_cert = </etc/ssl/certs/ssl-cert-snakeoil.pem
ssl_key = </etc/ssl/private/ssl-cert-snakeoil.key
#ssl_cert = </etc/dovecot/dovecot.pem
#ssl_key = </etc/dovecot/private/dovecot.pem
```

Also set mailbox format to Maildir or all of your legacy data will be hosed.

```
root@naomi:/etc/dovecot/conf.d# nano 10-mail.conf
  mail_location = maildir:~/Maildir
...
```

Notice issues with sending mail using ssl/tls

```
don@bob2:~$ openssl s_client -connect mail.suspectdevices.com:465 -starttls smtp
connect: Connection refused
connect:errno=111
```

Add ssl/tls to postfix for outgoing mail

```
root@naomi:/etc/postfix# nano master.cf
...
# ==========================================================================
# service type  private unpriv  chroot  wakeup  maxproc command + args
#               (yes)   (yes)   (no)    (never) (100)
# ==========================================================================
smtp      inet  n       -       y       -       -       smtpd
#smtp      inet  n       -       y       -       1       postscreen
#smtpd     pass  -       -       y       -       -       smtpd
#dnsblog   unix  -       -       y       -       0       dnsblog
#tlsproxy  unix  -       -       y       -       0       tlsproxy
submission inet n       -       y       -       -       smtpd
  -o syslog_name=postfix/submission
  -o smtpd_tls_security_level=encrypt
  -o smtpd_sasl_auth_enable=yes
#  -o smtpd_reject_unlisted_recipient=no
#  -o smtpd_client_restrictions=$mua_client_restrictions
#  -o smtpd_helo_restrictions=$mua_helo_restrictions
#  -o smtpd_sender_restrictions=$mua_sender_restrictions
#  -o smtpd_recipient_restrictions=
  -o smtpd_relay_restrictions=permit_sasl_authenticated,reject
  -o milter_macro_daemon_name=ORIGINATING
root@naomi:/etc/postfix# service postfix check
postfix/postfix-script: warning: group or other writable: /usr/lib/postfix/./sbin/lmtp
postfix/postfix-script: warning: group or other writable: /usr/lib/postfix/./libpostfix-tls.so.1
postfix/postfix-script: warning: group or other writable: /usr/lib/postfix/./libpostfix-global.so.1
postfix/postfix-script: warning: group or other writable: /usr/lib/postfix/./libpostfix-master.so.1
postfix/postfix-script: warning: group or other writable: /usr/lib/postfix/./libpostfix-dns.so.1
postfix/postfix-script: warning: group or other writable: /usr/lib/postfix/./libpostfix-util.so.1
postfix/postfix-script: warning: group or other writable: /usr/lib/postfix/sbin/./lmtp
root@naomi:/etc/postfix# service postfix reload
```

Link authentication to dovecot and enable auth server in dovecot. '' apparently this can be avoided by installing a single package buried in ubuntu's documentation (g: Mail-Stack Delivery).

```
root@naomi:/etc/postfix# nano /etc/dovecot/conf.d/10-master.conf
...
  #Postfix smtp-auth
  unix_listener /var/spool/postfix/private/auth {
    mode = 0666
  }

  # Auth process is run as this user.
  #user = $default_internal_user
}

service auth-worker {
  # Auth worker process is run as root by default, so that it can access
  # /etc/shadow. If this isn't necessary, the user should be changed to
  # $default_internal_user.
  user = root
}
...

root@naomi:/etc/postfix# nano main.cf
# TLS parameters
smtpd_tls_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
smtpd_use_tls=yes
smtpd_tls_auth_only = yes
smtpd_sasl_type = dovecot
smtpd_sasl_path = private/auth
smtpd_sasl_auth_enable = yes
smtpd_recipient_restrictions = permit_sasl_authenticated permit_mynetworks reject_unauth_destination
```

Follow up on above errors

NOTE: the above errors are related to symlinks and not the files. Both debian and canonical aren't concerned about it and may or may not fix it at some point. https://bugs.launchpad.net/ubuntu/+source/postfix/+bug/1728723

eliminate pop3 as it isn't needed

```
mv /usr/share/dovecot/protocols.d/pop3d.protocol /usr/share/dovecot/pop3d.protocol.disabled
service dovecot reload
netstat -ta
```

5.11.2 SPF and openDKIM

Gmail currently requires that any email you send that isn't controlled by them use both SPF and DKIM.

What the hell is it?

According to linuxbabe https://www.linuxbabe.com/mail-server/setting-up-dkim-and-spf

SPF and DKIM are two types of TXT records in DNS that can help prevent email spoofing and ensure legitimate emails are delivered into the recipient’s inbox instead of spam folder. If your domain is abused by email spoofing, then your emails are likely to landed in recipient’s spam folder if they didn’t add you in address book.

SPF (Sender Policy Framework) record specifies which hosts or IP addresses are allowed to send emails on behalf of a domain. You should allow only your own email server or your ISP’s server to send emails for your domain.

_DKIM (DomainKeys Identified Mail) uses a private key to add a signature to emails sent from your domain. Receiving SMTP servers verify the signature by using the corresponding public key, which is published in your DNS manager. _

SPF

We only want to send email through a single server which is accomplished with the following record. Which needs to be added for each domain using the email server.

```
root@naomi:~# nano /etc/bind/zones/fromhell.hosts
... add the following ...
@ TXT "v=spf1 ip4:198.202.31.141 -all"
```

openDKIM

GOTCHAS

- convoluted and complex configuration involving 3 major services (dns,postfix,opendkim).
- postfix is chrooted and milter version is currently 6
- sample output from current opendkim-tools is wrong and requires manual correction.
- Relaying requires masquerading.

INSTALLATION

Install opendkim and edit configuration file

```
root@naomi:~# apt-get install opendkim opendkim-tools
root@naomi:~# nano /etc/opendkim.conf
... add/correct the following ...
Socket          local:/var/spool/postfix/var/run/opendkim/opendkim.sock
PidFile              /var/run/opendkim/opendkim.pid
Syslog          yes
UMask           002
UserID          opendkim
KeyTable            refile:/etc/opendkim/key.table
SigningTable        refile:/etc/opendkim/signing.table
ExternalIgnoreList  refile:/etc/opendkim/trusted.hosts
InternalHosts       refile:/etc/opendkim/trusted.hosts
```

For each domain being handled create a signing key and add to dns zone files.

```
root@naomi:~# cd /etc/opendkim/keys/
root@naomi:/etc/opendkim/keys# opendkim-genkey -b 2048 -h rsa-sha256 -r -s 201807 -d suspectdevices.com -v
root@naomi:/etc/opendkim/keys# mv 201807.private suspectdevices.private
root@naomi:/etc/opendkim/keys# cat 201807.txt >>/etc/bind/zones/suspectdevices.hosts
```

Fix the error in dns entry and increment the zones serial number

```
root@naomi:/etc/opendkim/keys# nano /etc/bind/zones/suspectdevices.hosts
@               IN      SOA  dns1.digithink.com. don.digithink.com (
                2018072200 10800 3600 3600000 86400 )
...change.this. YYYYMMDDxx  ....
...
... and change h=rsa-sha256 to h=sha256 ...      ...as below...
201807._domainkey       IN      TXT     ( "v=DKIM1; h=sha256; k=rsa; s=email; "
          "p=MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA6ymvRll+pEDThA6fMersYbr6dB5HKIFl4SMSF3ORxkFmrYC//wm6/vrqWNft3AWy4zC7AQNiKyQGg7$
          "BUpxeL2bSGUhMrcZ+OheWWzw7aF746IOYO0IR4oMTFNP9a6hrmwBrLmnA8ploFYUWCa2ETq/VYP6i14LU7P/yi8JhDMu4ZVI6ytlynBcLU42orcNWjWNLHqy/F3L$
```

Reload bind and check key

```
root@naomi:/etc/opendkim/keys# service bind9 reload
root@naomi:/etc/opendkim/keys# service bind9 status
● bind9.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/bind9.service; enabled; vendor preset: enabled)
....
Jul 25 22:35:15 naomi named[28512]: reloading zones succeeded
....
root@naomi:/etc/opendkim/keys# opendkim-testkey -d suspectdevices.com  -s 201807 -vvv
opendkim-testkey: using default configfile /etc/opendkim.conf
opendkim-testkey: checking key '201807._domainkey.suspectdevices.com'
opendkim-testkey: key not secure .... ignore this ....
opendkim-testkey: key OK
```

Add entries to key.table signing.table and trusted hosts.

```
    root@naomi:/etc/opendkim# nano key.table
    fromhell     fromhell.com:201807:/etc/opendkim/keys/fromhell.private
    suspectdevices suspectdevices.com:201807:/etc/opendkim/keys/suspectdevices.private
    root@naomi:/etc/opendkim# nano signing.table
 *@fromhell.com fromhell
 *@suspectdevices.com suspectdevices
    root@naomi:/etc/opendkim# nano trusted.hosts
    127.0.0.1
    ::1
    198.202.31.221
    198.202.31.242
    localhost
 *.fromhell.com
 *.suspectdevices.com
```

Configure socket file to communicate with postfix and add postfix to opendkim group.

```
root@naomi:~# mkdir -p /var/spool/postfix/var/run/opendkim
root@naomi:~# chown -R opendkim:opendkim /var/spool/postfix/var/run/opendkim
root@naomi:~# touch /var/spool/postfix/var/run/opendkim/opendkim.sock
root@naomi:~# chmod 775 /var/spool/postfix/var/run/opendkim/opendkim.sock
root@naomi:~# usermod -a -G opendkim postfix
root@naomi:~# nano /etc/default/opendkim
...
DAEMON_OPTS="-vvvv"
SOCKET="local:/var/spool/postfix/var/run/opendkim/opendkim.sock"
RUNDIR=/var/spool/postfix/var/run/opendkim
USER=opendkim
GROUP=opendkim
PIDFILE=$RUNDIR/$NAME.pid
EXTRAAFTER=
...
```

Add filter to postfix and restart both services.

```
root@naomi:~# nano /etc/postfix/main.cf
...
milter_protocol = 6
milter_default_action = accept
smtpd_milters = unix:/var/run/opendkim/opendkim.sock
non_smtpd_milters = unix:/var/run/opendkim/opendkim.sock
...
root@naomi:~# service opendkim reload
root@naomi:~# service postfix reload
```

Send test mail

```
root@naomi:~# echo "dkim test" |mail -testopendkim  check-auth@verifier.port25.com
```

ADDING SIGNATURES TO RELAYED HOSTS

To relay mail from other hosts on the local networks requires the following additions to postfix's main.cf

```
root@naomi:~# nano /etc/postfix/main.cf
...
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128, 198.202.31.128/25
...
masquerade_domains = suspectdevices.com, fromhell.com
```

OPENDKIM/SPF LINKS

- https://www.cioby.ro/2013/11/14/configuring-opendkim-to-sign-postfix-emails/
- https://linuxaria.com/howto/using-opendkim-to-sign-postfix-mails-on-debian
- http://www.openspf.org/SPF_Record_Syntax
- https://blog.whabash.com/posts/send-outbound-email-postfix-dkim-spf-ubuntu-16-04
- https://www.linode.com/docs/email/postfix/configure-spf-and-dkim-in-postfix-on-debian-8/
- https://www.linuxbabe.com/mail-server/setting-up-dkim-and-spf
- https://tools.ietf.org/html/rfc6376
- https://tweenpath.net/opendkim-postfix-smtp-relay-server-on-debian-7/
- https://qureshi.me/how-to-setup-postfixdkimspfdmarc-on-ubuntu-plesk-onyx/

5.11.3 Configure root/notification mail from other systems (esp bs2020)

Systems need to be able send email to notify us of issues such as security updates (apticron) etc. In order for email to be signed by opendkim and validated by spf the email needs to strip the hostname from mail sent from it before being relayed through the mail server.

```
root@bs2020:~# apt-get install mailutils apticron
... select satellite server when asked ...
root@bs2020:~# nano /etc/postfix/main.cf
... add the following ...
relayhost = naomi.suspectdevices.com
compatibility_level=2
masquerade_domains = suspectdevices.com
```

Since all systems will be striped of their machine names insure the full name of common accounts is made to be uniq

```
root@bs2020:~# chfn -f "Root at BS2020"
```

- http://www.postfix.org/STANDARD_CONFIGURATION_README.html
- https://www.tecmint.com/setup-postfix-mail-server-smtp-using-null-client-on-centos/ Todo:
- I think postfix is a little heavy handed to run a null client. Investigate simpler secure solution.
- add amivis,and other filters linked in at https://help.ubuntu.com/community/MailServer
- make procmail do some work since its enabled by default
- make damned sure that it wont accept mail from the entire c-block

5.12 SSACLI - hp's utilities for configuring its hardware raid controller

5.12.1 Install ssacli

```
apt install gpg
apt install curl
curl -x http://192.168.31.2:3128/ -fsSL https://downloads.linux.hpe.com/SDR/hpPublicKey2048.pub | gpg --dearmor -o /etc/apt/trusted.gpg.d/hpPublicKey2048.gpg
curl -x http://192.168.31.2:3128/ -fsSL https://downloads.linux.hpe.com/SDR/hpePublicKey2048_key1.pub | gpg --dearmor -o /etc/apt/trusted.gpg.d/
hpePublicKey2048_key1.gpg
curl -x http://192.168.31.2:3128/-fsSL https://downloads.linux.hpe.com/SDR/hpPublicKey2048_key1.pub | gpg --dearmor -o /etc/apt/trusted.gpg.d/
hpPublicKey2048_key1.gpg
apt update
apt install ssacli
```

5.12.2 Using ssacli

Using ssacli to set the primary boot disk.

```
=> set target ctrl slot=0

   "controller slot=0"

=> show config detail
... find the drive that coresponds to what you want

=> ld 10 modify bootvolume=primary
=>
```

To recover if the selected drive does not boot log into the ilo.

```
</>hpiLO-> power reset

status=0
status_tag=COMMAND COMPLETED
Thu Nov 28 17:04:30 2024

Server resetting .......

</>hpiLO-> vsp
```

Wait for the eternity it takes to run through the hardware and memory on the hp. Once it gets to the actual bios change to the text console.

```
<ESC>(
</>hpiLO-> textcons
```

The text console is nice because (inspite of char set differences) the function keys work. Press f8 when you get to the raid controller (after it searches for the disks)

Text console will not work until it actually gets to the bios and you can switch back to the VSP by escaping out

```
<ESC>(
</>hpiLO-> vsp

Virtual Serial Port Active: COM2
```

5.12.3 References

- https://www.n0tes.fr/2024/03/04/CLI-HPE-ssacli-and-hpssacli-tools/