

Automating My FreeBSD Lab: From Setup to Daily Use with Ansible & Salt



ANSIBLE



What we will cover

Quite a lot to cover!

- Why automate?
- FreeBSD lab & laptop setup overview
- Bootstrapping a new system
- Automating system installation & package setup
- Adding a laptop with just SSH access
- Managing machine states with Salt

Lab host setup

```
# pkg install -y sudo git-lite py311-ansible
```

```
# visudo (give %wheel group privileges)
```

```
— $ git clone
```

```
https://github.com/BSD-Programming-Workshop/zfs-jail-host-lab.git
```

Why automate?

The FreeBSD Jails Lab

—



96 GB RAM
12-Core XEON
3 NVME SSD



Home0





**ZFS's Atomic I/O
and PostgreSQL**

**Virtual Lab—
BSD Programming
Workshop**

An Introduction to ZFS



FreeBSD

Journal Article about the Jails Lab

Jan/Feb 2023

VM Network Setup

VM1:
The FreeBSD Jail Host

/etc/rc.conf



Network

Adapter 1: Intel PRO/1000 MT Desktop (NAT)
Adapter 2: Intel PRO/1000 MT Desktop (NAT Network, 'HomeNetwork')
Adapter 3: Intel PRO/1000 MT Desktop (NAT)

```
ifconfig_em0_name="lab0"  
ifconfig_lab0="up"  
ifconfig_em1_name="home0"  
ifconfig_home0="up"
```

```
lab0: flags=1008943<UP,BROADCAST,
mtu 1500
    options=8500b9<RXCSUM,VLAN_F
LTER,VLAN_HWTSO,HWSTATS>
    ether 08:00:27:af:dc:33
    media: Ethernet autoselect
    status: active
    nd6 options=29<PERFORMNU
home0: flags=1008943<UP,BROADCAST
0 mtu 1500
    options=8500b9<RXCSUM,VLAN_F
LTER,VLAN_HWTSO,HWSTATS>
    ether 08:00:27:f0:7e:79
    media: Ethernet autoselect
    status: active
    nd6 options=29<PERFORMNU
em2: flags=1008802<BROADCAST,SIM
    options=48505bb<RXCSUM,T
SO4,LRO,VLAN_HWFILTER,VLAN_HWTSO
    ether 08:00:27:40:b6:0d
    media: Ethernet autoselect
    status: active
    nd6 options=29<PERFORMNU
```

```
lab0bridge: flags=1008843<UP,BRO
tu 1500
    options=0
    ether 58:9c:fc:10:be:4b
    id 00:00:00:00:00:00 pri
maxage 20 holdcnt 6 prot
root id 00:00:00:00:00:00
member: e0a_gateway flag
    ifmaxaddr 0 port
member: lab0 flags=143<L
    ifmaxaddr 0 port
groups: bridge
home0bridge: flags=1008843<UP,BRO
mtu 1500
    options=0
    ether 58:9c:fc:10:bc:63
    inet 192.168.66.11 netmask 0
    id 00:00:00:00:00:00 priorit
maxage 20 holdcnt 6 proto rs
root id 00:00:00:00:00:00 pr
member: e2a_gateway flags=14
    ifmaxaddr 0 port 13
member: home0 flags=143<LEAR
    ifmaxaddr 0 port 2 p
groups: bridge
```

```
labnet: flags=8103<UP,BROADCAST,PROMISC,MULTICAST> metric 0 mtu 1500
options=0
ether 00:00:00:00:00:00
groups: vlan
vlan: 0 vlanproto: 0x0000 vlanpcp: 0 parent interface: <none>
```

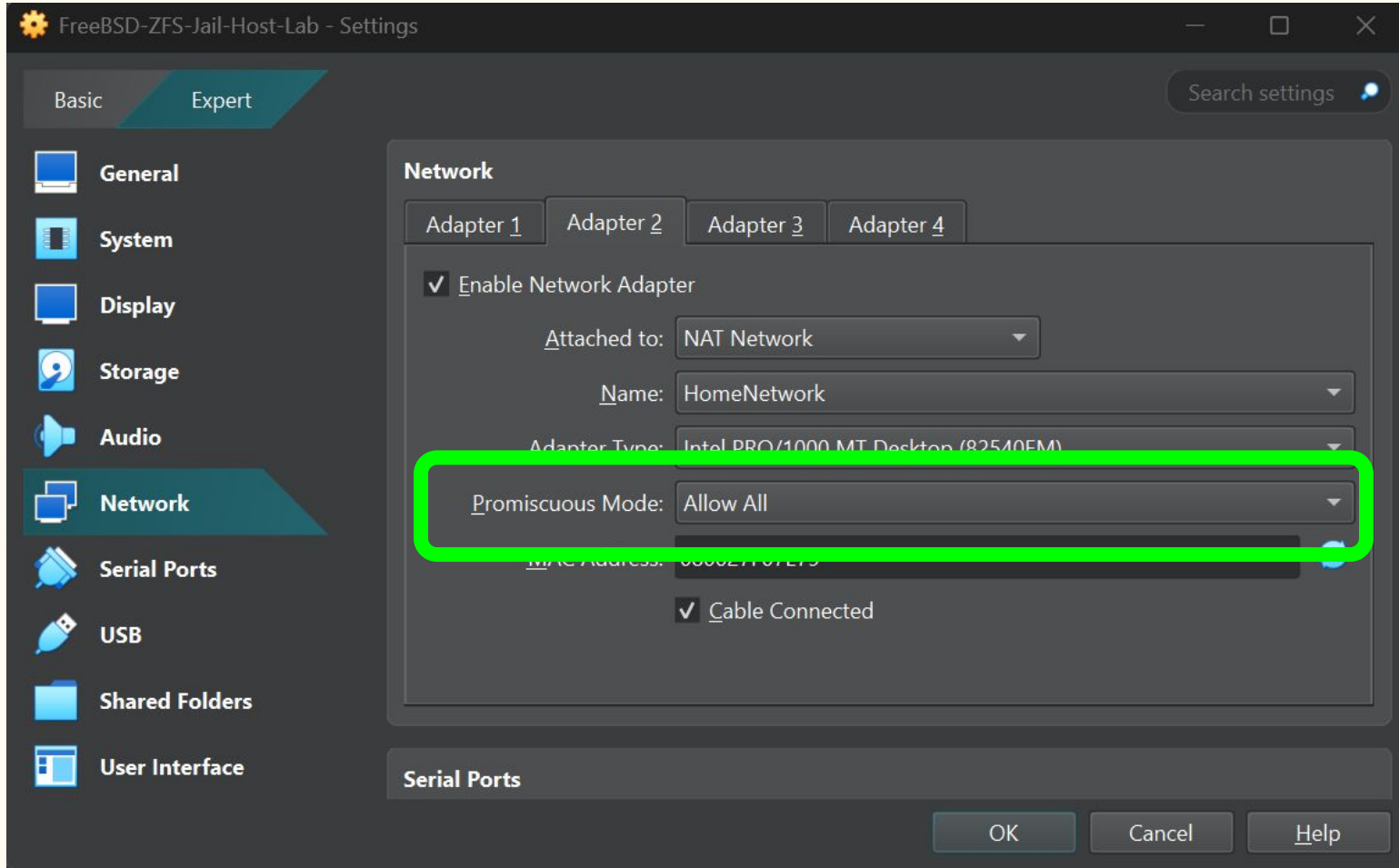
```
labnetbridge: flags=1008843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST>
mtu 1500
options=0
ether 58:9c:fc:10:a3:53
id 00:00:00:00:00:00 priority 32768 hellotime 2 fwddelay
maxage 20 holdcnt 6 proto rstp maxaddr 2000 timeout 120
root id 00:00:00:00:00:00 priority 32768 ifcost 0 port 0
member: e0a_update flags=143<LEARNING,DISCOVER,AUTOEDGE,AUTOPRUNE>
ifmaxaddr 0 port 19 priority 128 path cost 2000
member: e0a_salt flags=143<LEARNING,DISCOVER,AUTOEDGE,AUTOPRUNE>
ifmaxaddr 0 port 16 priority 128 path cost 2000
member: e1a_gateway flags=143<LEARNING,DISCOVER,AUTOEDGE,AUTOPRUNE>
ifmaxaddr 0 port 10 priority 128 path cost 2000
member: labnet flags=143<LEARNING,DISCOVER,AUTOEDGE,AUTOPRUNE>
ifmaxaddr 0 port 5 priority 128 path cost 55
groups: bridge
nd6 options=9<PERFORMNUD,IFDISABLED>
```

```
e0b_gateway: flags=1008843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST,LOWER_UP> metric 0
mtu 1500
    options=8<VLAN_MTU>
    ether 0e:c4:df:af:dc:33
    hwaddr 02:fb:8a:3a:dc:0b
    inet 10.0.2.15 netmask 0xffffffff00 broadcast 10.0.2.255
    groups: epair
    media: Ethernet 10Gbase-T (10Gbase-T <full-duplex>)
    status: active
    nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
e1b_gateway: flags=1008843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST,LOWER_UP> metric 0
mtu 1500
    options=8<VLAN_MTU>
    ether 0e:c4:df:00:00:00
    hwaddr 02:f3:c0:57:b7:0b
    inet 10.66.6.1 netmask 0xffffffff00 broadcast 10.66.6.255
    groups: epair
    media: Ethernet 10Gbase-T (10Gbase-T <full-duplex>)
    status: active
    nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
e2b_gateway: flags=1008843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST,LOWER_UP> metric 0
mtu 1500
    options=8<VLAN_MTU>
    ether 0e:c4:df:f0:7e:79
    hwaddr 02:95:90:67:de:0b
    inet 192.168.66.1 netmask 0xffffffff00 broadcast 192.168.66.255
    groups: epair
    media: Ethernet 10Gbase-T (10Gbase-T <full-duplex>)
    status: active
    nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
root@gateway:~#
```

```
home0bridge: flags=1008843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST,LOWER_UP>  
mtu 1500  
options=0  
ether 58:9c:fc:10:bc:63  
inet 192.168.66.11 netmask 0xffffffff00 broadcast 192.168.66.255  
id 00:00:00:00:00:00 priority 32768 hellotime 2 fwddelay 15  
maxage 20 holdcnt 6 proto rstp maxaddr 2000 timeout 1200  
root id 00:00:00:00:00:00:00 priority 32768 ifcost 0 port 0  
member: e2a_gateway flags=143<LEARNING,DISCOVER,AUTOEDGE,AUTOPTP>  
ifmaxaddr 0 port 13 priority 128 path cost 2000  
member: home0 flags=143<LEARNING,DISCOVER,AUTOEDGE,AUTOPTP>  
ifmaxaddr 0 port 2 priority 128 path cost 20000  
groups: bridge  
nd6 options=9<PERFORMNUD,IFDISABLED>
```

```
roller@freebsd-zfs-jail-host-lab:~ % sudo dhclient home0bridge  
DHCPREQUEST on home0bridge to 255.255.255.255 port 67  
DHCPACK from 192.168.66.1  
bound to 192.168.66.11 -- renewal in 300 seconds.  
roller@freebsd-zfs-jail-host-lab:~ % █
```

Very important VM setting!



VM2: The Laptop

Network

Adapter 1: Intel PRO/1000 MT Desktop (NAT Network, 'HomeNetwork')

Host-only Networks

NAT Networks

Cloud Networks

Name	IPv4 Prefix	IPv6 Prefix	DHCP Server
HomeNetwork	10.28.6.0/24	fd17:625c:f037:1c06::/64	Disabled

General Options

Port Forwarding

Name:

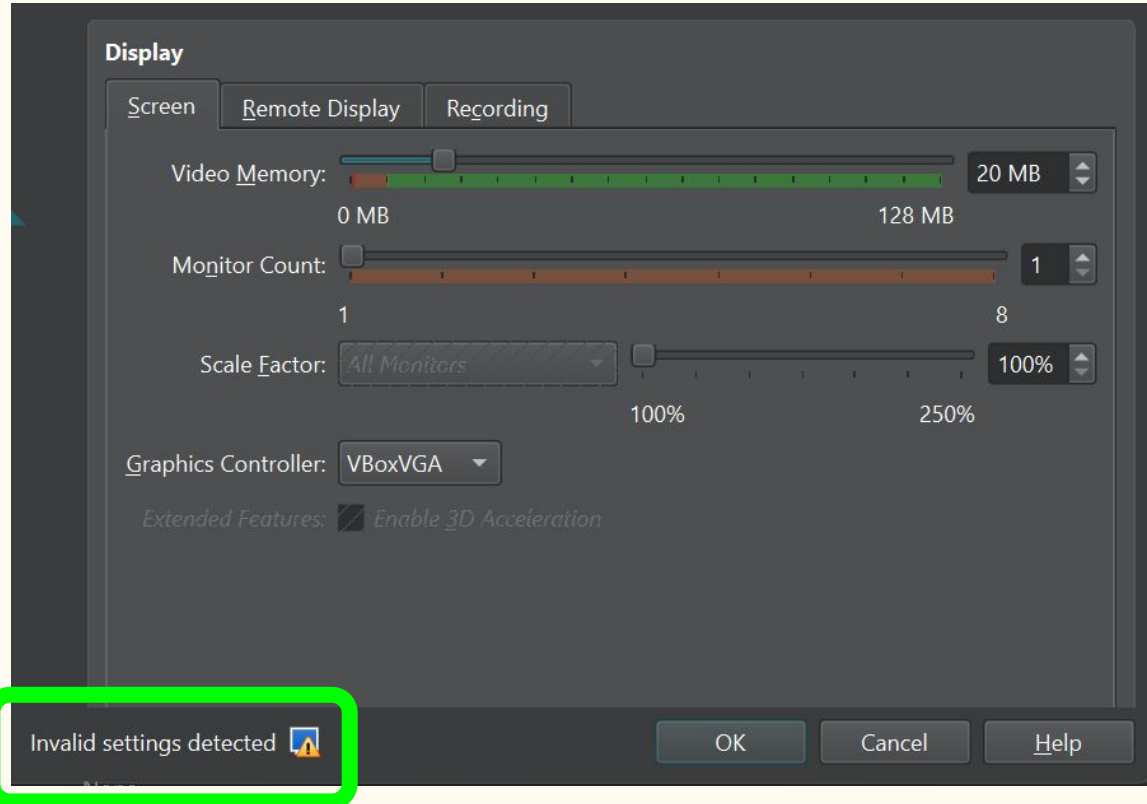
HomeNetwork

IPv4 Prefix:

10.28.6.0/24

☐ Enable DHCP

VM2 Graphics Controller : VBoxVGA



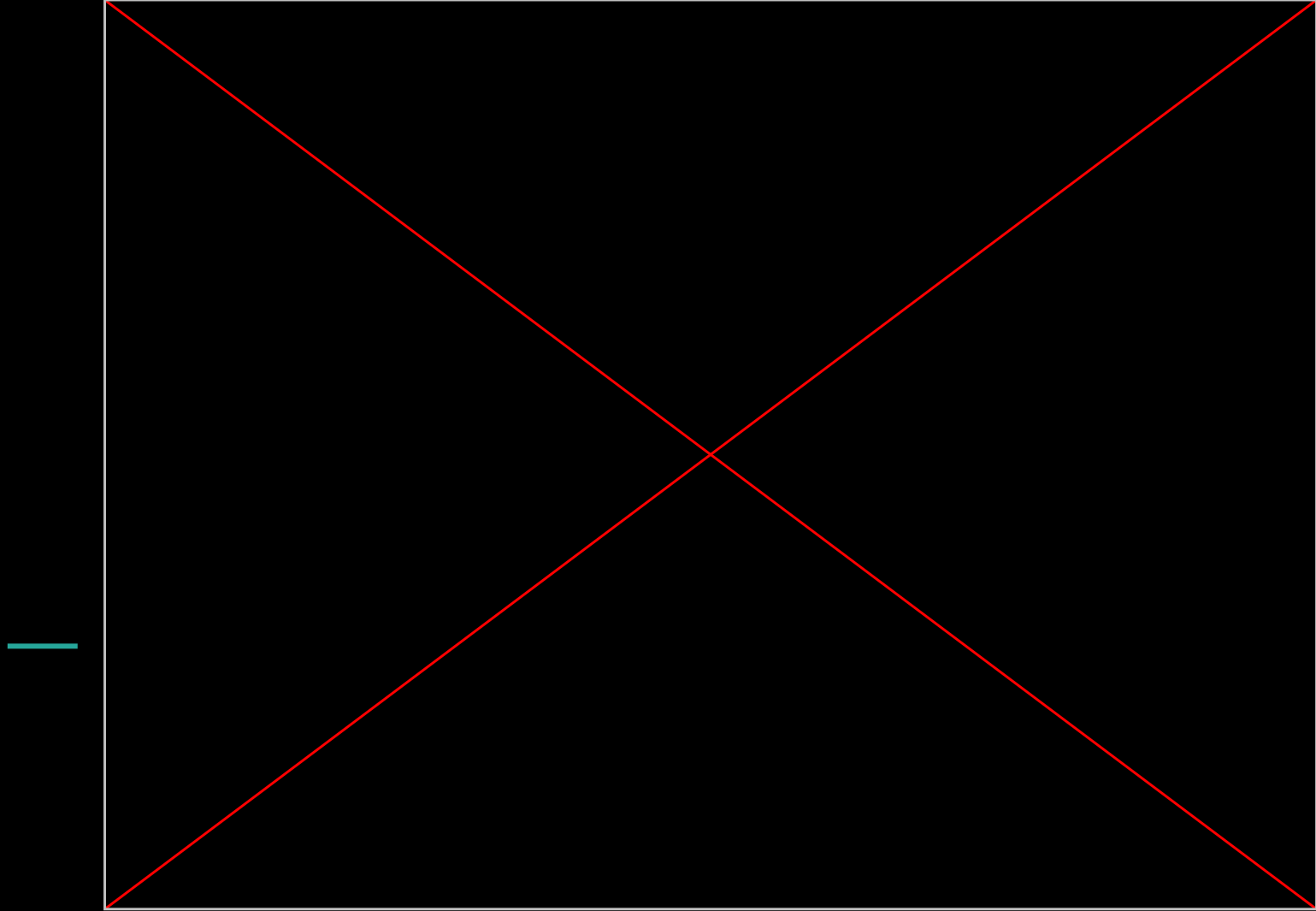
Ignore this,
it's okay!!!

The FreeBSD jails Lab

—

The laptop

The additional jail



Maintaining & Scaling with Salt

```
salt '*' state.highstate
```

```
salt '*' freebsd_update.fetch
```

```
salt '*' freebsd_update.install
```

run in batches with **-b 2** to run on two machines at a time, then the next 2, etc.

```
salt '*' pkg.update
```

```
salt '*' pkg.upgrade
```

```
salt '*' cmd.run freebsd-version
```

```
root@salt:~ # salt '*' state.highstate
```

Ansible quirks

sshpass

- - start-at-task=""

- - step -k -K

Salt quirks

```
root@salt:~ # salt '*' state.highstate
```

```
salt.lab.bsd.pw:
```

```
Minion did not return. [No response]
```

The minions may not have all finished running and any remaining minions will return upon completion. To look up the return data for this job later, run the following command:

```
salt-run jobs.lookup_jid 20250613153946721896
```

```
gateway.lab.bsd.pw:
```

```
Minion did not return. [No response]
```

The minions may not have all finished running and any remaining minions will return upon completion. To look up the return data for this job later, run the following command:

```
salt-run jobs.lookup_jid 20250613153946721896
```

```
ERROR: Minions_returned with non-zero exit code
```

use the event bus to see all the events hitting the Salt event bus

You'll see things like:

- Minion key acceptance
- state.apply start/finish
- Custom events (if you're firing them)
- Orchestration events

```
salt-run state.event pretty=True | grep myminion
```

Any Questions? Feel free to ask

<http://BSD.pw>

<https://docs.freebsd.org/en/books/handbook/>

https://docs.ansible.com/ansible/latest/user_guide/

<https://docs.saltproject.io/en/latest/contents.html>

<https://2018.eurobsdcon.org/tutorial-speakers/>

(Managing BSD Systems with Ansible)

