

## FreeBSD Administration - Feature #403

### Running A VirtualBox Host Inside a FreeBSD Jail Using FreeNAS

06/11/2014 09:09 AM - Daniel Curtis

<b>Status:</b>	Closed	<b>Start date:</b>	06/11/2014
<b>Priority:</b>	High	<b>Due date:</b>	
<b>Assignee:</b>	Daniel Curtis	<b>% Done:</b>	100%
<b>Category:</b>	Jails / Containers	<b>Estimated time:</b>	3.00 hours
<b>Target version:</b>	FreeNAS 9.x	<b>Spent time:</b>	16.00 hours

#### Description

Now that I have most of my web-based servers switched over to FreeBSD using the popular open source FreeNAS variant of FreeBSD. FreeNAS supports Jails, which are lightweight para-virtualized environments, that allows for numerous separate servers to live on a single machine, utilizing the same kernel. I need a full virtualization solution however, this is where VirtualBox comes in. Until recently FreeNAS would fail to load VirtualBox inside of a Jail, due to the lack of a kernel module. This problem was [reported to be resolved](#) by adding a few System Tunables to the FreeNAS installation to enable the required kernel modules.

### ~~Add the required kernel modules to FreeNAS~~

NOTE: As of FreeNAS 9.2.1.6-BETA, there is a VirtualBox template that installs all the necessary packages and kernel modules from the Jails type section, while creating new jails. This should be stable by the next STABLE release.

- **Copy the kernel modules to FreeNAS**

~~I used scp to copy the kernel modules to FreeNAS. According to the discussions going on, the kernel modules can be built in a jail by downloading the kernel sources and compiling the VirtualBox kernel modules and then copying the kernel modules to the core of FreeNAS. Luckily, one developer compiled the VirtualBox 4.3.6 kernel modules for FreeNAS 9.2.1; which I have included. Once the kernel modules have been transferred to a place that FreeNAS can access them, log into the admin panel and open a shell.~~

- **Add the kernel modules to FreeNAS**

```
cd /path/to/modules
mount -uw /
cp *.ko /boot/kernel/
mount -r /
```

- **Enable the kernel modules**

~~To enable these module immediately:~~

```
kldload vboxdrv
kldload vboxnetflt
kldload vboxnetadp
```

~~NOTE: This can be verified by running kldstat, and checking if the kernel modules loaded correctly.~~

~~To enable these modules at boot time, log in to the FreeNAS admin panel and go to System -> Tunables -> Add Tunable and add the following three Tunables:~~

- ~~Variable: **ng\_ether\_load**  
Value: **YES**~~
- ~~Variable: **ng\_gif\_load**  
Value: **YES**~~
- ~~Variable: **vboxdrv\_load**  
Value: **YES**~~
- ~~Variable: **vboxnetflt\_load**  
Value: **YES**~~
- ~~Variable: **vboxnetadp\_load**  
Value: **YES**~~

That's it!

## Create a new Jail for the VirtualBox Host

Now create a Jail by going to [Jails -> Add Jail](#), and configure the Jail accordingly. I chose to use a **Standard** Jail Type in order to stick to the barest possible FreeBSD installation. Once the Jail is finished installing, open the VirtualBox Jail shell and enable and start SSH, create new root SSH key, set root password, add a new privileged user and then close the web shell. I won't go into detail on how to do this, since it is cover in Issue [#329](#)

- When the initial configuration is done, log into the jail using ssh:

```
ssh user@examplejail.com
```

- Update the Jails packages and ports tree:

```
pkg upgrade && portsnap fetch extract
```

- Add a standard user for VirtualBox to be ran as

```
adduser
bob
Bob
[Enter]
[Enter]
[Enter]
[Enter]
[Enter]
[Enter]
[Enter]
[Enter]
[Enter]
[Enter]
[Enter]
[Enter]
SuperSecretPassword
SuperSecretPassword
[Enter]
yes
no
```

- Now install VirtualBox

```
pkg install virtualbox-ose
```

- Add the standard user to the VirtualBox Users group

```
pw groupmod vboxusers -m bob
```

- Enable Network Bridging Support

```
echo 'vboxnet_enable="YES"' >> /etc/rc.conf
```

- Now at this point, log back into FreeNAS and reboot it.

## (Method 1) Create the VirtualBox VM

Since this is a headless (No GUI) installation, the VirtualBox machine must be created from the command line.

- **Create a Windows 7 64-bit machine**

```
VBoxManage createvm --name "Windows 7" --register
VBoxManage modifyvm "Windows 7" --ostype Windows7_64 --memory 2048 --ioapic on --pae on --hwv
rtex on
VBoxManage modifyvm "Windows 7" --nic1 bridged --bridgeadapter1 epair0b
```

NOTE: I needed to use the interface provided to the jail, which can be found using ifconfig.

- **Create the Virtual hard drive**

```
VBoxManage createhd --filename /home/user/VirtualBox\ VMs/Windows\ 7/virtualmachine-disk1.vmdk
--size 10000
```

- **Use a virtual hard drive from a .ova file**

I had a snag trying to import an OVA directly, luckily .ova files are just simple tar archives and can be extracted like so:

```
tar xvf /path/to/virtualmachine.ova
```

Once the OVA has been extracted, there will be a few files:

1. virtualmachine.ovf - The virtualmachine manifest file
2. virtualmachine-disk1.vmdk - The virtualmachine hard drive
3. virtualmachine.mf - A file containing the SHA1 checksums of the above two files

The virtual hard drive can be moved to the VirtualBox machine folder

```
mv /path/to/virtualmachine-disk1.vmdk /home/user/VirtualBox\ VMs/Billing\ VM/
```

- **Attach the Virtual hard drive to the Windows 7 host**

```
VBoxManage storagectl "Windows 7" --name "IDE Controller" --add ide
VBoxManage storageattach "Windows 7" --storagectl "IDE Controller" --port 0 --device 0 --type
hdd --medium /home/user/VirtualBox\ VMs/Billing\ VM/virtualmachine-disk1.vmdk
```

## (Method 2) Import Virtual Machine using existing OVA

This can easily be done by using the VBoxManage command:

```
VBoxManage import Windows7.ova
```

NOTE: Make sure the user is a part of the vboxusers group:

```
pw groupmod vboxusers -m exampleuser
```

NOTE: When transferring large files make sure to check the sha256 sum on both ends.

## Start the Virtual Machine

- Once the VM is created or imported, it can be started using:

```
VBoxHeadless --startvm "Windows 7"
```

- NOTE: I needed to upgrade the packages in the Jail, so after ssh and logging in as root:

```
pkg lock virtualbox-ose-kmod  
pkg update  
pkg upgrade
```

- NOTE: When I launched the Guest VM, I received the following error:

```
VBoxManage: error: Implementation of the USB 2.0 controller not found!  
VBoxManage: error: Because the USB 2.0 controller state is part of the saved VM state, the VM cannot be started. To fix this  
problem, either install the 'Oracle VM VirtualBox Extension Pack' or disable USB 2.0 support in the VM settings  
(VERR_NOT_FOUND)  
VBoxManage: error: Details: code NS_ERROR_FAILURE (0x80004005), component Console, interface IConsole
```

I decided to turn off USB2.0 support, as I will not actually be needing USB remotely (though remote USB should be possible):

```
VBoxManage modifyvm "Windows 7" --usb off
```

- NOTE: When I went to relaunch the VM, I received the following error:

```
VBoxManage: error: Failed to open/create the internal network 'HostInterfaceNetworking-Realtek PCIe GBE Family Controller'  
(VERR_INTNET_FLT_IF_NOT_FOUND).  
VBoxManage: error: Failed to attach the network LUN (VERR_INTNET_FLT_IF_NOT_FOUND)  
VBoxManage: error: Details: code NS_ERROR_FAILURE (0x80004005), component Console, interface IConsole
```

I decided to switch the virtual network card from gigabit (support on the extension pack) to 100M:

```
VBoxManage modifyvm "Windows 7" --nictype1 82540EM  
VBoxManage modifyvm "Billing VM" --bridgeadapter1 epair0b
```

NOTE: Make sure the correct interface is bridged. This can be found using ifconfig.

I still needed to access the VirtualBox GUI to finish installing Windows. Luckily I did not need RDP or VNC, I only needed to forward X over SSH. I logged on with the -X option and opened VirtualBox:

```
ssh -X user@examplehost.com  
VirtualBox
```

## Automatically Start Virtual Machine at Boot

I need to have the VM start when the NAS boots, so I needed to add the following to /etc/rc.conf:

```
vboxheadless_enable="yes"
vboxheadless_machines="vm0"
vboxheadless_vm0_name="3c5f7875-ca0f-494a-9c81-dc29d8119666"
vboxheadless_vm0_user="bob"
```

## Resources

- <https://bugs.freenas.org/issues/1974>

## History

---

**#1 - 06/12/2014 08:37 AM - Daniel Curtis**

- Description updated

**#2 - 06/12/2014 12:41 PM - Daniel Curtis**

- Description updated

**#3 - 06/18/2014 08:59 AM - Daniel Curtis**

- Description updated

**#4 - 06/18/2014 10:56 AM - Daniel Curtis**

- Description updated

**#5 - 06/19/2014 12:21 PM - Daniel Curtis**

- Description updated

**#6 - 06/19/2014 12:41 PM - Daniel Curtis**

- Description updated

- Status changed from In Progress to Resolved

**#7 - 06/19/2014 01:18 PM - Daniel Curtis**

- Description updated

- Status changed from Resolved to Closed

**#8 - 02/15/2015 08:44 PM - Daniel Curtis**

- Project changed from 88 to FreeBSD Administration

- Category set to Jails / Containers

- Target version set to FreeNAS 9.x