FreeBS Server

- Upgrade FreeBSD
- Change Hostname
- First Steps
- Create Jail, Networking and NAT

Upgrade FreeBSD

Upgrading FreeBSD from one minor version to another, such as from FreeBSD 14.0 to 14.2, can typically be done using the `freebsd-update` tool. Here’s a step-by-step guide on how to perform the upgrade:

Step 1: Backup Your Data

Before performing any upgrade, it's crucial to back up your important data and configuration files. You can use tools like `tar`, `cp`, or `rsync` to create backups.

Step 2: Update Your Current System

Make sure your current system is fully updated. Run the following commands:

```
freebsd-update fetch
freebsd-update install
```

Step 3: Upgrade to the New Version

Now, you can upgrade to the new version. Run the following commands:

```
freebsd-update fetch
freebsd-update upgrade -r 14.2-RELEASE
```

Step 4: Install the Upgrade

After fetching the upgrade, install it with:

```
freebsd-update install
```

Step 5: Reboot the System

Once the installation is complete, reboot your system to apply the changes:

```
reboot
```

Step 6: Verify the Upgrade

After rebooting, verify that the upgrade was successful by checking the FreeBSD version:

```
uname -r
```

You should see 14.2-RELEASE as the output.

Step 7: Clean Up

After the upgrade, you can clean up any old files:

```
freebsd-update clean
```

Additional Notes If you have custom kernel configurations or installed ports, you may need to rebuild or reconfigure them after the upgrade. Always refer to the official FreeBSD Handbook or release notes for any specific instructions or considerations related to your system. By following these steps, you should be able to successfully upgrade your FreeBSD system from version 14.0 to 14.2.

Change Hostname

Changing the hostname in FreeBSD can be done in a few simple steps. Here’s how to do it:

Step 1: Change the Hostname Temporarily

To change the hostname temporarily (until the next reboot), you can use the hostname command. Open a terminal and run:

```
sudo hostname new-hostname
```

Replace `new-hostname` with your desired hostname.

Step 2: Change the Hostname Permanently

To make the hostname change permanent, you need to edit the `/etc/rc.conf` file. Open the file in a text editor, such as `vi` or `nano`:

```
sudo vi /etc/rc.conf
```

or

```
sudo nano /etc/rc.conf
```

Look for a line that starts with hostname=. If it exists, change it to your new hostname. If it doesn’t exist, add the following line:

```
hostname="new-hostname"
```

Again, replace `new-hostname` with your desired hostname.

Step 3: Update the /etc/hosts File

It’s also a good idea to update the /etc/hosts file to reflect the new hostname. Open the file:

```
sudo vi /etc/hosts
```

or

```
sudo nano /etc/hosts
```

Find the line that contains the old hostname and change it to the new hostname. It might look something like this:

```
127.0.0.1   localhost
127.0.0.1   old-hostname
```

Change old-hostname to new-hostname:

```
127.0.0.1   localhost
127.0.0.1   new-hostname
```

Step 4: Reboot or Restart Networking

To apply the changes, you can either reboot the system:

```
sudo reboot
```

Or, you can restart the networking service:

```
sudo service netif restart
```

Step 5: Verify the Change

After rebooting or restarting the network, you can verify that the hostname has been changed by running:

```
hostname
```

This should display your new hostname.

By following these steps, you should be able to successfully change the hostname on your FreeBSD system.

First Steps

Add User

To system

```
root@balrock:~ # adduser
Username: username
Full name:
Uid (Leave empty for default):
Login group [username]:
Login group is username. Invite username into other groups? []: wheel
Login class [default]:
Shell (sh csh tcsh bash rbash nologin) [sh]: bash
Home directory [/home/username]:
Home directory permissions (Leave empty for default):
Use password-based authentication? [yes]:
Use an empty password? (yes/no) [no]:
Use a random password? (yes/no) [no]:
Enter password:
Enter password again:
Lock out the account after creation? [no]:
Username   : username
Password   : *****
Full Name  :
Uid        : 1001
Class      :
Groups     : username wheel
Home       : /home/username
Home Mode  :
Shell      : /usr/local/bin/bash
Locked     : no
OK? (yes/no) [yes]:
adduser: INFO: Successfully added (username) to the user database.
```

```
Add another user? (yes/no) [no]:
Goodbye!
```

User SSH-Key

Copy PC SSH KEY

```
ssh-copy-id username@server-ip-address
```

Create SSH-Keypair

```
ssh-keygen -t rsa -b 4096
```

Configure SSHD

```
vim /etc/ssh/sshd_config
```

```
# Authentication:

#LoginGraceTime 2m
PermitRootLogin no
StrictModes yes
MaxAuthTries 6
MaxSessions 10

AllowUsers username

PubkeyAuthentication yes
PasswordAuthentication no
PermitEmptyPasswords no
```

Then delete the line

```
PasswordAuthentication yes
```

from the bottom of your `/etc/ssh/sshd_config`

To Doas

The configuration file for `doas` is typically located at `/usr/local/etc/doas.conf` or `/etc/doas.conf`. You can create or edit this file using a text editor:

```
sudo vim /usr/local/etc/doas.conf
```

```
# Allow user john to execute root commands
permit john
```

Create Jail, Networking and NAT

Step 1: Enable IP Forwarding

First, you need to enable IP forwarding on your FreeBSD host. This allows the host to forward packets between the jail and the outside network.

Edit the `/etc/sysctl.conf` file and add the following line:

```
net.inet.ip.forwarding=1
```

Apply the changes:

```
sysctl net.inet.ip.forwarding=1
```

Step 2: Configure the Host Network Interface

You need to configure the host's network interface to allow NAT.

Identify your network interface (e.g., em0, re0, etc.) using:

```
ifconfig
```

Set up NAT using `pf` (Packet Filter). First, ensure that `pf` is enabled. Edit `/etc/rc.conf` and add:

```
pf_enable="YES"
```

Create or edit the `/etc/pf.conf` file to include NAT rules. Here’s a basic example:

```
ext_if="eth0"  # Replace with your external interface
jails_net="10.10.10.0/24"  # Replace with your jail network

# Set the default policy
set block-policy return
set loginterface $ext_if

# Jail
nat on $ext_if from $jails_net to any -> ($ext_if)
pass in on $ext_if proto tcp from any to ($ext_if) port { 22, 80, 443 }

# Block all incoming traffic by default
block in all

# Allow incoming traffic on specific ports
pass in on $ext_if proto tcp from any to any port { 22, 80, 443 }

# Allow all outgoing traffic
pass out all
```

Load the `pf` rules:

```
sysrc pf_enable="YES"
kldload pf
pfctl -f /etc/pf.conf
pfctl -e
```

Create Classic Jails

Step 1: Enable the Jail Feature

Make sure the jail feature is enabled in your FreeBSD system. You can check this by looking for the `jail` keyword in your `/etc/rc.conf` file. If it's not there, you can add it.

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

Step 2: Create a Directory for the Jail

Create a directory where the jail's filesystem will reside. This is typically done in `/usr/jails`.

```
mkdir -p /usr/jails/website
```

Step 3: Install the Base System

You need to populate the jail directory with a FreeBSD base system. You can use the `make` command to extract the base system into the jail directory.

```
mkdir -p /usr/jails/website
mkdir /usr/jail/media
fetch https://download.freebsd.org/ftp/releases/amd64/amd64/14.2-RELEASE/base.txz -o
/usr/jails/media/14.2-RELEASE-base.txz
tar -xf /usr/jails/media/14.2-RELEASE-base.txz -C /usr/jails/website --unlink
```

Setp 4: Copy important Files & Update

```
cp /etc/resolv.conf /usr/jails/website/etc/resolv.conf
cp /etc/localtime /usr/jails/website/etc/localtime
freebsd-update -b /usr/jails/website fetch install
```

Step 5: Create Network interface for Jail

```
sysrc cloned_interfaces+="lo1"
```

Step 6: Configure the Jail in `/etc/jail.conf`:

```
website {
    path = "/usr/jails/website";
    sysvshm = "new";
    host.hostname = "website.local";
    ip4.addr = "lo1|10.10.10.100/24";  # Assign an IP from your jail network
    allow.raw_sockets;
    allow.socket_af;
    allow.mount;
    mount.devfs;
    devfs_ruleset = 111;
    exec.clean;
    exec.start = "/bin/sh /etc/rc";
    exec.stop = "/bin/sh /etc/rc.shutdown";
}
```

Step 7: Reboot

Reboot Host

```
reboot
```

Step 8: Start the Jail

```
jail -c website
```

Now you should have a jail with networking

Destroy Jail

```
service jail stop website
chflags -R 0 /usr/jails/website/
rm -rf /usr/jails/website/
```