

# Perceus Community Answers to Frequently Asked Questions

## Edited from public support sources by Infiscale

### Basic Perceus Questions

**Q:** What is Perceus?

**A:** Perceus is software that manages operating system provisioning of computer systems. Basically it provides a new method for booting up machines that has more flexibility and scalability than traditional boot-to-disk methods. A Perceus master control server listens on a network for machines booting. When these machines make contact with the Perceus control server, they are sent an operating system image called a VNFS capsule to boot up. This server-client relationship works with both large and small scales of nodes and servers, allowing deployments of computer resources that can meet any task in a manner that is simple to manage, update, upgrade, and re-purpose as needed.

**Q:** How does Perceus work?

**A:** Perceus works in a server-to-nodes relationship. There is a Perceus master control server, or several master servers for very large or hardened deployments. This master server either hosts or has mounted from another machine a state directory on the network fabric. As cluster nodes boot, they make contact with the Perceus master control server. Typically nodes do this via PXE booting on the network fabric. The Perceus master control server answers the node's DHCP request and sends it the Perceus operating system to boot from the state directory. Note that this Perceus operating system is not the runtime operating system the node will be running, but instead is the first of a two stage boot process. Nodes running the Perceus operating system will communicate with the Perceus master control server and receive their final provisioning instructions, including the configured VNFS capsule for that node. This VNFS capsule contains the runtime operating system the node will use. When the node has received its VNFS capsule over the network, it will execute the provisioning instructions from the Perceus master control server, loading the runtime operating system into memory (for a stateless VNFS) or onto the disk (for a stateful VNFS). When complete the node will execute the freshly loaded runtime operating system kernel and clear the first stage Perceus operating system environment out of memory. From this point on, the node will function like any other machine running its loaded operating system. If the VNFS capsule has the Perceus provisiond package installed then the node will communicate its operational state with the Perceus master control server while running, otherwise the node will operate independent and unaware of the Perceus master control server if the provisiond package is not installed.

**Q:** What are the basic options for laying out Perceus?

**A:** The simplest Perceus layout uses a single Perceus master control server that shares on the network a state directory from a locally-mounted filesystem. A second layout would be to still have a single Perceus master control server, but the state directory is remotely mounted from a file server, NAS device, or parallel filesystem that shares the network fabric with the Perceus cluster. The third and complex layout uses multiple Perceus master control servers sharing the provisioning load, all mounting the same non-local state directory over the network.

**Q:** What can Perceus be used for?

**A:** Perceus was designed for the deployment and management of clustered computer systems. However, its flexible architecture has allowed it to be implemented in a whole range of scalable infrastructure uses. Some of these proven scalable usages are:

- High Performance Computing clusters
- Cloud infrastructure provisioning
- Content delivery networks
- Render farms and utility computing
- Large scale web services
- Databases and file systems
- Virtual Machine hosting (VMware, Xen, KVM)
- Appliances and network devices

**Q:** What are the default locations in the filesystem for Perceus components?

**A:** Perceus stores its configuration, local state, libraries, and various files throughout the filesystem. The primary configuration for Perceus is hard-coded to `/etc/perceus`, several components reside here. If multiple Perceus masters are used, each will have its own configuration instance at this location. The Perceus state tree contains node-specific configurations, Perceus modules, provisioning data like VNFS capsules and node scripts, and the primary database directory. It is located at `/var/lib/perceus` by default (but this can be changed by using the `--localstatedir=` option of the configure script, for example, `/usr/var/lib/perceus`). Perceus modules that require configure files store them in `/etc/perceus/modules`. For convenience, links to the imported VNFS capsules and node scripts are located in `/etc/perceus`.

**Q:** What daemons or services are associated with Perceus?

**A:** There are two major services associated with Perceus, and they are both controlled with the init script `/etc/init.d/perceus`. These two services are the Perceus daemon and the Perceus network manager. The Perceus daemon assembles specific node scripts for each node for each provisionary state. The Perceus network manager uses DNSMasq for the internal network functionality Perceus uses to provide DHCP and network booting for nodes. This Perceus network manager service runs independently of the Perceus core and can be used to allocate addresses to non-node machines (like workstations) without having an effect on Perceus or its configurations.

**Q:** What is the difference between the `masterauth` and `passwdfile/groupfile` Perceus modules?

**A:** The `masterauth` module transfers the Perceus master control server's `/etc/passwd` and `/etc/group` files to the nodes. It does not transfer the shadow file, so account authorization needs to be configured to use secure SSH keys and shared home directories from the master.

The `passwdfile` and `groupfile` modules enable fine control over the respective contents of the nodes' `/etc/passwd` and `/etc/group` files, based on node groups or VNFS images. They both also require non-password-based SSH key authentication. The module configuration directory `/etc/perceus/modules/passwdfile/(or groupfile/)` contains a

hierarchy through which the passwd (or group) file may be assembled from node, node group, or VNFS components, as well as the global 'all' file. The 'all' file usually contains the 'root' entry and is included first, all other files are appended after.

## Usage and how-to questions

**Q:** I've installed and configured Perceus, now what?

**A:** After Perceus is installed and configured, it needs to be initialized. This is performed automatically when any Perceus command is run for the first time. What occurs first is a simple registration of the software. The information collected is never used for anything other than informing the developers about what sort of environments Perceus is being used in. The initialization procedure includes creating the Perceus database, creating NFS exports and starting NFS services (if the provisioning mechanism is configured to use local NFS), creating SSH host keys and making them available to VNFS capsules, creating SSH keys for the root user, and starting (also configuring to start on boot) the Perceus services.

**Q:** What is the Perceus command-line structure?

**A:** Perceus commands have this structure:

```
# perceus <command> <subcommand> <arguments>
```

The primary commands are: node, group, vnfs, module, info, configure, contact  
For help with the subcommands and arguments run:

```
# perceus <command> help
```

**Q:** How do you add nodes to Perceus?

**A:** After Perceus is installed, configured, and initialized, the Perceus daemon waits and listens for new nodes to check in or for nodes to be manually added using:

```
# perceus node add <MAC address>
```

The simplest method of adding nodes is to power on the nodes in a logical order and let the Perceus daemon automatically add and configure the nodes as they are detected. The best method for adding a large scale of nodes is to put together a list of the MAC addresses of each node in order and add them to the Perceus setup using a script. Whichever method you use, Perceus will assign the nodes the the settings defined in the /etc/perceus/defaults.conf file.

**Q:** How do you script adding nodes to Perceus?

**A:** A basic script that will add nodes to a Perceus cluster follows:

```
cat /path/to/MACaddressList.txt | while read MAC; do
    echo Adding $MAC
    perceus node add $MAC
done
```

**Q:** How do you import nodes into Perceus from an existing (non-Perceus) cluster?

**A:** Several scripts are distributed with Perceus that facilitate adding nodes from existing clusters, located in `/usr/share/perceus/`. Scripts are available that will import these types of cluster nodes into Perceus:

- IBM SCM
- xCAT
- OSCAR
- Platform OSC
- ROCKS
- Warewulf

There are also utility scripts that will add nodes in the order of switch port assignment, or scan syslog and add a node for any DHCP request, or read a `dhcpd.conf` file for static node address assignments.

**Q:** How do you use XCPU with Perceus?

**A:** XCPU is a framework to run binary applications in a scalable manner on nearly empty nodes. In regards to Perceus, this is done without even provisioning a VNFS to the nodes. If the Perceus `xcpu` module is activated, Perceus will implement XCPU activity directly on the nodes from the first stage bootstrap.

**Q:** How do you generate passphraseless SSH keys for spawning commands on nodes?

**A:** Each individual user will need to configure their own SSH environment to support non-password-based access. This method also requires a shared home directory on all nodes of the cluster, including the Perceus master control server. These commands will generate the passphraseless SSH keys:

```
$ mkdir ~/.ssh
$ ssh-keygen -f ~/.ssh/id_dsa -t dsa -P ""
$ cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys
$ chmod 400 ~/.ssh/authorized_keys
```

**Q:** Nodes are not provisioning correctly, how can I debug this?

**A:** Some troubleshooting advice for nodes not provisioning:

-If the node doesn't PXE boot, on the node end make sure the network interface ports are correct, active in BIOS, not being named differently by the firmware, etc.

-If the node doesn't PXE boot, on the server end make sure the networking is correct for provisioning. Check that Perceus services are running, NFS is working, the firewall isn't active (this blocks DHCP with the nodes), and the /etc/hosts file has entries for master and nodes.

-If the node PXE boots Perceus but hangs somewhere while provisioning try debug modes. Run:

```
# perceusd --debug --foreground
```

or set the node to debug mode with:

```
# perceus node set debug <1,2,3> <node name>
```

Debug 1 and 2 give increasingly verbose debugging messages. Level 3 drops to the BusyBox shell.

## Compatibility (hardware and software) Related Questions

**Q:** What are the requirements for a node to be used with Perceus?

**A:** First, the node must be able to contact the Perceus master control server, so either it needs to support the PXE network boot subsystem or have the Perceus boot software embedded on the hardware itself, for example using Intel Rapid Boot Technology. Second, the node's hardware must be compatible with the first stage Perceus operating system's ininramfs. Third, the node's hardware must be supported by the runtime operating system contained in the provisioned VNFS capsule.

**Q:** What package dependencies does Perceus require?

**A:** Perceus depends on:

```
nfs-kernel-server
autoconf
automake
openssl-devel
elfutils-libelf-devel
zlib-static
nasm
libtool
uuid-devel
```

- rsync
- perl
- perl-DBI
- perl-IO-Interface
- perl-Unix-Syslog
- perl-Net-ARP
- perl-CGI
- perl-Time-HiRes
- bash\_completion (optional, for Perceus commands)
- httpd (optional, for perceus-cgi web interface, or VNFS deliver over HTTP)

**Q:** I've provisioned a node, but its attached keyboard isn't working. What is wrong?

**A:** If the keyboard is USB, there have been cases where the USB controller does not load properly in the Perceus kernel. This was more of an issue before recent Perceus updates, but the problem may still be present with some hardware. You might have to update the Perceus kernel.

**Q:** I want to use a custom network driver kernel module on my Perceus cluster nodes, what should I do?

**A:** You can roll a new VNFS kernel with it installed, or you can use the Perceus modprobe module to detect and activate it when provisioning. To use the Perceus modprobe module:

-Activate the modprobe Perceus module with command:

```
# perceus module activate modprobe init/all
```

-Add the kernel module to the /etc/perceus/modules/modprobe file.

-Make sure the path to modprobe in /var/lib/perceus/nodescripts/init/all/50-modprobe.sh is correct for your distribution if you get a "modprobe: command not found found" error.

**Q:** Can I run DOS or Windows on a Perceus cluster?

**A:** Yes, you can provision different operating systems with Perceus using KVM on the cloud GravityOS VNFS. Pass through special hardware if needed to the Virtual Machine.

**Q:** I've still got this Linux Networx cluster that I'd like to use with Perceus, but the bios is getting in the way. What can I do?

**A:** First, it must be said that even a small cluster with new processors will outperform with better power efficiency an older Linux Networx cluster. If one still wants to use one with Perceus, it is possible for Perceus to reflash the bios' in the nodes itself using a pmod (Perceus module). We've also heard of creating a boot image for the nodes that will perform PXE booting and have Perceus send that to the nodes by changing the /etc/perceus/dnsmasq.conf to have lines:

```
dhcp-option=vendor:Etherboot,60,"Etherboot"  
dhcp-vendorclass=ETHERBOOT,Etherboot  
dhcp-boot=net:ETHERBOOT,bootfile.elf,perceus.your.domain,192.168.0.2  
dhcp-boot=pxelinux.0
```

This will have the LinuxBIOS node boot the first boot image that will then provision normally from Perceus. Or use the lbfash utility to put a proper bios on the motherboard.

## Booting and Provisioning Questions

**Q:** Can I provision a Xen guest with Perceus?

**A:** Yes. You will need to kexec a Xen-enabled kernel from your VNFS capsule. Try the chroot2stateful.sh script, and edit it to specify the XEN\_KERNEL in the config file. That will build a capsule to correctly load the Xen files during kexec.

**Q:** Why do nodes start to PXE boot but fail to get an IP address via DHCP?

**A:** The firewall could be blocking it. Your network adapters could also have come up in a different order than in the previous step, if the node has more than one.

**Q:** How do you get NFSv4 to work with Perceus and DHCP clients to provision hybridized VNFS capsules?

**A:** Your /etc/hosts file needs to be populated with IP addresses for nodes to start. A NFSv3 /etc/init.d/nfs init script file can be modified to only launch rpc.statd, rpc.lockd, and rpc.idmapd. If NFSv4 with kerberos is used, also launch rpc.gssd. You do not need rpc.mountd.

**Q:** My nodes aren't provisioning right, and the errors go by on the monitor too fast to read. What should I do?

**A:** Try using serial redirection on the node, if it is possible with your hardware. When importing a VNFS capsule, Perceus will ask where it should send the terminal output. You can override the local default tty0 and specify a COM port (for example ttyS0). Connect a serial cable to this port, and log the output with a serial terminal (or terminal emulator program) connected to the other end.

## VNFS Activity Questions

**Q:** I want to use a non-RHEL kernel in my RHEL compatible cluster environment, but the VNFS capsule I made isn't working. Why?

**A:** With custom kernels and non-RHEL kernels in a RHEL, CentOS, Scientific Linux compatible environment, you may have to remove from your VNFS' KEXEC\_ARGS the "--args-linux" argument.

**Q:** I need to make my VNFS capsule smaller, what should I do?

**A:** The best way is to use hybridization to mount some of the file system over the network. It is often more practical to just set up the stateless nodes to use internal disks as swap only. This will allow a larger VNFS capsule to be loaded into tmpfs using the swap device. Stripping down VNFS capsules is probably only practical if the nodes have a small amount of ram or the VNFS transfer time over the network needs to be decreased to speed up node boot time. Trimming packages from the VNFS would only gain small amounts of space, and could make the system unstable.

**Q:** I mounted the VNFS, installed a new kernel, and unmounted, but the node boots to old kernel. Why?

**A:** One more thing needs to be changed. In `/var/lib/perceus/vnfs/<VNFS name>/config` is a line to set which kernel version to load when booting from that VNFS capsule.

**Q:** I'd like to make my own RHEL/Caos-NSA/CentOS/Scientific/SUSE/Ubuntu VNFS capsule. How do I do it?

**A:** The most straightforward way is to install Perceus on a machine running the Linux operating system version you want to build a VNFS of and use the appropriate genchroot script in `/usr/share/perceus/vnfs-scripts/`. For example, use the `centos-5-genchroot.sh` script if you are running CentOS 5.4, or the `el6-light.sh` if you are running CentOS 6. These VNFS scripts will build a functioning chroot directory. You can add programs or make configuration changes to this chroot if needed, and when ready to turn it into a VNFS capsule run the `chroot2stateful.sh` or `chroot2stateless.sh` on the chroot directory.

**Q:** I'd like to make a VNFS capsule of a Linux distribution that there is no genchroot script for. What should I do?

**A:** The genchroot scripts found at [http://mirror.infiscale.org/Perceus/vnfs/creation\\_scripts/](http://mirror.infiscale.org/Perceus/vnfs/creation_scripts/) have been created by the Perceus community for many common Linux distributions. If you want to try to build a VNFS of a different distribution than a script is available for, we recommend examining the scripts of related distributions that are available. The contents of the `ubuntu-lucid-genchroot.sh` script, for example, could be a useful starting point for modification in making a Linux Mint or Debian VNFS genchroot script. Once a good chroot directory is in place, the `chroot2stateful.sh` or `chroot2stateless.sh` script can be ran on it to create the VNFS capsule.

**Q:** If my Perceus master node is running a different Linux distro than my provisioned nodes, how can I make changes to the VNFS capsule?

**A:** If the Linux distribution differs, you may not have access to the appropriate package management tools to make changes to the VNFS' mounted chroot directory on the Perceus master. An alternative technique would be to export the rootfs of the capsule from the Perceus master, located at `/var/lib/perceus/vnfs/<VNFS capsule>/rootfs` and mount it from a system running the matching Linux distribution as the VNFS capsule. This way you can use the correct package management tool to add and remove packages to the mounted rootfs directory with the `"--installroot"` (or similar, depending on package manager) option. When changes are complete, from the Perceus master run this command to rebuild the capsule and make those changes final:

```
# perceus vnfs rebuild <VNFS capsule>
```



## Configuration and Customization Questions

**Q:** I'm having trouble with my masterauth module. Perceus is up and running correctly, the module is activated, and provisiond is installed and running on the nodes. Why can't I log in on the nodes with the new user I created?

**A:** The passwd and group files only synchronize with the masterauth module when the nodes check in with provisiond, so make sure the nodes have been restarted or provisiond has re-checked in after a new user has been created. Also, do not use the masterauth and passwdfile Perceus modules at the same time, or the user authorization synchronization between master and nodes will fail. Running perceusd or provisiond in debug foreground mode can help identify errors with Perceus modules.

**Q:** Does Perceus only work with one version of kexec? Is kexec-legacy usable?

**A:** No, both kexec and kexec-legacy are available in /libexec. The item at /sbin/kexec is a wrapper script. You cannot just change the KEXEC variable to use kexec-legacy. You need to edit the /var/lib/perceus/tftp/pxelinux.cfg/default file and add " kexec-legacy" to the end of the append line for the Perceus label. This will get picked up at provisioning time and kexec-legacy will be used instead of kexec.

**Q:** Why isn't a node rebooting into its OS after the first stage of provisioning?

**A:** An incomplete NFS setup could be the problem. Make sure NFS utilities and portmap are installed and running on the Perceus master. Provisioning starts with PXE, but by default finishes using an NFS share.

**Q:** How do you use a remote state directory for Perceus?

**A:** If you intend to use a remote state directory with Perceus, create and mount it via NFS on the master server prior to the installation of Perceus. If Perceus is installed first, a local state directory will be created automatically (/var/lib/perceus) and this data will have to be moved to the remote directory.

**Q:** What are the node attributes, and how do you change them?

**A:** Node attributes are the values contained in the Perceus database pertaining to each node. You can view the node attributes with:

```
# perceus node show <node name>
```

The node attributes you can change are debug, desc, enabled, group, hostname, and vnfs. Changing node attributes is done in this format:

```
# perceus node set <attribute> <attribute value> <node name>
```

**Q:** Why isn't the Perceus masterauth module working? The node boots, but the users' accounts aren't exported from the master.

**A:** The authorization files synchronize when provisiond checks in with the master from the node, so provisiond needs to be installed in the VNFS capsule and it needs to have checked in since any user authorization changes have been made.

**Q:** I want the newer Perceus kernel, can I just install that on my older Perceus deployment?

**A:** You can try by extracting the kernel and initramfs.img files from a newer RPM or .deb and replace the older files on the system (/var/lib/perceus/tftp).

**Q:** Can Perceus be used in a multi-master High Availability setup?

**A:** It can, using additional software. Perceus 2.0 will provide inherent cluster fail-over with redundant control servers. To deploy a multi-master active/passive Perceus cluster use heartbeat. Also provide data synchronization between the masters with DRBD or something similar. Whatever services the master's provide will have to be set up identically. Make /var/lib/nfs and /var/lib/perceus be your synchronized DRBD file systems, plus any other NFS shares the High Availability nodes will be serving. The NFS state directory has to be shared or client mounts will go stale after the passive master takes over. Also, make sure that sm-notify binds (using the "-v" option?) to the shared IP address of the cluster. The dnsmasq leases file needs to be writable on both active and passive servers to survive the fail-over. A standalone dnsmasq may work better here than Perceus' default built-in dnsmasq.

High Availability can also be achieved with virtualization, by migrating the image to different servers. The NFS export for Perceus in this case needs to come from some separate cluster storage.