System Administration

Contents

1 Users and account management **3**
- 1.1 User accounts . . . 3
- 1.2 Service accounts . . . 4
- 1.3 Roles . . . 4
- 1.4 Superuser account . . . 4
  - 1.4.1 SUperuser DO: sudo(1m) . . . 5
- 1.5 Managing accounts . . . 5
  - 1.5.1 useradd . . . 5
  - 1.5.2 userdel . . . 6
  - 1.5.3 usermod . . . 6
  - 1.5.4 groupadd . . . 6
  - 1.5.5 groupdel . . . 6
  - 1.5.6 groupmod . . . 6
  - 1.5.7 roleadd . . . 6
  - 1.5.8 roledel . . . 6
  - 1.5.9 rolemod . . . 6
  - 1.5.10 passwd . . . 6
- 1.6 Role-Based Access Control (RBAC) . . . 6
  - 1.6.1 What is RBAC . . . 7
  - 1.6.2 How to use RBAC . . . 7
- 1.7 Active Directory Integration . . . 7
  - 1.7.1 Introduction . . . 7
  - 1.7.2 Native AD integration . . . 8
  - 1.7.3 Kerberos and LDAP . . . 8
  - 1.7.4 winbind . . . 8

2 Management of System Resources **9**
- 2.1 Basic system information . . . 9
  - 2.1.1 System processes . . . 9
  - 2.1.2 Disk usage . . . 9
  - 2.1.3 Largest files in a directory . . . 9
  - 2.1.4 Who is logged on to the system . . . 9
  - 2.1.5 List all software packages installed on the system . . . 9
- 2.2 System shutdown, reboot, … . . . 9

3 Configuring and Tuning **10**
- 3.1 Configuring a UPS . . . 10

3.2 Fault management (FMA) . . . 10
3.3 Virtual Terminals/Consoles (VT) . . . 11
3.4 Service management (SMF) . . . 11
3.5 Systems logging and monitoring . . . 12

4 Illumos boot process 12

5 Security 12

6 Zones 12
6.1 Zone networking model . . . 13
6.2 Quick Setup Example . . . 14
6.3 System repository configuration . . . 15
6.4 Troubleshooting . . . 16
6.4.1 Fixing zone installation issues . . . 16

7 Storage 17
7.1 Mounting file systems . . . 17
7.1.1 Mounting and Unmounting ISO images . . . 17
7.1.2 Mounting NTFS Volumes - 3rd party support . . . 18
7.2 Configuring OpenIndiana as an ISCSI Target Server - (COMSTAR) . . . 18
7.3 System backups . . . 18
7.4 ZFS . . . 19
7.4.1 Importing ZFS disks . . . 19
7.4.2 How does one mirror their root zpool? . . . 19
7.4.3 How does one create additional zpools? . . . 19
7.4.4 Modifying zpool settings and attributes . . . 20
7.4.5 Modifying zfs file system settings and attributes . . . 20
7.4.6 How does one create additional zfs datasets? . . . 20
7.4.7 Configuring system swap . . . 20

8 Virtualization 20
8.1 OpenIndiana as a virtualization host server . . . 20

9 Localization 21

10 Dtrace 21

11 Configuring Networking 21
11.1 Automatic Configuration (NWAM) . . . 21
11.2 Changing from NWAM to Manual Configuration . . . 21
11.3 More on Automatic Configuration (NWAM) . . . 23
11.3.1 Using NWAM configuration tools . . . 23
11.3.2 Network automagic online help . . . 27
11.3.3 Troubleshooting NWAM . . . 27

12 Clustering with Open HA Cluster 27

Once OpenIndiana has been installed, the system will require monitoring to ensure smooth operation. Software will periodically have to be updated, redundant software removed, new users added to the system, etc. All these activities and many more are referred to as system administration.

Basic system administration can be reduced to a number of common tasks:

- Users and account management
- Management of system resources
- Installation and maintenance of software

While it is certainly possible to add more to this list or select alternative items, this small selection is readily absorbed and is convenient to illustrate a number of essential concepts central to OpenIndiana system administration.

> ⓘ **NOTE:**
> Administrative commands are usually expected to be run with elevated privileges - directly from root user, via `sudo(1M)` or `pfexec` (if user was granted privileges via RBAC). In this document commands which require elevated privileges are prefixed with "#". Commands, which don't require elevated privileges, are prefixed with "$".

1 Users and account management

OpenIndiana allows multiple users to work on the same computer at the same time. Only one person can sit in front of the monitor and keyboard. However, many can be remotely logged into machine and work on it. If a user wants to use the system, he needs an account.

There are also special service accounts which are used by system services.

1.1 User accounts

User accounts are primarily used for day-to-day tasks. Every user accessing the system should have his own unique account. This allows the administrator of the system to find out who is doing what. This also allows him to set different access rights for each user separately.

When system is installed, usually one user account is created. Additional accounts can be added later.

Every user account has some attributes associated with it.

- **username** - name of the account, which is typed at the login screen. Username format is briefly described in passwd(4)
- **password** - password associated with the account.

> △ **CAUTION:**
> Accounts without password should not exist on the system as they could put the system at the security risk!

- **UID** (user ID) - unique numerical ID of the account in the system. The maximum value for uid is 2147483647. However, for compatibility reasons one should not use numbers over 65535.
- **GID** (group ID) - unique numerical ID of the account's primary group.
- **comment** (also referred to as gecos) - account description Often this filed is set to real user name.
- **home directory** - a path where user will be after he logs into the system.
- **login shell** - user's initial shell program.
- **password last change time** - time when the password was lastly changed.
- password aging information - several shadow(4) fields determining when password should be changed
- **min** - the minimum number of days required between password changes;
- **max** - the maximum number of days the password is valid;
- **warn** - the number of days before password expires that the user is warned.

1.2 Service accounts

Service (system) accounts are used by applications, which are running on the system and are providing some services to the network such as DNS, SMTP or WWW. For security reasons these services are ran under non-privileged accounts.

OpenIndiana comes with several service accounts such as `webservd` for web servers, `pkg5srv` for `pkg(5)` server or `nobody`. `nobody` is a system account for services needing unprivileged user. However, the more services are ran under this account, the more privileged it becomes as it gains access to service processes and files.

1.3 Roles

A role is a basic unit of Role-Based access control (RBAC) or set of privileges one can assume. A role can be thought of as a no-login account. It has most of the attributes of normal user account and is identified like normal user, but is not allowed to log into a system directly. One should login using regular account and use su(1M) to assume the role.

1.4 Superuser account

Every UNIX-like system has the superuser account, named root, used for system administrative tasks. This account has UID 0.

Using this account for everyday usage like web browsing, email reading or movie watching is not recommended as root account can operate without any restrictions or limits and could cause serious damage to the system.

If you created user during installation, root is created as role (not a regular account). This role is assigned to the user created during the installation. This means that you can't directly log into a system using the root account. One has to log in as the created user and switch to root role using su(1M).

However, if the user was not created during installation, then the root is created as regular account and is able to log in directly. Note, that even when root is created as role, one can use this account directly to log into the system when it's booted in single user mode.

1.4.1 SUperuser DO: sudo(1m)

It is not always feasible for one user to perform all administrative tasks. It would be more flexible if some tasks could be performed by some, say, experienced users. To enable some users to carry out a command with *all* root privileges, or to *do* an administrative command `sudo(1M)` can be used.

The `sudo` command, i.e., superuser do, permits a regular user to execute the specified set of commands with supperuser privileges without having to become the superuser.

If user was permitted to use some commands with superuser privileges via `sudo` he/she can execute them with elevated privileges simply starting command with `sudo`.

To permit a user to use `sudo`, the superuser edits `/etc/sudoers`.

This can be done using special `visudo(1M)` tool. `visudo` tool edits `/etc/sudoers` file performing various syntax checks. `sudoers(1)` provides details on the precise means to appropriately grant user elevated privileges via sudo.

Example:

To shutdown the system, root privileges are required. If a standard user issues the `shutdown` command, the system will issue a warning. However, if the user has been enabled to use sudo, then the user can now shutdown the system:

```
$ sudo shutdown -i5 -g0 -y
```

1.5 Managing accounts

There are several tools in OpenIndiana to manage user accounts. They are listed in the table below.

Tool	Description
useradd(1M)	creates new account on the system
userdel(1M)	deletes account from the system
usermod(1M)	modifies account information on the system
groupadd(1M)	creates group on the system
groupdel(1M)	deletes group from the system
groupmod(1M)	modifies group information on the system
roleadd(1M)	creates new role on the system
roledel(1M)	deletes role from the system
rolemod(1M)	modifies role information in the system
passwd(1)	changes login password and password attributes

1.5.1 useradd

useradd is a program to create user accounts on the system. When creating account one can set some attributes of the newly created account such as comment, group membership, home directory path, UID number of the account or login shell.

1.5.2 userdel

One can use `userdel` to remove the account from the system.

1.5.3 usermod

`usermod` is used to modify properties of existing account.

1.5.4 groupadd

`groupadd` creates a new group definition on the system by adding the appropriate to the /etc/group file.

1.5.5 groupdel

`groupdel` deletes a group definition from the system.

1.5.6 groupmod

When one needs to modify group attributes, `groupmod` should be used.

1.5.7 roleadd

`roleadd` is used create roles in the system.

1.5.8 roledel

`roledel` deletes selected role from the system.

1.5.9 rolemod

`rolemod` modifies role's information on the system.

1.5.10 passwd

`passwd` changes password for user accounts. An unprivileged user may only change the password for his/her own account, while the superuser may change the password for any account. Privileged user can also use `paasswd` to change login password attributes (such as expiration date) or lock the account.

1.6 Role-Based Access Control (RBAC)

The *all-or-nothing* power assigned to the root user has its obvious limitations. While sudo is an improvement by limiting root privileges for only several commands, more granular control is often desired.

An improvement on the above systems would be one in which privileges could be assigned on a more fine-grained and selective basis.

Imagine a user assigned the task of administrating some particular hardware, for example, printers attached to the system. A more desirable system would be one in which this user had the ability to permit users to use a printing device, remove print jobs from the print spool or add new printers to the system. Moreover, it would be advantageous if it were possible to assign privileges to perform only these actions and none other.

RBAC was developed to accomplish this.

1.6.1 What is RBAC

1.6.2 How to use RBAC

The root user or a user with sudo enabled can shutdown the system. We can use RBAC to enable a user to be able to shutdown the system. However, we can create a role that allows only the privilege to shutdown the system, and no additional privileges. We can then assign this role to one or several users.

- assign a privilege to a role to shutdown the system

```
# roleadd shutdown
```

- Assign a password

```
# passwd shutdown
```

- Assign this role to a user

```
# usermod -R shutdown whoever
```

- Create a SHUTDOWN profile

```
# echo "SHUTDOWN:::profile to shutdown:help=shutdown.html" >>
 ↪  /etc/security/prof_attr
```

- Okay, now assign the role profile SHUTDOWN to the role shutdown

```
# rolemod -P SHUTDOWN shutdown
```

- Assign some administrative command to profile

```
# echo "SHUTDOWN:suser:cmd:::/usr/sbin/shutdown:uid=0" >> /etc/security/exec_attr
```

- Use it

```
$ su shutdown
# shutdown -i5 -g0 -y
```

Now user whoever can shutdown the system.

The `pfexec` command is more flexible in the number of privileges that can be assigned to a user.

1.7 Active Directory Integration

1.7.1 Introduction

There are at least three different possible approaches for Active Directory authentication and each has its pros and cons.

1. Use the new native AD integration with **idmap**, **nss_ad** and **kclient**, this will work with CIFS and NFS out of the box.
2. Use Kerberos and LDAP (**kclient**, **ldapclient**, **pam_krb5** and **nss_ldap**).
3. Use **windbind** (**pam_winbind** and **nss_winbind**).

1.7.2 Native AD integration

- Pro: Fully integrated and native tools only
- Cons: Doesn't work for UNIX services other than CIFS and NFS. The ephemeral id mapping strategy supposedly wasn't designed for other UNIX services. As a result several problems arrise, one of them is that the mappings aren't constant over the lifetime of UNIX processes which severly breaks UNIX semantics. Depending on your UNIX service you will see unexpected results or even process crashing.

In order to use this approach with any UNIX service (eg **FTP**) you need to enable *directory-based name mapping* and install **IDMU** (Identity Management for UNIX) on the AD server.

Refer to the original documentation from Oracle for getting this working: nss_ad, CIFS.

1.7.3 Kerberos and LDAP

- Pro: Fully integrated and native tools only
- Cons: Requires installation of additional role services (IDMU, Identity Management for UNIX) on the Active Directory side

> ⓘ **NOTE:**
> On the Wiki the 'Kerberos and LDAP' page was a separate detailed article on configuration of Windows Server 2008 Active Directory to work with OI. As of 2021, Windows Server 2008 is EoL. This section should not be migrated until it has been checked and updated for recent Windows. Ideally this information should be on a separate page (potentially community contributions) as it's large and contains a lot of Windows information.

1.7.4 winbind

- Pro: Easy setup, no AD modification
- Cons: Depends on 3rd party software (Samba), group membership resolution didn't work

> ⓘ **NOTE:**
> As above note, the 'winbind' page was a separate detailed article which needs to be checked and updated before it's migrated.

2 Management of System Resources

2.1 Basic system information

2.1.1 System processes

```
$ prstat
```

This command provides a host of information on all processes running on the system. Some of the information provided is as follows:

- percentage of CPU used by each process
- amount of memory consumed by each process
- unique id of each process (which can, for example, be used to stop the process)

2.1.2 Disk usage

```
# df -h
```

Provides information on disk size, amount of space used and available free space for all attached storage devices. The `-h` option reports this information in human readable format.

2.1.3 Largest files in a directory

Go to the directory using the `cd` command and issue the following command:

```
$ du | sort -n
```

This will list the size of each file in the current directory and all sub-directories, starting with the smallest up to the largest files.

2.1.4 Who is logged on to the system

```
$ listusers
```

2.1.5 List all software packages installed on the system

```
$ pkg list
```

2.2 System shutdown, reboot, …

OpenIndiana defines a number of different system states known as run-levels. You can change from one system state to another by using the `shutdown` command and specify the run-level using the `i` option. You can always determine the run-level via `who -r`.

You must be root or have root privileges (e.g., using `sudo`) to send the system into a different state, i.e., turn off, reboot, etc. Shutdown and turn off all hardware (if supported by the hardware) now:

```
# shutdown -i5 -g0 -y
```

Changing the run-level of the system can be disruptive to other users currently using the system. Thus, it is always wise to establish who is currently logged onto the system before changing the run-level.

- `-i [run-level]` is used to specify the run-level. This is either a digit or a single letter. Here are some run-levels available:
- `5` stop all system services, and turns off hardware devices, etc.
- `6` reboot the system.
- `1` single-user mode. Primarily used for system maintenance.
- `S` single-user mode where only a command line terminal is available.
- `-g [seconds]` is used to specify the number of seconds after which to commence shutting down services. `0` immediately intiates shutting down all services.
- `-y` automatically answers all system questions with 'yes'. The shutdown process is not interrupted by system prompts requiring user-interactive intervention.

3 Configuring and Tuning

There are a few tools to configure and tune an OpenIndiana system. One of the tools is sysding, a tool that is used by the current OpenIndiana installer to setup static IP addresses when NWAM (automatic network configuration) is not used.

For more information on sysding, see the manpage and example config file :

```
# man sysding
```

The configuration file is :

```
/etc/sysding.conf
```

There is also a logfile :

```
/var/log/sysding.log
```

For example a simple /etc/sysding.conf configuration file for a static IP is :

```
setup_interface e1000g0 v4 10.0.2.16
setup_route default 10.0.2.2
```

This service is meant to be run only once and sets a boolean flag config/finished. In case you want to run sysding again, you'll have to force the finished flag to false :

```
svccfg -s sysding:system
setprop config/finished = false
refresh
```

3.1 Configuring a UPS

- NUT?

3.2 Fault management (FMA)

< place holder >

3.3 Virtual Terminals/Consoles (VT)

Virtual Terminals/Virtual Consoles (VT) are used to switch between terminals and using system in text mode with *Ctrl+Alt+F1,F2..F8* key combinations.

> ⓘ **NOTE:**
> Switching to VTs using *Ctrl+Alt+Fn* during current desktop session (at **VT7**) can result in **X server** and **lightdm** (or **gdm**) restarting and stopping all applications running within it. Currently, there could be problems with getting back to gdm/Xorg session if switching to VTs after gdm restart. Use `svcadm restart ligthdm` (or `svcadm restart gdm`) command again, to have lightdm/gdm Xorg session restarted at **VT8** (*Ctrl+Alt+F8*).

At fresh Openindiana install, VT/Consoles are **not enabled by default**. One needs to set up vtdaemon and console-login:vt2 (till :vt6) and enable **vtdaemon** options/hotkeys property:

```
pfexec svcadm enable vtdaemon
pfexec svcadm enable console-login:vt2
pfexec svcadm enable console-login:vt3
pfexec svcadm enable console-login:vt4
pfexec svcadm enable console-login:vt5
pfexec svcadm enable console-login:vt6
```

Or do that in a one-liner Bash script: `for i in 2 3 4 5 6 ; do pfexec svcadm enable console-login:vt$i; done;`

Then, enable options/hotkeys property (*Ctrl+Alt+Fn*) to switch VTs and refresh and restart **vtdaemon** service:

```
pfexec svccfg -s vtdaemon setprop options/hotkeys=true
pfexec svcadm refresh vtdaemon
pfexec svcadm restart vtdaemon
```

Optionally, you can also disable VT consoles auto screen locking (recommended for personal use, not recommended on server): `pfexec svccfg -s vtdaemon setprop options/secure=false`

The above was inspired by this blog by Danx on Virtual Consoles.

3.4 Service management (SMF)

> ⓘ **NOTE:**
> ITEMS TO WRITE ABOUT: provide more detailed explanations.

List services:

```
$ svcs # list (permanently) enabled services
$ svcs -a # list all services
$ svcs -vx # list faulty services
```

Get information about a service:

```
$ svcs <service name> # one-line status
$ svcs -x <service name> # important information
$ svcs -d <service name> # check the service's dependencies
$ svcs -l <service name> # all the available information
```

Start a service:

```
# svcadm enable <service name> # permanently enable/start
# svcadm enable -t <service name> # temporary start (won't survive a reboot)
# svcadm enable -r <service name> # permanently enable/start service along with its
 ↪  dependencies
```

Restart / reload a service:

```
# svcadm refresh <service name> # reload the service's configuration
# svcadm restart <service name> # restart the service
```

3.5 Systems logging and monitoring

> ⓘ **NOTE:**
> ITEMS TO WRITE ABOUT:
> - Where to find the logs (`/var/log`, `/var/svc/log`).

4 Illumos boot process

< place holder >

5 Security

< place holder >

6 Zones

> ⓘ **NOTE:**
> ITEMS TO WRITE ABOUT:
> - Need to mention some of the changes to zone management…e.g..
>   - sys-unconfig gone.
>   - sysding replaced syscfg
>   - now have to have DNS, root password, etc. all configured inside the zone before being able to logon using `zlogin -C <zonename>`, otherwise have to do `zlogin <zonename>`.
>
> So a fair amount of stuff has changed there.

Zones are an OpenIndiana feature that provides operating system-level virtualization. Each zone is managed as a completely separate OpenIndiana machine. Zones have very low overhead and are one of the most efficient forms of OS virtualization.

The global zone (GZ) is the operating system itself, which has hardware access. From the global zone, non-global zones (NGZ) are created and booted. Boot time for non-global zones is very fast, often a few seconds. The CPU, network, and memory resources for each zone can be controlled from the global zone, ensuring fair access to system resources. Disk space access is usually controlled by ZFS (with quotas and reservations if needed), as well as mounting of filesystem resources with NFS or lofs. As with other forms of virtualization, each zone is isolated from the other zones – zones cannot see processes or resources used in other zones. The low marginal cost of a zone allows large systems have tens or even hundreds of zones without significant overhead. The theoretical limit to the number of zones on a single platform is 8,192.

Different releases of (Open)Solaris used different packaging distribution method for the global zone. OpenIndiana zones use two basic brands - “ipkg” and “nlipkg”, which are based on IPS Packaging. The brand determines how zone is initialized and how zone’s processes are treated by kernel. Both type of zones represent a PKG image. “ipkg”-branded zones are tightly coupled with GZ. Image pakaging system (IPS) knows about ipkg-branded zones and can perform several actions simultaneously in GZ and NGZ. For example, you can update all your zones and GZ with a single “pkg update -r” command. IPS can ensure some depenencies between packages in GZ and NGZ. To allow this it cheks that NGZ’s publishers are a superset of GZ’s publishers and their properties are the same (for example, stickiness or repository location). As this is not always suitable for development zones, “nlipkg”-branded zones were introduced. “nlipkg”-branded zone behave like completely independent instance and IPS ignores them during operations in GZ.

An easy way to implement zones is to use a separate ZFS file system as the zone root’s backing store. File systems are easy to create in ZFS and zones can take advantage of the ZFS snapshot and clone features. Due to the strong isolation between zones, sharing a file system must be done with traditional file sharing methods (eg NFS).

When each zone is created it comes with a minimal set of packages, and from there you can add and use most packages and applications as required.

6.1 Zone networking model

OpenIndiana zones can use one of two networking models: a shared IP stack and an exclusive IP stack. There are pros and cons to each of them.

Low-level system networking components, such as the ipfilter firewall and the kernel IP routing tables attach to an “IP stack”, and are thus either unique to a zone or shared by all zones with the one shared stack. There are also some other nuances, such as that the zones with the shared stack can communicate over IP directly, regardless of their subnetting and, to some extent, default firewall packet filtering (that has to be specially configured), while exclusive-IP zones with addresses in different subnets have to communicate over external routers and are subject to common firewall filtering.

The global zone defines which physical networks and VLANs the NGZ has access to, and hands down the predefined networking interfaces (the NGZ can not use or create other interfaces). Also, while shared networking allows to configure and attach (or detach) network interfaces from GZ to the NGZ “on the fly”, changes in exclusive networking require reboot of the zone to propagate device delegation.

A zone with an exclusive IP stack can have all the benefits of dedicated hardware network-

ing, including a firewall, access to promiscuous sniffing, routing, configuration of its own IP address (including use of DHCP and static network addressing), etc. This requires a fully dedicated NIC. Usually this is a VNICs - a virtual adapter with own MAC address, that operate as if the VNIC was plugged directly into local LAN in a particular (or default) VLAN. Note also that the local zones with an exclusive IP stack are not subject to the host's shared-stack firewall.

> ⓘ **NOTE:**
> Note that if you create zone in VM, the hypervisor can require you to provide some information about its mac addresses. For example, if you configure bridged networking on VirtualBox running on OpenIndiana, you must set secondary-macs property of the VNIC, delegated to VM, to the set of mac addresses of VNICs created inside VMs and enable promiscous mode on VirtualBox network adapter.

6.2 Quick Setup Example

Zone creation consists of two steps - creating zone configuration and zone installation or cloning. Zone configuration determines basic parameters, such as zone's root location and provided resources. Zone configuration is performed using zonecfg tool, zone administration (for example, installation) is performed using zoneadm tool.

For example, we create a simple zone with shared networking model:

```
# zonecfg -z example
example: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:example> create
zonecfg:example> add net
zonecfg:example:net> set physical=e1000g0
zonecfg:example:net> set address=192.168.0.10/24
zonecfg:example:net> end
zonecfg:example> set zonepath=/zones/example
zonecfg:example> verify
zonecfg:example> commit
zonecfg:example> exit
```

Here `create` puts you inside the zone configuration program where you can change and update settings particular to the zone specified with -z. `zonecfg` break different resource groups of data, you add a new resource with add. The next block adds ressource "net", configuring network in default `shared` ip-type mode. It allows zone to share IP stack with GZ. If you want to get dedicated nic in NGZ, you have to use `set ip-type=exclusive`. In exclusive mode zone has complete control over network interface and you can't assigned address in zonecfg prompt. After network configuration `zonepath` is set. It's a location for zone's root file system, which should be a ZFS filesystem. The `verify` command checks that no mistakes were made. Finally changes are committed (saved to zone configuration file).

If you want to use dedicated networking for your zone, you should create VNIC and delegate it to the zone.

```
# dladm create-vnic -l e1000g0 vnic0
```

If you want to specify VLAN id for VNIC, use `-v` `dladm create-vnic` option:

```
# dladm create-vnic -l e1000g0 -v 123 vnic0
```

Now you can create your zone and delegate VNIC to it (note that we set zone's `ip-type` to `exclusive`):

```
# zonecfg -z example
example: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:example> create
zonecfg:example> set ip-type=exclusive
zonecfg:example> add net
zonecfg:example:net> set physical=vnic0
zonecfg:example:net> end
zonecfg:example> set zonepath=/zones/example
zonecfg:example> verify
zonecfg:example> commit
zonecfg:example> exit
```

After configuring a zone you can install it with `zoneadm install` subcommand:

```
# zoneadm -z example install
```

During installation pkg image rooted at $zonepath/root is created and minimal set of packages is installed to the image. When installation finishes, zone can be booted with `zoneadm -z example boot` command. If you want your zone to boot automatically during system startup, you should set autoboot parameter to true during zone configuration:

```
zonecfg:example> set autoboot=true
```

Once zone is booted you can log in locally with `zlogin example`. If you created zone with dedicated network adapter, you should configure it inside zone.

> ⓘ **NOTE:**
> Note, that on first zone boot sysding(1M) will set root's password to NP. Before this happened you will not be able to login to zone with zlogin, so this command will not work on early startup stage.

6.3 System repository configuration

On OpenIndiana it is possible to allow NGZs to access configured publishers via GZ proxy service (so-called zone proxy daemon). This can be useful for sharing pkg cache between zones or to provide network access for performing updates to otherwise restricted zone environment (i.e. to zone without Internet access).

The functionality is provided by series of services in GZ and NGZs. In GZ two services are running: system repository service and zones proxy daemon (see `pkg.sysrepo(1M)`). In NGZ zones proxy client communicates with GZ's zone proxy daemon. System repository service `svc:/application/pkg/system-repository` is responsible for providing access to the package repositories configured in a reference image through a centralized proxy. Zones

proxy daemon service `svc:/application/pkg/zones-proxyd` starts on system boot and registers door in each running ipkg-branded zone (the door is created at `/var/tmp/zoneproxy_door` path). Later, on zone startup or shutdown /usr/lib/zones/zoneproxy-adm is used to notify zones-proxyd, so that it could create the door for the zone or to cleanup it. Zones proxy daemon client `svc:/application/pkg/zones-proxy-client:default` runs in NGZ and talks to GZ's zones-proxyd via created door.

> ⓘ **NOTE:**
> Note, you can't use system repository with nlipkg-branded zones.

IPS determines if it should use zones proxy client in zone based on image's `use-system-repo` property (defaults to False).

To configure your system to use system repository, perform the following actions.

1) In global zone:

```
# pkg install pkg:/package/pkg/system-repository pkg:/package/pkg/zones-proxy
# svcadm enable svc:/application/pkg/system-repository:default
# svcadm enable svc:/application/pkg/zones-proxyd:default
```

2) In non-global zone:

```
# pkg install pkg:/package/pkg/zones-proxy
# svcadm enable svc:/application/pkg/zones-proxy-client:default
# pkg set-property use-system-repo True
```

After this in NGZ's publisher description you'll see system-repository location:

```
# pkg publisher
PUBLISHER                    TYPE     STATUS P LOCATION
openindiana.org (non-sticky, syspub) origin   online T <system-repository>
hipster-encumbered (syspub)     origin   online T <system-repository>
```

You can check if your configuration works by issuing `pkg refresh` command in the zone. `pkg(1M)` should contact repository indirectly via zones-proxy-client.

To revert your zone to proxy-less configuration, run

```
# pkg set-property use-system-repo False
```

6.4 Troubleshooting

Zone configuration and management operations can fail for a number of reasons, sometimes obscure. This section contains information on how to solve different well-known issues.

6.4.1 Fixing zone installation issues

Zones on OpenIndiana use ZFS features to manage boot environments. Zone's root and its parent directory should be a ZFS dataset. It's not enough for zone's root (or its parent directory) to be just a directory on ZFS filesystem. If you try to create a zone which root doesn't satisfy this requirement, you can get the following error:

```
# zoneadm -z example install
Sanity Check: Looking for 'entire' incorporation.
ERROR: the zonepath must be a ZFS dataset.
The parent directory of the zonepath must be a ZFS dataset so that the
zonepath ZFS dataset can be created properly.
```

To fix this, you can just uninstall the zone and create its root dataset manually:

```
# zoneadm -z example uninstall
Are you sure you want to uninstall zone example (y/[n])? y
# zfs create rpool/zones/example
# chmod 700 /zones/example
# zoneadm -z example install
```

7 Storage

< place holder >

7.1 Mounting file systems

> ⓘ **NOTE:**
> ITEMS TO WRITE ABOUT:
> - Need a walkthrough of mounting options for other filesystems…FAT, UFS, etc.

7.1.1 Mounting and Unmounting ISO images

One can use lofiadm(1M) to mount ISO images by attaching them to a block device.

```
pfexec lofiadm -a /path/to/foo.iso /dev/lofi/1
```

When the above line is repeated for several ISO images, issue the `lofiadm` command to list which ISO images are attached to which block devices.

```
pfexec lofiadm
Block Device File Options
/dev/lofi/1 /home/scarcry/foo.iso -
/dev/lofi/2 /home/scarcry/bar.iso -
```

Use the mount(1M) command to mount an image:

```
pfexec mount -F hsfs -o ro /dev/lofi/1 /mnt
```

Check the mounted image by issuing `ls` on the mount point.

```
ls /mnt
```

To unmount and detach the image(s):

```
pfexec umount /mnt
pfexec lofiadm -d /dev/lofi/1
```

7.1.2 Mounting NTFS Volumes - 3rd party support

> ⓘ **NOTE:**
> For removable storage devices, first make sure your external disk drive is connected and powered on.
> To list attached removable storage devices: `rmformat -l`

Verify the pX partition number that contains the NTFS filesystem, typically “p1”, using fdisk. Even though it may seem counterintuitive, include the partition number “p0” as shown by rmformat in fdisk inquiries.

```
pfexec fdisk /dev/rdsk/c6t0d0p0
```

7.1.2.1 Additional software installation and configuration NTFS and FUSE support is provided by Tuxera’s NTFS-3G project that aims at providing a stable NTFS driver for several operating systems.

It is made possible thanks to Jean-Pierre André, on Openindiana page: http://jp-andre.pagesperso-orange.fr/openindiana-ntfs-3g.html Follow instructions on this page to install. Please install 64-bit package on 64-bit installations.

7.1.2.2 Mounting the NTFS filesystem Now mount the NTFS partition using:

```
pfexec ntfs-3g /dev/dsk/c6t0d0p1 /mnt/backup/
```

You can now also add a vfstab entry like so:

```
/dev/dsk/c6t0d0p1 /dev/rdsk/c6t0d0p1 /mnt/backup ntfs-3g - no -
```

7.2 Configuring OpenIndiana as an ISCSI Target Server - (COMSTAR)

< Place holder for content >

7.3 System backups

OpenIndiana offers several backup solutions. Here are just a few of them:

- Borg Backup
- Bacula
- Time-Slider
- rdiff-backup
- Rsync
- Zetaback
- ZFS exports
- cpio
- tar, zip, etc.

7.4 ZFS

ⓘ **NOTE:**
ITEMS TO WRITE ABOUT:
Gotcha's such as the following:

```
<e^ipi> don't suppose there's any solution to this:
<e^ipi> cannot replace 1509280528045021472 with
 ↪  /dev/dsk/c0t5000C5009204EB9Bd0s0: devices have different sector alignment
<tsoome> thats 512 versus non-512 sector issue
<tsoome> you need to build new pool based on larger sector
<tsoome> if its mirror, you can attach 512B disk to 4k pool, but not vice
 ↪  versa...
<e^ipi> well, damn.
<tsoome> that error message is too confusing, should be replaced by more clear
 ↪  one;)
<e^ipi> I swear this pool is already mix & match, freebsd complained about it
<e^ipi> (but still used it)
<tsoome> there is that thing that ashift is vdev property;)
<tsoome> not pool property (one reason why that linux zpool create ashift=
 ↪  option is bad)
<tsoome> or sort of bad anyhow
```

7.4.1 Importing ZFS disks

ⓘ **NOTE:**
ITEMS TO WRITE ABOUT:

- Talk about the ZFS import command.

7.4.2 How does one mirror their root zpool?

ⓘ **NOTE:**
ITEMS TO WRITE ABOUT:

- Adding a 2nd disk to the root pool

7.4.3 How does one create additional zpools?

ⓘ **NOTE:**
ITEMS TO WRITE ABOUT:

- zpool create command
  - Mirrors
  - Raidz

7.4.4 Modifying zpool settings and attributes

> ⓘ **NOTE:**
> ITEMS TO WRITE ABOUT:
> - zpool get/set commands

7.4.5 Modifying zfs file system settings and attributes

> ⓘ **NOTE:**
> ITEMS TO WRITE ABOUT:
> - zfs get/set commands

7.4.6 How does one create additional zfs datasets?

> ⓘ **NOTE:**
> ITEMS TO WRITE ABOUT:
> - zfs create command

7.4.7 Configuring system swap

> ⓘ **NOTE:**
> ITEMS TO WRITE ABOUT:
> - zfs set command
> - swap -l

8 Virtualization

< Place holder >

8.1 OpenIndiana as a virtualization host server

> ⓘ **NOTE:**
> ITEMS TO WRITE ABOUT:
> - Qemu-KVM (KVM) walkthrough
>   - illumos KVM port does not support AMD processors.
>   - Intel processors require EPT support.
> - Virtualbox walkthrough
>   - There is no package for this yet, but folks do have it working, see the wiki for details.

ⓘ **NOTE:**
ITEMS TO WRITE ABOUT:
In a nutshell, most modern Intel processors such as i3, i5, i7, and Xeon provide EPT support. Most older processors such as Core2duo and Core2Quad lack EPT support, and a few of them lack virtualization support at all. You can check your processor for EPT support via the following link: http://ark.intel.com/Products/VirtualizationTechnology

9 Localization

ⓘ **NOTE:**
ITEMS TO WRITE ABOUT:
Possible resources to help write this section:

- https://wiki.openindiana.org/oi/4.4+Localization
- https://docs.oracle.com/cd/E23824_01/html/E26033/glmen.html

10 Dtrace

< Place Holder >

11 Configuring Networking

11.1 Automatic Configuration (NWAM)

During installation of OpenIndiana, there are three options to configure networking : Automatic, Manual (static IP), and None.

Option Autamatic enables NWAM and configures all interfaces automatically.

Network Auto-Magic (NWAM) manages network interfaces as they are dynamically added or removed to or from the system, using network profiles, and is able to change network settings on the fly.

NWAM is suitable for servers, laptops, desktops and workstations alike, to automatically configure all wired or wireless network interfaces.

Option Manual disables NWAM and creates a /etc/sysding.conf file to setup static IP addresses.

Option None can be used to configure networking manually, without NWAM, after installation.

11.2 Changing from NWAM to Manual Configuration

If during install you enabled NWAM, and if you want to disable NWAM you can proceed as follows :

```
# svcadm disable physical:nwam
```

Define your IP/hostname in `/etc/hosts`. For example:

```
192.168.1.22 hostname hostname.local localhost loghost
# Subsittude 192.168.1.22 for YOUR IP
```

Enable the default physical service with `svcadm` and configure the `interface`:

```
# svcadm enable physical:default
```

There are multiple ways to configure the interface : using sysding or directly with ipadm.

For more information on sysding, see the section on Configuring and Tuning.

To configure an interface with ipadm:

```
# ipadm create-addr -T static -a local=192.168.1.22/24 bge0/v4static
```

or

```
# ipadm create-addr -T dhcp bge0/dhclient
```

The previous example sets up DHCP without using NWAM. NWAM is more than DHCP, because NWAM automatically configures any new interfaces, while the above way to manually setup DHCP is fixed for a specific interface.

If you do not know what the interface name is (bge0 in this case); then type in

```
$ dladm show-link
```

or:

```
$ kstat -c net | grep net

# look for hme0, bge0, e1000g0 or soemthing that resembles the driver in use.
```

Add gateway

```
# route -p add default 192.168.1.121
```

or

```
# nano /etc/defaultrouter

# Enter in your gateways IP
```

or use the /etc/sysding.conf file to configure a default router and DNS servers.

Set DNS server(s)

```
# nano /etc/resolv.conf

# Enter in the DNS server IP(s)
nameserver 192.168.1.121
```

or

```
# echo "nameserver 192.168.1.121" >> /etc/resolv.conf
```

Restart

```
# reboot
```

> ⓘ **NOTE:**
> IF you cannot ping an external IP (e.g. google.com) run this command and try again.

```
# cp /etc/nsswitch.dns /etc/nsswitch.conf
```

credit for this section of the docs go to /u/127b

11.3 More on Automatic Configuration (NWAM)

The following is a more in depth discussion of NWAM, which normally works without any user interaction, but NWAM can be customized using profiles by the user.

11.3.1 Using NWAM configuration tools

Usually NWAM configuration is done via GUI `nwam-manager` applet and `nwam-manager-properties` program.

From CLI all configuration can be done using two tools, `nwamcfg` to configure network profiles and `nwamadm` to manage them.

On its backend, NWAM relies on `nwamd`, the NWAM policy engine daemon and `netcfgd`, the NWAM repository daemon.

Access to the command line and GUI tools is controlled via security profiles ‘Network Autoconf Admin’ and ‘Network Autoconf user’.

To create NWAM profile use `nwamcfg` tool.

```
# nwamcfg
nwamcfg> create ncp Abroad
nwamcfg:ncp:Abroad> end
```

Now you can see a new profile in the list of NCPs:

```
nwamcfg> list
NCPs:
Abroad
Automatic
Locations:
Automatic
NoNet
User
nwamcfg>
```

The network interface to be managed under the new profile can be set and configured with `create ncu` command:

```
# nwamcfg
nwamcfg> select ncp Abroad
nwamcfg:ncp:Abroad> create ncu phys e1000g0
Created ncu 'e1000g0'. Walking properties ...
```

```
activation-mode (manual) [manual|prioritized]> prioritized
enabled (true) [true|false]>
priority-group> 0
priority-mode [exclusive|shared|all]> exclusive
link-mac-addr>
link-autopush>
link-mtu>
nwamcfg:ncp:Abroad:ncu:e1000g0>
```

Most important here is the `activation-mode`, which states if the profile is to be automatically set based on certain policies or if it is to be manually set using nwamadm. The latter should be most likely the case in a server environment. The `priority-group` and `priority-mode` here are set to (0 / exclusive). This says that the profile is for wired access and will always be the only active one at a time.

If after listing the changes you are satisfied with the configuration, permanently store it using `commit` command:

```
nwamcfg:ncp:Abroad:ncu:e1000g0> list
ncu:e1000g0
 type link
 class phys
 parent "Abroad"
 activation-mode prioritized
 enabled true
 priority-group 0
 priority-mode exclusive
nwamcfg:ncp:Abroad:ncu:e1000g0> commit
Committed changes
nwamcfg:ncp:Abroad:ncu:e1000g0> end
```

After an interface has been assigned to the profile, its IP configuration has to be defined:

```
nwamcfg:ncp:Abroad> create ncu ip e1000g0
Created ncu 'e1000g0'. Walking properties ...
enabled (true) [true|false]>
ip-version (ipv4,ipv6) [ipv4|ipv6]> ipv4
ipv4-addrsrc (dhcp) [dhcp|static]> static
ipv4-addr> 192.168.100.100
ipv4-default-route> 192.168.100.1
nwamcfg:ncp:Abroad:ncu:e1000g0> list
ncu:e1000g0
 type interface
 class ip
 parent "Abroad"
 enabled true
 ip-version ipv4
 ipv4-addrsrc static
 ipv4-addr "192.168.100.100"
 ipv4-default-route "192.168.100.1"
 ipv6-addrsrc dhcp,autoconf
nwamcfg:ncp:Abroad:ncu:e1000g0>
```

As you can see, the profile uses a static IP address and, for simplicity reasons, only provides IPv4 networking. Again, if you're happy with the results, commit them.

```
nwamcfg:ncp:Abroad:ncu:e1000g0> commit
Committed changes
nwamcfg:ncp:Abroad:ncu:e1000g0>
```

As it stands, now you have defined a pysical and ip layer attached to a network profile:

```
nwamcfg:ncp:Abroad:ncu:e1000g0> end
nwamcfg:ncp:Abroad> list
NCUs:
 phys e1000g0
 ip e1000g0
nwamcfg:ncp:Abroad>exit
```

To activate the profile, you use nwamadm. First of all, check for the current situation:

```
# nwamadm list
TYPE     PROFILE    STATE
ncp      Abroad     disabled
ncp      Automatic  online
ncu:phys e1000g0    online
ncu:ip   e1000g0    online
loc      Automatic  online
loc      NoNet      offline
loc      User       disabled
```

As you can see, Automatic is the active profile while Abroad is currently disabled. We can change that easily, but be aware that you might lock yourself out if you are connected via SSH when changing the profile!

```
# nwamadm enable -p ncp Abroad
Enabling ncp 'Abroad'
```

To check if the change has worked, you can use nwam again, and also ifconfig should show that interface is up:

```
# nwamadm list
TYPE PROFILE STATE
ncp Abroad online
ncu:phys e1000g0 online
ncu:ip e1000g0 online
ncp Automatic disabled
loc Automatic online
loc NoNet offline
loc User disabled
```

```
# ifconfig e1000g0
e1000g0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 8
inet 192.168.100.100 netmask ffffff00 broadcast 192.168.100.255
ether 0:c:29:56:46:73
```

If you want to revert to default `Automatic` ncp, use the following command:

```
# nwamadm enable -p ncp Automatic
```

Although it is possible to directly modify profiles using an editor, it is not advisable and will be hardly necessary, anyway. One of the coolest features of NWAM tools is that they can be completely scripted. All steps above could also be put in one line:

```
# nwamcfg "create ncp Abroad;create ncu phys e1000g0;set activation-mode=manual;set
 ↪  enabled=true;set priority-group=0;set priority-mode=exclusive;end;create ncu ip
 ↪  e1000g0;set enabled=true;set ip-version=ipv4;set ipv4-addrsrc=static;set
 ↪  ipv4-addr=192.168.100.100;set ipv4-default-route=192.168.100.1;commit"
```

Or, more nicely formatted, commented and stored in a file:

```
#
# Profile for use on the road
#
# Created on 13/01/2013 by S. Mueller-Wilken
#
create ncp Abroad

# Create physical interface definition for 1st network card
create ncu phys e1000g0
set activation-mode=manual
set enabled=true
set priority-group=0
set priority-mode=exclusive
end

# Create IP configuration for first network card
create ncu ip e1000g0
set enabled=true
set ip-version=ipv4
set ipv4-addrsrc=static
set ipv4-addr=192.168.100.100
set ipv4-default-route=192.168.100.1

# Commit the settings
commit
```

This file can then be read by nwamcfg:

```
# nwamcfg -f abroad.cfg
Configuration read.
```

While there is no longer a need to fiddle around in /etc/nwam, the configuration is still completely there, as can be easily verified:

```
# ls /etc/nwam
loc loc.conf ncp-Abroad.conf ncp-Automatic.conf
```

All configuration is placed in readable ASCII files so that configuration from a global zone is possible:

```
# cat /etc/nwam/ncp-Abroad.conf
```

```
link:e1000g0 type=uint64,0;class=uint64,0;parent=string,Abroad;enabled=boolean,tru
↪  e;activation-mode=uint64,4;priority-group=uint64,0;priority-mode=uint64,0;
interface:e1000g0 type=uint64,1;class=uint64,1;parent=string,Abroad;enabled=boolea
↪  n,true;ipv6-addrsrc=uint64,0,1;ip-version=uint64,4;ipv4-addrsrc=uint64,2;ipv4-
↪  default-route=string,192.168.100.1;ipv4-addr=string,192.168.100.100;
```

11.3.2 Network automagic online help

Comprehensive and fully illustrated online help for using NWAM is available by right clicking the NWAM tray icon and selecting *Help*. This opens help browser.

11.3.3 Troubleshooting NWAM

If NWAM is already configured and fails to connect to a wireless network try restarting the service.

For example:

```
# svcadm restart nwam
```

Sometimes the location gets set to *NoNet* and it's nessessary to manually change the location to *Automatic*.

When the location setting is configured to *Switch Locations Automatically*, it's not possible to change the location. This is resolved by reconfiguring the location to allow manual switching. To perform this task, do the following:

Right click the NWAM tray icon and select ***Location > Switch Locations Manually***. Right click the NWAM tray icon and select ***Location > Automatic***.

12 Clustering with Open HA Cluster

ITEMS TO WRITE ABOUT:

See old sun docs

- http://docs.oracle.com/cd/E19735-01/

Also see:

- http://zfs-create.blogspot.nl/