



# Oracle Linux and Oracle Solaris Administration

Haim Tzadok

# GRIGALE

[www.grigale.com](http://www.grigale.com)

Property of Grigale LTD. ©

# Safe Usage

The material provided is not free of errors and should be treated as such. If you find any error, please email me to [haim.tzadok@grigale.com](mailto:haim.tzadok@grigale.com) and I will try my best to correct any error.

Before using this material, always test and test again, before implementing ! The technical material may be subject to changes and should be treated as a recommendation only.

# Table Of Content

## Local administration

1. The Boot process
2. Services
3. User Management
4. Package Management
5. Process Management
6. Disk Management
7. LVM – Linux Volume Manager
8. Linux File-Systems
9. ZFS – Solaris Volume Manager and File-system
10. Backup & recovery
11. Rescue boot

# Table Of Content

## **Network administration**

1. Network Configuration
2. Name services
3. Network services
4. Creating Package repository
5. Apache
6. NFS
7. Samba
8. OpenLDAP
9. IPTables
10. FirewallD
11. Network Security tools
12. Performance Introduction

# Local Administration

## The boot process

- Boot process overview
- The boot process
- Grub
- Handling grub on Linux
- Handling grub on Solaris
- Summary

## User Management

- Users and groups configuration files
- Handling groups
- Handling users
- Login status
- The su command
- The sudo command
- Summary

## Package Management

- Packages on Linux
- Packages on Solaris
- Using yum on Linux
- Using pkg on Solaris
- Summary

## Process Management

- The process architecture
- Handling processes on Linux
- Handling processes on Solaris

# Local Administration

## Disk Management

- About disks and partitions
- Handling disks on Linux
- Handling disks on Solaris

## Linux Volume Manager (LVM)

- LVM topology
- Handling Physical volumes
- Handling Volume Groups
- Handling Logical Volumes

## Linux File-Systems

- File-System overview
- Creating file-system
- Mounting file-system – temporary
- Mounting file-system - permanently

## Solaris ZFS

- ZFS topology
- Handling pools
- Handling ZFS File-systems

## Backup and Recovery

- Backup on Linux
- Backup on Solaris

**GRIGALE**

Grigale Ltd T +972 3 938 1169 F +972 3 938 0070

**"The world's most popular open source database"** Rosh Ha'ayin industrial zone, Rosh Ha'ayin Israel

# The Boot Process

# The Boot Process

## Objectives

- Boot Process Overview
- The Boot Process
- GRUB
- GRUB version 1
- GRUB version 2
- Handling boot on Solaris
- Summary



# Boot Process Overview

<p><b>BIOS Operations</b> Power On <b>POST</b></p>
<p><b>First Stage Grub</b> Boot loader boot the first available boot-device using it's MBR. Loads file-system drivers</p>
<p><b>Second Stage Grub</b> Read Grub configuration file. Loads initial ram disk (init-rd), Then loads the Linux kernel and the init process.</p>
<p>Linux/Solaris Kernel</p>
<p>Kernel loads Init/Upstart/Systemd first process</p>

# GRUB

- **GRUB** - **GR**and **U**nified **B**oot Loader is a GNU project for boot loading Linux and other UNIX OS'es.
- Each grub line is a grub entry which can lead to a different OS or different OS kernels or settings.

```
Oracle Linux Server 7.4, with Linux 3.10.0-693.2.2.el7.x86_64
Oracle Linux Server 7.4, with Unbreakable Enterprise Kernel 4.1.12-103.3
Oracle Linux Server (4.1.12-94.3.9.el7uek.x86_64 with Unbreakable Enterp
Oracle Linux Server (3.10.0-693.el7.x86_64 with Linux) 7.4
Oracle Linux Server (0-rescue-97a29ec24a2042bbbb3ef279c544d91c with Linu
```

```
Use the ↑ and ↓ keys to change the selection.
Press 'e' to edit the selected item, or 'c' for a command prompt.
```

# GRUB version 1

- **Grub version 1** – First version of GRUB. Usually found on Oracle Linux/RHEL/CentOS 6 and lower.
- **/boot/grub/grub.conf** - This file contains grub version 1 configuration and is read at boot.
- **/etc/grub.conf** or **/boot/grub/menu.lst** - also known as grub version 1 configuration file other locations.
- Show GRUB entries:  
cat /etc/grub.conf
- Simply edit /etc/grub.conf to change grub behavior.

# GRUB version 1

E.g:

```
[root@olp1 ~]# cat /etc/grub.conf
```

```
#boot=/dev/sdb
```

```
default=0
```

```
timeout=5
```

```
splashimage=(hd0,0)/grub/splash.xpm.gz
```

```
hiddenmenu
```

0:

```
title Oracle Linux Server Unbreakable Enterprise Kernel (3.8.13-118.14.2.el6uek.x86_64.debug)
```

```
root (hd0,0)
```

```
kernel /vmlinuz-3.8.13-118.14.2.el6uek.x86_64.debug ro root=/dev/mapper/vg_olp1-lv_root rd_NO_LUKS  
rd_LVM_LV=vg_olp1/lv_root LANG=en_US.UTF-8 rd_NO_MD SYSFONT=latarcyrheb-sun16 rd_LVM_LV=vg_olp1/lv_swap  
KEYBOARDTYPE=pc KEYTABLE=us rd_NO_DM rhgb quiet
```

```
initrd /initramfs-3.8.13-118.14.2.el6uek.x86_64.debug.img
```

1:

```
title Oracle Linux Server Red Hat Compatible Kernel (2.6.32-642.11.1.el6.x86_64)
```

```
root (hd0,0)
```

```
kernel /vmlinuz-2.6.32-642.11.1.el6.x86_64 ro root=/dev/mapper/vg_olp1-lv_root rd_NO_LUKS  
rd_LVM_LV=vg_olp1/lv_root LANG=en_US.UTF-8 rd_NO_MD SYSFONT=latarcyrheb-sun16 rd_LVM_LV=vg_olp1/lv_swap  
KEYBOARDTYPE=pc KEYTABLE=us rd_NO_DM rhgb quiet crashkernel=auto
```

```
initrd /initramfs-2.6.32-642.11.1.el6.x86_64.img
```

2:

```
title Oracle Linux Server Unbreakable Enterprise Kernel (3.8.13-55.1.2.el6uek.x86_64.debug)
```

```
root (hd0,0)
```

```
kernel /vmlinuz-3.8.13-55.1.2.el6uek.x86_64.debug ro root=/dev/mapper/vg_olp1-lv_root rd_NO_LUKS  
rd_LVM_LV=vg_olp1/lv_root LANG=en_US.UTF-8 rd_NO_MD SYSFONT=latarcyrheb-sun16 rd_LVM_LV=vg_olp1/lv_swap  
KEYBOARDTYPE=pc KEYTABLE=us rd_NO_DM rhgb quiet
```

```
initrd /initramfs-3.8.13-55.1.2.el6uek.x86_64.debug.img
```

# GRUB version 2

- **Grub version 2** – Second version of GRUB. Usually found on Oracle Linux/RHEL/CentOS 7 +
- **/boot/grub2/grub.cfg** - This file locations contains grub version 2 configuration and is read at boot.
- **/etc/grub2.cfg** – another location for grub2 configuration file.
- Show GRUB entries:

```
awk -F\' '$1=="menuentry " {print i++ " : " $2}' /etc/grub2.cfg
```

**E.G:**

```
[root@ol174vm1 ~]# awk -F\' '$1=="menuentry " {print i++ " : " $2}' /etc/grub2.cfg
0 : Oracle Linux Server 7.4, with Linux 3.10.0-693.2.2.el7.x86_64
1 : Oracle Linux Server 7.4, with Unbreakable Enterprise Kernel 4.1.12-103.3.8.el7uek.x86_64
2 : Oracle Linux Server (4.1.12-94.3.9.el7uek.x86_64 with Unbreakable Enterprise Kernel) 7.4
3 : Oracle Linux Server (3.10.0-693.el7.x86_64 with Linux) 7.4
4 : Oracle Linux Server (0-rescue-97a29ec24a2042bbbb3ef279c544d91c with Linux) 7.4
```

# GRUB version 2

- **Useful files and commands:**
- `awk -F\ ' '$1=="menuentry " {print i++ " : " $2}' /etc/grub2.cfg` - shows available entries.
- `grub2-set-default 2` - set the default entry to be line entry number 2.
- `grub2-editenv list` - shows the default entry.
- `/etc/default/grub` – file that helps to configure grub behavior
- `grub2-mkconfig -o /boot/grub2/grub.cfg` – after changing `/etc/default/grub` you should issue this command to make changes effective.

# Solaris GRUB2 Administration

- **Useful files and commands:**
- **bootadm list-menu** - shows available entries and default and timeout values.
- **bootadm set-menu default=2** - set the default entry to be line entry number 2.
- **bootadm set-menu timeout=15** - set the timeout value to be 15 seconds.

# Services



# Services

## Objectives

- Solaris Run levels
- Linux Run Levels
- Old Sys-V services
- Solaris services – SMF services
- Linux Services

# Solaris Run Levels

- List of the available run levels in Solaris:
- **Run level 6** – same as performing the command: **reboot**
- **Run level 0** – OBP mode(BIOS mode)
- **Run level S/1** – starts the system in single user mode (administrative mode).
- **Run level 2** – starts the system with no network services.
- **Run level 3** – starts the system with network services enabled.
- **Run level 5** – Powered off
  
- `init <run-level-number>`
- E.g: **init 5** -> power off.

# Linux Run Levels

- List of the available run levels in Linux:
- **Run level 6** – same as performing the command: **reboot**
- **Run level 0** – Same as poweroff.
- **Run level S/1** – starts the system in single user mode (administrative mode).
- **Run level 2** – starts the system with no network services.
- **Run level 3** – starts the system with network services enabled.
- **Run level 5** – starts the system with Desktop GUI.
- Init <run-level-number>
- E.g: **init 5** -> full service !!!

# Old SYS-V Services

- **/etc/init.d/** - the directory where you store the service script.
- Each script in this directory can get one of the 2 arguments – [stop | start]
- e.g
- **/etc/init.d/nfsd [stop|start]**
- 
- The way to configure a service, to run or stop in a run level, is to link the file to be one of the 2 files in one of the run level directories.
- e.g:
- **In /etc/init.d/nfsd /etc/rc3.d/S88nfsd**
- **/etc/rc3.d/S88nfsd** – this script will start nfsd when entering level 3.
- 
- **In -s /etc/init.d/nfsd /etc/rc2.d/K88nfsd**
- **/etc/rc2.d/K88nfsd** – this script will start nfsd when entering level 2.

# Old Sys-V Services

- **/etc/rcS.d/** - the directory for storing Starter scripts when entering Level S from lower levels and Killer scripts when entering from upper levels.
- **/etc/rc1.d/** - the directory for storing Starter scripts when entering Level 1 from lower levels and Killer scripts when entering from upper levels.
- **/etc/rc2.d/** - the directory for storing Starter scripts when entering Level 2 from lower levels and Killer scripts when entering from upper levels.
- **/etc/rc3.d/** - the directory for storing Starter scripts when entering Level 3 from lower levels.

# Solaris SMF

Service Management Facility is a new method introduced in Solaris 10 to handle services in much more sophisticated way then using the Sys-V method.

- **svcs** – this commands lists services.
- **svcadm** – this command administer services.
- **svccfg** – this command manages service configuration.
- **svcs [-a]** – without anything svcs lists only enabled services.
- **-a** – list all services (also the disabled ones).

# Solaris SMF

svcs switch	Description
-p	Shows processes that are handled by the service.
-d	Shows which Services the service is depended on.
-D	Shows which services depends on me.
-v	Shows verbose information.
-x	Health information.
-l	Long listing.
-a	Shows all services - also the disabled services.

# Solaris SMF

**FMRI – a descriptive for service. For example: print/server**

- **svcadm enable <FMRI>** – enable the service named <FMRI>.
- **svcadm disable <FMRI>** – disable the service named <FMRI>.
- **Securing network services**
- Securing network services can be done using the command **netservices**.
- **netservices [limited | open]** –
- **limited** – will secures the Solaris system to enable only services that are secured by default.
- **open** – will open all services like telnet/ftp which are not secured by default.
- **inetconv** – checks /etc/inetd.conf file and converts inetd services to SMF services.



# Linux 6 Services

## Linux 5/6 services:

- **service <service-name> start|stop|status** – start or stop the service named service-name.
- **service --list-all**
- **/etc/init.d - <service-name>**

## Enabling/Disabling service on run-level

chkconfig – shows available services and their corresponding configuration.

chkconfig [--level <levels>] <name> <on|off|

e.g: chkconfig --level 35 httpd on

chkconfig --level 12 httpd off

# Linux 7 Services

## Linux 5/6 services:

- **service <service-name> start | stop | status** – start or stop the service named service-name.
- **service --list-all**
- **/etc/init.d - <service-name>**

## Linux 7 services:

- **systemctl start | stop | status <service-name>** – start | stop | the service named service-name.
- To see non-active services:

**systemctl | grep -v active**

# User Management

# User Management

## Objectives

- Users and groups configuration files
- Handling groups
- Handling users
- Login status
- The su command
- The sudo command
- The sudoers file
- Summary

# User and Group Configuration files

- **/etc/passwd** - This file contains user specific configuration files. This file stores all user information in the system except passwords.
- Sample entry: root:x:0:0:root:/root:/bin/bash
- **/etc/shadow** - This file contains user password in a hash format and password aging.
- **/etc/group** - This file contains group specific configuration file.
- Sample entry: dba:x:500:
- **/etc/skel** - This directory contains all skeleton files – files that will be added to each user that is added to the system.

# Handling Groups

- **groupadd** – The command to add a group to the system.
- **groupadd [-g GID] GROUP**
- E.g: `groupadd -g 500 dba`
  
- **groupdel** – The command to delete a group from the system.
- **groupdel GROUP**
- E.g: `groupdel dba`
  
- **TIPP** : normally it is recommended to create a group with a round number and then add users with the followed numbers.
- **!** : group name in Linux and Solaris is case sensitive ! The group Dev is not equal to the group dev.

# Handling Users

- **useradd** – The command to add a user to the system.
- **useradd [-g primary-GID] [-G secondary-GID] [-u UID] NAME**
- E.g: `useradd -g dba -u 501 oracle`
- **usermod** – The command to modify user credentials in the system.
- **userdel** – The command to delete a user from the system.
- **userdel [-r] NAME**
- E.g: `userdel oracle`
  
- **TIPP** : normally it is recommended to create a group with a round number and then add users with the followed numbers.
- **!** : user name in Linux and Solaris is case sensitive ! The user Haim is not equal to the user haim.

# Login Status

- **who** – command that shows all the logged in users in the system.
- **who [-a]** – show all users including dead processes time of last system boot, active processes spawned by init and other information.
- **who [-r]** – print run-level information.

**E.g:**

```
[root@olp1 ~]# who
root      pts/1          2018-02-26 19:56 (212.29.203.226)
root      pts/2          2018-02-26 19:56 (212.29.203.226)
```



# The su command

- **su [OPTION] [-] [USER [ARG]...]** – command to run a shell or command with a substitute user and group id's.
- With out any specific USER it will change to root.
- - or -l – make the shell a login shell, clear all environment variables except TERM.
- **su – haim** – switch to be user – haim, including making the shell a login shell. If root is initiating this command no password is required. If regular user is initiating this command haim's password is required.

E.g:

```
[root@olp1 ~]# su - haim  
[haim@olp1 ~]$
```

# The sudo command

- **sudo** – command that allows executing commands as another user. (normally as root) it requires to know only the password of the current user who is invoking the sudo command.
- **sudoedit** – command that allows editing files as another user. (normally as root)
- **/etc/sudoers** – sudo configuration file.
- **visudo** – command that allows editing safely the sudo configuration file - using vi text editor.

E.g:

**sudo ls** – runs ls as root

**sudo -u haim ls** – runs ls as user haim.

# The sudoers file

- **/etc/sudoers** – a file that contains the policy how to treat a user, when this user is invoking the sudo command.
- 90% of understanding the file is by understanding the following line:

User Host = (Runas) Command

- Read this as “User may run Command as the Runas user on Host”.

E.g:

haim ALL = (ALL) ALL

Allow the user haim to run any command as any user.

# Summary

	Create	Remove	Status
USER	useradd -g dba -u 501 oracle	userdel oracle	id [login][-a] – without anything shows user identity. With NAME - shows information about the user – login. -a – shows all groups for this user. who – shows who is logged in.
Group	groupadd -g 500 dba	groupdel dba	groups – shows all the groups the user is member.

# Package Management

# Package Management

## Objectives

- Packages on Linux
- Packages on Solaris
- Using yum on Linux
- Using pkg on Solaris
- Summary

# Packages on Linux

- **RPM-** RedHat Package Management is the standard package format for CentOS based Linuxes.
- **rpm -i** – install packages
- **rpm -e** – erase packages
- **rpm -q** – query on packages
- **rpm -qa** – shows all installed packages.
- **rpm -qa |wc -l** – count how many packages are installed.
- **rpm -qa |grep mysql** – check if mysql packages are installed.
- **rpm -qf /usr/sbin/ifconfig** – shows the package that contains the command ifconfig (which ifconfig – shows the full path of the command ifconfig)
- **rpm -ql net-tools** – shows all the files that the package net-tools contains.

# SYS-V Packages on Solaris

- **pkginfo | pkgadd | pkgrm** – Sys-V Package Management is the standard package format in Solaris 10.
- **pkgadd <package-name>** – add packages to the system.
- **pkgrm <package-name>** – remove packages from the system.
- **pkginfo** – shows installed packages.



# YUM – Package manager

- **YUM** - yum is a repository based package manager.
- Handle packages and install packages with dependencies.
- /etc/yum.repos.d – configuration directory for available repositories in the system.
- **yum repolist** – shows available repositories.
- **yum clean all** – clean all the yum repository cache.
- **yum install <package-name>** – install packages
- **Yum erase <package-name>** – erase packages
- **Yum list** – shows all installed packages.
- **yum search <keyword>** – search availability of a keyword in the package repositories.

# YUM – Package manager

- **yum localinstall <NAME.rpm>** – installs the package NAME.rpm and all of its dependencies.
- **yum check-update**– check if updates are needed on the system.
- **yum update**– updates the system with newly available packages.

# YUM –

## create a local dvd repository

- `/etc/yum.repos.d/` – configuration directory for available repositories in the system. Every repository is stored in a `*.repo` file.
- Clean `/etc/yum.repos.d` directory – `rm /etc/yum.repos.d/*`
- `vi /etc/yum.repos.d/dvd.repo`

- dvd.repo content:

```
[dvd]
```

```
name=DVD repository
```

```
Baseurl=file:///mnt
```

```
gpgcheck=1
```

```
enabled=1
```

# Process Management

# Process Management

## Objectives

- The process architecture
- Handling processes Linux/Solaris
- Handling processes on Linux
- Handling processes on Solaris

# Process Architecture

- Every program that runs in the system runs as a process.
- The Linux kernel is responsible for managing all the processes in the system and is doing this by using a process table for storing all the available processes.
- Other than using a process table, another way to represent all processes in the system is by using a file-system called - /proc the processes file-system or – procfs.

# Process Architecture

- Every process has credentials

Process ID–PID
PPID – Parent Process ID
Owner
GROUP
PRIORITY
Scheduling Class

## Process hierarchy

- Every process in the system is a child of another process and initially inherit from it's father - it's credentials and attributes.
- After creation of a process using the system call - fork, you can change the credentials and attributes of a process using exec and other available system calls.

# Handling Processes – Linux/Solaris

- The way to handle processes is by signaling a process.
- There are more than 40-70 ways to signal a process.
- A way to see available signals is by using the command kill by using:
- **kill -l**



# Handling Processes – Linux/Solaris

Signal Name	Signal ID	Description
KILL	kill -9 <PID>	Brutal killing of a process
TERM	kill -15 <PID>	Terminate signal (less brutal)
HUP	kill -1 <PID>	Hang up
SIGFPE	kill -8 <PID>	Floating point exception signal
SIGCONT	kill -25 <PID>	Continue
SIGSTOP	kill -23 <PID>	Stop

# Handling Processes – Linux

Command	Description
ps, jobs	Show process, show jobs
top	Shows online view of running processes
pkill [-9] <string>	Brutal killing of a process that has a name that contains the string - <string>
pstop <PID>	Send the STOP signal
prun <PID>	Send the CONT signal
pmap	Prints the process address space
pstack	Prints the process stack
pargs	Print the available arguments that the process started with.
pfiles	Shows which files are open and their credentials.

# Handling Processes – Solaris

Command	Description
prstat/top	Shows online view of running processes
pkill [-9] <string>	Brutal killing of a process that has a name that contains the string - <string>
pstop <PID>	Send the STOP signal
prun <PID>	Send the CONT signal
pmap	Prints the process address space
pstack	Prints the process stack
Pargs	Print the available arguments that the process started with.
pfiles	Shows which files are open and their credentials.
ptree	Show process hierarchy in a tree format

# Disk Management

# Disk Management

## Objectives

- About disks and partitions
- Handling disks on Linux
- Handling disks on Solaris

# About Disks and Partitions

- Disk – consists of a 2 devices –
  - **raw/character device** crwxrwx /dev/rdisk/cXtXdX
  - **block device.** brwxrwx /dev/dsk/cXtXdX
- **Raw/character device** - a device that has no file-system
- **Block device** - a device that represents a file-system. /dev/sdX X – is an alphabetical letter.
- Disk is a collection of sectors
- Sector = 512 bytes = ½ KB.

# Handling Disks on Linux

- Handling Disks on Linux is done by fdisk command.
- **fdisk** /dev/sda
- fdisk -l – shows all the disks in the system.

Command (m for help): m

- a toggle a bootable flag
- c toggle the dos compatibility flag
- **d delete a partition**
- l list known partition types
- m print this menu
- **n add a new partition**
- **p print the partition table**
- q quit without saving **changes**
- u change display/entry units
- v verify the partition table
- w write table to disk and exit
- x extra functionality (**experts only**)

# Handling Disks on Solaris

- Handling Disks on Solaris is done by format command.
- **format** -> [n]choose disk number:n ->[p]artition -> [p]rint



# Handling Disks on Solaris

- Handling Disks on Solaris is done by format command.

- FORMAT MENU:

- disk - select a disk
- type - select (define) a disk type
- partition - select (define) a partition table
- current - describe the current disk
- format - format and analyze the disk
- fdisk - run the fdisk program
- repair - repair a defective sector
- label - write label to the disk
- analyze - surface analysis
- defect - defect list management
- backup - search for backup labels
- verify - read and display labels
- inquiry - show disk ID
- volname - set 8-character volume name
- !<cmd> - execute <cmd>, then return
- quit

# LVM

# Linux Volume Manager

# LVM

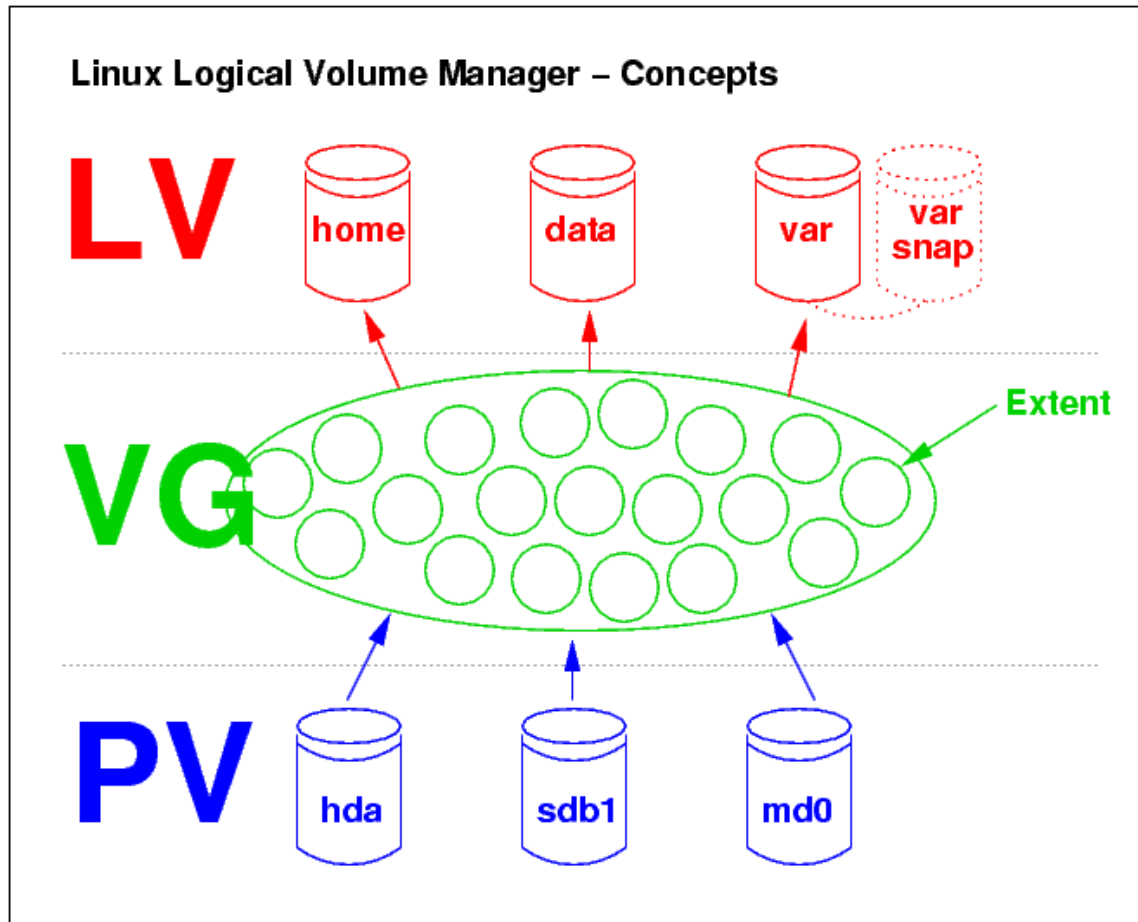
## Objectives

- LVM topology
- Handling Physical volumes
- Handling Volume Groups
- Handling Logical Volumes

# LVM Topology

- LVM Topology
- Physical Volume – PVs (pvcreate)
- Volume Groups – VGs
- Logical Volumes - LVs

# LVM Topology



# Handling Physical Volumes

- **pvs** – list pv's
- **pvcreate** – creates physical volumes
- **pvremove** – removes a physical volume.
  
- Examples:
- **pvcreate /dev/sdb /dev/sdc** – this will create physical volumes from sdb and sdc.
- **pvremove /dev/sdb** – this will remove sdb from lvm.

# Handling Volume Groups

- vgs – list pv's
- vgcreate – creates physical volumes
- vgremove – removes a physical volume.
  
- vgcreate VG PV
- Examples:
- **vgcreate datavg /dev/sdb /dev/sdc** – this will create a new datavg and will add sdb and sdc to be members in the datavg volume group.

# Handling Logical Volumes

- `lvs` – list pv's
- `lvcreate` – creates physical volumes
- `lvremove` – removes a physical volume.
  
- Examples:
- **`lvcreate -n dblv datavg -L 2g`** – This will create a new logical volume with name: `dbl` that is in `datavg` volume group, that has a size of 2GB.
- `lvremove VGname LVname`



# Linux File Systems

# Linux File-Systems

## Objectives

- File-System overview
- Creating file-system
- Mounting file-system – temporary
- Mounting file-system - permanently

# File-Systems overview

- File-system is a collection of:
- Inodes – an object that contain the metadata of a file or directory.
- Blocks – an object that contain the actual data of a file or a directory.
- Another features of file-system may be: logging, snapshots, COW, compression, deduplication, self healing and many others.

# File-Systems overview

- **Logging** – logging is a mechanism that a file-system is using to avoid data corruption. The way it is done is by writing all the activities of the file-system to a different place (before performing them). After the file-system finishes doing an activity, it removes the activity from the logging place.
- **Compression** – compression is a feature that may be done in the block level. By doing compression, the file-system may reduce the capacity of the data that will be written to disk, which may save a lot of space.
- **Deduplication** – de-duplication is a feature that may be done in the block level. The file-system perform a strong checksum to each block and if it finds that two blocks has the same checksum it doesn't write the same block again.
- **COW** – Copy on Write is a mechanism to avoid corruption.  
COW is done every time that the file-system needs to update an existing data. it will not perform the update on the same block, but it will perform the update on a new block that will be referenced after it finishes the update.

# Creating File-System

- Creating file-system is done by the command `mkfs`
- `mkfs.ext4 /dev/sda1` (physical disk) 5-7.3
- **`mkfs.xfs /dev/mapper/datavg-dblv`** (logical volume) 7.4+
- When working with logical devices the device semantics is:
- `/dev/mapper/Vgname-Lvname`
- For example: `/dev/mapper/datavg-backuplv`

# Mounting File-System - temporary

- Mounting file-system is done by the command `mount`
- `mkdir /db`
- `mount /dev/mapper/datavg-dblv /db`

# Mounting File-System - permanently

- Mounting file-system permanently is done by the adding a new line to `/etc/fstab`
- Edit `/etc/fstab` and add the following line to the end of the file.

```
/dev/mapper/datavg-dblv /db xfs defaults 0 0
```

- After editing the file issue the command: **mount /db**
- You may now see `/db` in `df -h`.

# Unmounting File-System

- Unmounting a file-system is done by the command `umount`
- `umount /db`
- `rmdir /db`



# Summary

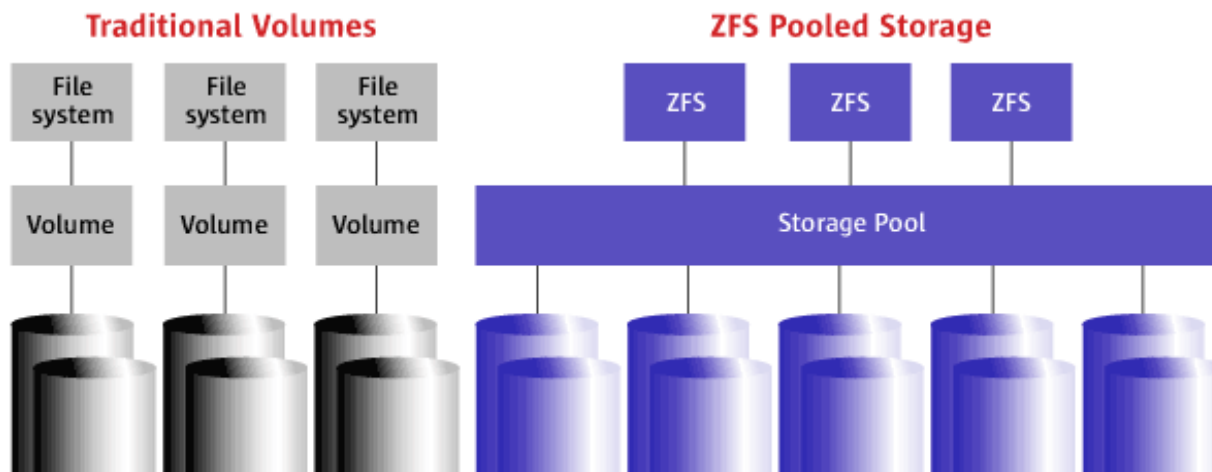
- VirtualBox – Create disks when vm is powered off. (Settings - > Storage - > SATA Controller -> Add Hard Disk
- Poweron VM
- `fdisk -l | grep Disk` (now we need to search for the new hard disks)
- `pvcreate /dev/sdb /dev/sdc`
- `vgcreate datavg /dev/sdb /dev/sdc`
- `lvcreate -n backuplv datavg -L 20g`
- `mkfs.xfs /dev/mapper/datavg-backuplv`
- `mkdir /backup`
- ~~`mount /dev/mapper/datavg-backuplv /backup`~~
- Edit `/etc/fstab` and add a new line at the end:  
`/dev/mapper/datavg-backuplv /backup xfs defaults 0 0`
- `mount /backup`

# ZFS File-system

## Objectives

- ZFS topology
- ZFS features
- Self healing
- Handling pools
- Handling ZFS File-systems

# ZFS topology



ZFS is a new revolutionary way to administer file-systems. With features and benefits currently not found in any other Volume manager or file-system.

ZFS treats storage the same as our Operating system administer physical memory (RAM).

Users should not need to know of the underlying structure while requesting or working on Storage. ZFS eliminates the need for volume management.

No need to create virtualized volumes and then file-systems. ZFS aggregates RAID structures into a pool by thus allowing simplicity and reduced complexity.

# ZFS features

## ZFS characteristics

- Transactional object file-system
  - Copy On write
  - No File-system Check
- Self Healing
  - Checksum
  - End-To-End Data integrity
  - Self healing for replicated data
- ZetaByte File-system – 128 bit support
- Simplified Administration
- Universal use: Use Swap, NFS, CIFS (Solaris express/OpenSolaris), iSCSI, Lun provisioning ( Solaris Express/OpenSolaris via COMSTAR)
- Vast number of features: quota, compression, dedup, snapshots, checksum, NFS, CIFS, integrated backup/restore, send and receive.

# ZFS terms and definitions

- **Terms and definitions**

- **Pool** – a collection of disks or partitions in a RAID format.

The pool is the place where we can create file-systems and volumes.

- **File-system** – a file-system in ZFS is simply a special directory with administration point.

A File-system in zfs does not have a size. It is a special directory that can be managed to have: quotas, snapshots, user-quotas, clones, compression, de-duplication and more...

- **Volume** – a special file that contains a chunk of space without a file-system.

In case we want a chunk of space in the pool without a file-system, a volume is the object we will want to use. Volume is also a special object that can be managed to have: quotas, snapshots, clones, compression, deduplication and more...

# ZFS terms and definitions

- **Basic Terms and Definitions**

- 
- **Pool** (Storage pool) – a logical group of devices or virtual devices that consist the pool.
- **Virtual device** – a logical device in the pool that consists of special structure of devices, such as mirror, raidz1 or raidz2 structures.
- **Device** – any device in ZFS can be a physical device or a file (built up from the mkfile command)
- **Dataset** – a generic name for ZFS entities like: file-systems, snapshots, clones or volumes
- **File system** – a dataset that contains a standard POSIX file-system
- **checksum** – A 256 bit hash of the data in a ZFS file-system block. Can range from the simple and fast fletcher2 (the default) to SHA256
- **Snapshot** – A read-only image of a file-system or volume at a given point of time.
- **Clone** – A file-system whose initial contents are identical to the contents of a snapshot
- **Resilvering** – the process of transferring data from one device to another device

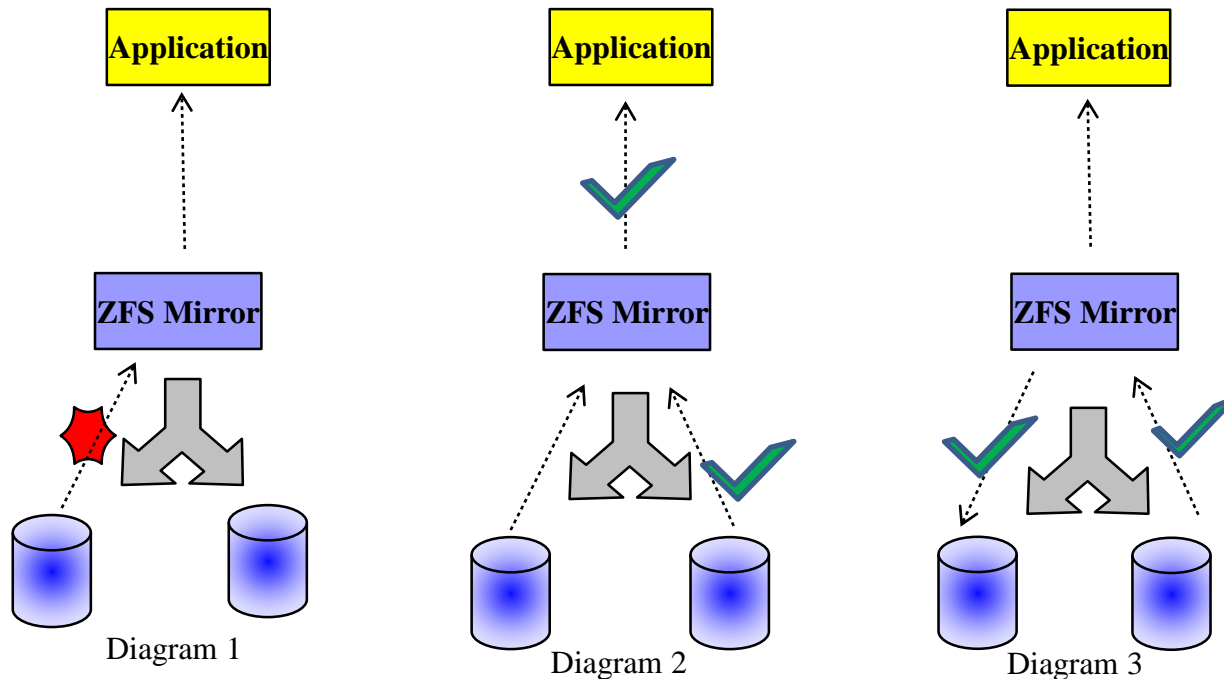
# Self healing

- In ZFS all data and metadata is checksummed by a user defined algorithm (the default is fletcher2). All checksumming and data recovery is done by file-system layer, and is transparent to the application or the user.

ZFS validates the entire I/O path eliminating the following errors:

- Bit rot
- Phantom writes
- Misdirected reads and writes
- DMA parity errors
- Driver bugs
- Accidental overwrites

# Self healing



Application issues a read for data block.

Diagram 1: Checksum reveals that the block is corrupted. ZFS doesn't send the data to the application.

Diagram 2: ZFS tries to get a good copy from an alternative data copy.

Diagram 3: ZFS returns the good copy to the application and repairs the damaged block.



# Handling ZFS pools

- Listing pool information
  - `zpool list` - lists all available imported pools
- Viewing pool statistics
  - `zpool iostat <poolName>`
- Viewing pool health
  - `zpool status <poolName>`
- Viewing pool properties
  - `zpool get all <poolName>`

# Handling ZFS pools

- **Creating pools**

- **Striped pool**

- `zpool create <poolName> disk1 disk2`

- **Mirrored pool**

- `zpool create <poolName> mirror disk1 disk2`

- **Raid-Z pool (RAID5)**

- `zpool create <poolName> raidz disk1 disk2 disk3`

- **RAID-Z2 pool (RAID6)**

- `zpool create <poolName> raidz2 disk1 disk2 disk3 disk4`

- **Destroying pools**

- `zpool destroy <poolName>`

# Handling ZFS pools

- **Exporting pools**

- `zpool export <poolName>`

- **Importing pools**

- `zpool import` – listing all available pools that can be imported
- `zpool import -d` – listing all available pools that can be imported included destroyed.
- `zpool import <poolName>`
- `zpool import <poolName> <NewpoolName>`

# Handling ZFS File-systems

- **Creating ZFS file-system**

- `zfs create <poolName>/<fileName>`
- Example: `zfs create data/backup`

- **Renaming a ZFS file-system**

- `zfs rename <poolName>/<fileName> <poolName>/<fileName>`

- **Destroying ZFS file-system**

- `zfs destroy <poolName >/<fileName>`

- **Setting compression**

- `zfs set compression=on /data/backup`

- **Setting deduplication**

- `zfs set dedup=on /data/backup`



# Thank You!

**GRIGALE**  
www.grigale.com

Grigale Ltd T +972 3 938 1169 F +972 3 938 0070  
Lev Ha'aretz industrial zone, Rosh Ha'ayin Israel