```
sudo apt update
sudo apt install zfsutils-linux gdisk -y

Pool Creation:


sudo fdisk -l
// You will find 4-extra disk besides OS-disk //

// Initialize all extra disks //
sudo sgdisk --zap-all /dev/xvdb
sudo sgdisk --zap-all /dev/xvdc
sudo sgdisk --zap-all /dev/xvde
sudo sgdisk --zap-all /dev/xvdf


// raid 1 //
sudo zpool create -o ashift=12 -f vol1 mirror /dev/xvdb /dev/xvdc
sudo zpool status           |||||| display pool status
sudo zpool list             |||||| See pool list // Multiple pool can be created in a single box


// Expand a raid volume //
sudo zpool add vol1 mirror /dev/xvde /dev/xvdf



sudo zpool destroy vol1
sudo zpool create -o ashift=12 -f vol1 mirror /dev/xvdb /dev/xvdc

// Add SSD Read cache | L2ARC //
sudo zpool add -f vol1 cache /dev/xvde                    // Single Read Cache Disk
sudo zpool remove vol1 /dev/xvde                          // Remove Read Cache Disk
sudo zpool add -f vol1 cache /dev/xvde /dev/xvdf          // Multiple Read Cache Disks


// Add SSD Write cache | ZIL //
```

```
sudo zpool remove vol1 /dev/xvde  /dev/xvdf                 // Remove Read Cache Disk
sudo zpool add -f vol1 log /dev/xvde                        // Add SSD Write cache ZIL
sudo zpool remove vol1 /dev/xvde                            // Remove Read Cache Disk
sudo zpool add -f vol1 log mirror /dev/xvde /dev/xvdf       // It is strongly recommended to keep write-
cache on Mirror disks

sudo zpool destroy vol1                                     // Destroy all pools

sudo zpool status


// Create Final Lab Storage Pool //
sudo zpool create -o ashift=12 -f groupXvol1 raidz /dev/xvdb /dev/xvdc /dev/xvde

// Replace X with your group number

// ********** File System/Dataset Manipulation *************** //
sudo zfs list
sudo zfs careate groupXvol1/gXsubvol1
sudo zfs set compression=on groupXvol1
sudo zfs set atime=off groupXvol1
sudo zfs set autoexpand=on groupXvol1
sudo zfs set quota=4G groupXvol1/gXsubvol1
sudo zfs set reservation=4G groupXvol1/gXsubvol1                // Thik Provisoning //

df -h
touch  file1.txt file2.txt file3.txt /groupXvol1/gXsubvol1/
ls -la /groupXvol1/gXsubvol1/


sudo zfs careate groupXvol1/gXsubvol2
sudo zfs unmount /groupXvol1/gXsubvol2
mkdir /mnt/mysql
sudo zfs set mountpoint=/mnt/mysql groupXvol1/gXsubvol2
sudo zfs mount groupXvol1/gXsubvol2
sudo zfs set mountpoint=/mnt/data groupXvol1/gXsubvol2
```

```
// Snapshot & Snapshot manipulation //
sudo zfs create groupXvol1/gXsubvol1
sudo zfs snapshot groupXvol1/gXsubvol1@version1
sudo zfs snapshot groupXvol1/gXsubvol1@`date +%Y%m%d-%H%M%S`
sudo zfs list -t snapshot |grep gXsubvol1

date_time=`date +%Y%m%d-%H%M%S`
sudo zfs snapshot -r groupXvol1/gXsubvol1@$date_time
sudo zfs rollback -r groupXvol1/gXsubvol1@version1
sudo zfs destroy -r groupXvol1/gXsubvol1@version1

// Destroy all previous snapshot //
sudo zfs list -rt snapshot -o name |grep groupXvol1/gXsubvol1@@ |sort |head -n -0 |xargs -n 1 zfs destroy -r


// Handling Disk Failure //

|||||||||| Manualy Fail a disk  (lab instructor will do)

syntax: zpool replace [poolname] [old drive id] [new disk drive]

// Replace old/damaged disks //
zpool status                // find the failed disk

 state: DEGRADED
status: One or more devices could not be used because the label is missing or
invalid.  Sufficient replicas exist for the pool to continue
functioning in a degraded state.
action: Replace the device using 'zpool replace'.
   see: http://zfsonlinux.org/msg/ZFS-8000-4J
  scan: none requested
config:

NAME                      STATE     READ WRITE CKSUM
groupXvol1                DEGRADED     0     0     0
  raidz1-0                DEGRADED     0     0     0
    xvdb                  ONLINE       0     0     0
    xvdc                  ONLINE       0     0     0
```

```
    16626827190568113194  UNAVAIL      0     0     0  was /dev/xvde1

errors: No known data errors

|||||||||| Add a new disk (lab instructor will do)
|||||||||| Initialize New Disk  (participant will do)
sudo sgdisk --zap-all /dev/xvdf


zpool replace groupXvol1 16626827190568113194 /dev/xvdf
watch zpool status
zpool clear groupXvol1
zpool status

# Ref-1: https://pthree.org/2012/04/17/install-zfs-on-debian-gnulinux/
# Ref-2: https://www.sotechdesign.com.au/zfs/
# Ref-3: https://wp.strahlert.net/wordpress/zfs-2/expanding-zpool/

 // ZFS Export Import Data Volumes //

// Export:
zfs list -t snapshot |grep gXsubvol1
zfs send groupXvol1/gXsubvol1@versionX > /root/gXsubvol1.zfs


// Import:
zfs destroy -r groupXvol1/gXsubvol1
zfs recv groupXvol1/gXsubvol1@versionX < /root/gXsubvol1.zfs


// Clone For Data Recovery:
zfs clone groupXvol1/gXsubvol1@versionN groupXvol1/gXsubvol1-Clone


 // ZFS Snapshot and Replication //
```

```
cd /root/
ssh-keygen -t rsa                    // Press all Enter //
ssh-copy-id -p 1122 -i /root/.ssh/id_rsa.pub 'root@172.16.108.71'

; test connection
ssh root@172.16.108.71 -p 1122 ; // this should not ask any password



# Step-1:
sudo zfs create groupXvol1/gXsysconfig
cd /root/
date_time=`date +%Y%m%d-%H%M%S`;
zfs snapshot -r groupXvol1/gXsubvol1@$date_time;
zfs send -R groupXvol1/gXsubvol1$date_time| ssh root@172.16.108.71 "zfs receive -dvF groupYvol1"
echo $date_time > /groupXvol1/gXsysconfig/gXsubvol1_lsnapid;

# Step-2:

vim /groupXvol1/gXsysconfig/gXreplication-script.sh

#!/bin/bash
zfs list -rt snapshot -o name |grep groupXvol1/gXsubvol1 |sort |head -n -5 |xargs -n 1 zfs destroy -r 2> /dev/null;
date_time=`date +%Y%m%d-%H%M%S`;
zfs snapshot -r groupXvol1/gXsubvol1@$date_time;
new_snap_id="groupXvol1/gXsubvol1@$date_time";
old_snap_id=`cat /groupXvol1/gXsysconfig/gXsubvol1_lsnapid`;
zfs send -I groupXvol1/gXsubvol1@$old_snap_id -R $new_snap_id | root@172.16.108.71 "zfs receive -dvF groupYvol1" ;
echo $date_time > /groupXvol1/gXsysconfig/gXsubvol1_lsnapid;


chmod +x /groupXvol1/gXsysconfig/gXreplication-script.sh


vim /etc/crontab
*/5 * * * * root /groupXvol1/gXsysconfig/gXreplication-script.sh
```

```
// ZFS Host System Tuning ...

## Allocate  12GB RAM
## 1GB = 1073741824
echo 12884901888 >> /sys/module/zfs/parameters/zfs_arc_max

sudo vim /etc/modprobe.d/zfs.conf
options zfs zfs_arc_max=12884901888

ulimit -n
1024000


sudo vim /etc/security/limits.conf
root soft nofile 1024000
root hard nofile 1024000

echo 'performance' > /sys/devices/system/cpu/cpufreq/policy0/scaling_governor
echo 'fs.file-max = 1024000' >> /etc/sysctl.conf

sudo vim /etc/sysctl.conf
# Log Martian Packets
#net.ipv4.conf.all.log_martians = 1
fs.file-max = 1024000



// Enable Jumbo-Frmae/Jumbo-Packets
// Set Network Interface MTU 9000
// Set Switch-Port / Global MTU  9000
```