There were two systems:  
- system1, main system on which most of the configuration take place  
- system2, some configuration here  
**1) Configure selinux.**  
**-  Configure your systems that should be running in Enforcing.**

# vim /etc/selinux/config  
SELINUX=enforcing

**After reboot and verify with this command**

# getenforce

**2) Configure repository.**  
**-  Create a Repository for your virtual machines. The URI is** [**http://station.network0.example.com/content/rhel7.0/x86\_64/dvd**](http://station.network0.example.com/content/rhel7.0/x86_64/dvd)

# vim /etc/yum.repos.d/local.repo  
  
[localrepo]  
name = Local Repo for RHCE Exam  
baseurl = http://station.network0.example.com/content/rhel7.0/x86\_64/dvd  
gpgcheck = 0  
enabled = 1

**Save and Exit (:wq)**  
  
**Then run this:**

# yum clean all  
# yum repolist

**3) SSH configuration.**  
**-  Configure SSH access on your virtual hosts as follows.**  
**-  Clients within my22ilt.org should NOT have access to ssh on your systems**

# vim /etc/hosts.deny  
sshd: .my22ilt.org

Save and Exit (:wq)  
  
**Then run this:**

systemctl restart sshd

**Optional:**

systemctl enable sshd  
firewall-cmd --permanent --add-service=ssh  
firewall-cmd --reload

**4) Configure port forwarding.**  
**-  Configure  serverX to forward traffic incoming on port 80/tcp from source network 172.25.X.0/255.255.255.0 to port on 5243/tcp.**

firewall-cmd --add-rich-rule='rule family="ipv4" source address="172.25.1.0/24" forward-port port="5423" protocol="tcp" to-port="80"' --permanent  
  
firewall-cmd --reload

**5) Customize User Environment.**  
**-  Create a command called qstat on both serverX and desktopX.**  
**-  It should able to  execute the following command(ps -eo pid,tid,class,rtprio,ni,pri,psr,pcpu,stat,wchan:14,comm)**  
**- The command should be executable by all users.**

**vim /etc/bashrc**  
alias qstat='ps -eo pid,tid,class,rtprio,ni,pri,psr,pcpu,stat,wchan:14,comm'  
  
**source /etc/bashrc**

**6) Configure ipv6 network.**  
**-  Configure eth0 with a static ipv6 addresses as follows.**  
**-  Configure a Static IPv6 address in serverX as  fddb:fe2a:ab1e::c0a8:64/64.**  
**-  Configure a Static IPv6 address in desktopX as fddb:fe2a:ab1e::c0a8:02/64.**  
**-  Both machines are able to communicate within the network fddb:fe2a:able/64**  
**-  The changes should be permanent even after the reboot.**  
  
**On ServerX:**

nmcli conn show ----> to find the connection name that attaches to the eth0 interface  
  
nmcli conn modify "System eth0" ipv6.addresses fddb:fe2a:ab1e::c0a8:64/64  
nmcli conn modify "System eth0" connection.autoconnect true  
nmcli conn modify "System eth0" ipv6.method manual  
  
nmcli conn down "System eth0"  
nmcli conn up "System eth0"

**On DesktopX:**

nmcli conn show ----> to find the connection name that attaches to the eth0 interface  
  
nmcli conn modify "System eth0" ipv6.addresses fddb:fe2a:ab1e::c0a8:02/64  
nmcli conn modify "System eth0" connection.autoconnect true  
nmcli conn modify "System eth0" ipv6.method manual  
  
nmcli conn down "System eth0"  
nmcli conn up "System eth0"

**On ServerX:**

ping6 -I eth0 ddb:fe2a:ab1e::c0a8:02

**On DesktopX:**

ping6 -I eth0 fddb:fe2a:ab1e::c0a8:64

**7) Link aggregation**  
**-  Configure your serverX and desktopX, which watches for link changes and selects an active port for data transfers.**  
**-  serverX should have the address as 192.168.X.10/255.255.255.0.**  
**-  desktopX should have the address as 192.168.X.11/255.255.255.0.**  
**(Note: where X is your station number)**  
  
**On Server Machine:**

nmcli con add type team con-name Team1 ifname Team1 config '{"runner": {"name": "activebackup"}}'  
nmcli con modify Team1 ipv4.addresses 192.168.1.10/24  
nmcli con modify Team1 ipv4.method manual  
nmcli con add type team-slave con-name Team1-slave1 ifname eth1 master Team1  
nmcli con add type team-slave con-name Team1-slave2 ifname eth2 master Team1  
  
nmcli con up Team1  
nmcli con up Team1-slave1  
nmcli con up Team1-slave2

Verification & Testing:

teamdctl Team1 state  
nmcli dev dis eth1 ---> Disconnect device for verification  
nmcli con up Team1-slave1  
teamnl Team1 ports  
teamnl Team1 getoption activeport  
teamnl Team1 setoption activeport PORT\_NUMBER  
  
ping –I Team1 192.168.1.11

**On Desktop Machine:**

nmcli con add type team con-name Team1 ifname Team1 config '{"runner": {"name": "activebackup"}}'  
nmcli con modify Team1 ipv4.addresses 192.168.1.11/24  
nmcli con modify Team1 ipv4.method manual  
nmcli con add type team-slave con-name Team1-slave1 ifname eth1 master Team1  
nmcli con add type team-slave con-name Team1-slave2 ifname eth2 master Team1  
  
nmcli con up Team1  
nmcli con up Team1-slave1  
nmcli con up Team1-slave2

Verification & Testing:

teamdctl Team1 state  
nmcli dev dis eth1 ---> Disconnect device for verification  
nmcli con up Team1-slave1  
teamnl Team1 ports  
teamnl Team1 getoption activeport  
teamnl Team1 setoption activeport PORT\_NUMBER  
  
ping –I Team1 192.168.1.10

**8) SMTP Configuration.**  
**-  Configure the SMTP mail service on serverX and desktopX which  relay the mail only from local system through station.network0.example.com, all outgoing mail have their sender domain as example.com. Ensure that mail should not store locally.**  
**- Verify the mail server is working by sending mail to a natasha user.**  
**- Check the mail on both serverX and desktopX  with the below URL**  
[**http://station.network0.example.com/system1**](http://station.network0.example.com/system1)  
[**http://station.network0.example.com/system2**](http://station.network0.example.com/system2)

yum install postfix  
systemctl enable postfix  
systemctl start postfix

vim /etc/postfix/main.cf  
  
inet\_interfaces = loopback-only  
mydestination =  
myorigin = example.com  
mynetworks = 127.0.0.0/8, [::1]/128  
relayhost = [station.network0.example.com]  
local\_transport = error: local delivery disabled

systemctl restart postfix

**9) NFS server.**  
**-  Configure serverX with the following requirements.**  
**-  Share the /nfsshare directory within the example.com domain clients only, share must be writable.**  
**-  Share the /nfssecure, enable krb5p security to secure access to the NFS share from URI** [**http://station.network0.example.com/pub/keytabs/serverX.keytab**](http://station.network0.example.com/pub/keytabs/serverX.keytab)  
**-  Create a directory named as protected under /nfssecure**  
**-  The exported directory should have read/write access from all subdomains of the example.com domain.**  
**-   Ensure the directory /nfssecure/protected should be owned by the user harry with read/write permission.**

yum install –y nfs\*

mkdir –p /nfsshare  
chmod 0777 /nfsshare

vim /etc/exports  
/nfsshare \*.example.com(rw)

systemctl restart nfs-server  
systemctl enable nfs-server  
firewall-cmd --permanent --add-service=nfs  
firewall-cmd --reload

mkdir –p /nfssecure  
wget –O /etc/krb5.keytab http://station.network0.example.com/pub/keytabs/serverX.keytab

vim /etc/sysconfig/nfs  
RPCNFSDARGS="-V 4.2"

systemctl enable nfs-secure-server  
  
  
mkdir /nfssecure/protected  
  
  
vim /etc/exports  
/nfssecure \*.example.com(rw,sec=krb5p,sync)  
  
  
grep –i “harry” /etc/passwd  
  
(If it return nothing, then create the user harry)  
[indent=1]·      useradd –u 3000 harry --- IT SHOULD BE nologin or not?[/indent]  
  
chown harry /nfssecre/protected  
  
Best it to do like this:  
  
setfacl –m u:harry:rwX /nfssecure/protected  
  
exportfs -r  
  
semanage fcontext -a -t public\_content\_rw\_t "/nfsshare(/.\*)?"  
semanage fcontext -a -t public\_content\_rw\_t "/nfssecure(/.\*)?"  
restorecon -Rv /nfssecure/  
  
firewall-cmd --permanent --add-service=rpc-bind  
firewall-cmd --permanent --add-service=mountd  
firewall-cmd –reload  
  
systemctl restart nfs-server  
systemctl restart nfs-secure-server  
systemctl enable nfs-secure-server

**10) Configure nfs mount.**  
**-  Mount /nfsshare directory on desktopX under /public directory persistently at system boot time.**  
**-  Mount /nfssecure/protected with krb5p secured share on desktopX beneath /secure/protected provided with keytab** [**http://station.network0.example.com/pub/keytabs/desktopX.keytab**](http://station.network0.example.com/pub/keytabs/desktopX.keytab)  
**- The user harry able to write files on /secure directory**

yum install –y nfs-utils  
  
wget –O /etc/krb5.keytab http://station.network0.example.com/pub/keytabs/desktopX.keytab  
  
systemctl start nfs-secure  
systemctl enable nfs-secure  
  
mkdir –p /public  
  
vim /etc/fstab  
  
server1.example.com:/nfsshare /public nfs defaults,sync 0 0  
  
mkdir –p /secure/protected  
  
vim /etc/fstab  
  
server1.example.com:/nfssecure/protected /secure/protected nfs defaults,v4.2,sec=krb5p,sync 0 0

**Verification from DesktopX:**

ssh harry@localhost  
cd /secure/protected  
echo “Is it writeable?” >> test.txt

**11) Configure smb access.**  
**- Share the /sambadir directory via SMB on serverX**  
**- Your SMB server must be a member of the TESTGROUP workgroup**  
**- The share name must be data**  
**- The data share must be available to example.com domain clients only**  
**- The data share must be browseable**  
**- susan must have read access to the share, authenticating with the same password “password”, if necessary**  
**-  Configure the serverX to share /opstack with SMB share name must be cluster.**  
**-  The user frankenstein has readable,writeable,accesseable to the /opstack SMB share.**  
**-  The user martin has read access to the /opstack SMB share.**  
**-  Both users should have the SMB passwd "SaniTago".**

yum install samba samba-client  
  
systemctl start smb nmb  
systemctl enable smb nmb  
  
firewall-cmd --permanent --add-service=samba  
firewall-cmd --reload  
  
mkdir -p /sambadir  
  
semanage fcontext -a -t samba\_share\_t "/sambadir(/.\*)?"  
restorecon -Rv /sambadir  
  
setfacl –m u:susan:r-X /sambadir  
  
vim /etc/samba/smb.conf  
  
workgroup = TESTGROUP  
[data]  
comment = data share  
path = /sambadir  
browseable = yes  
valid users = susan  
read only =yes  
hosts allow = 172.25.1. #(ifconfig and get your ip and only use the 3 octets)  
  
grep –i “susan” /etc/passwd  
(It it return nothing then create a user first)  
  
useradd -s /sbin/nologin susan  
smbpasswd -a susan  
  
  
  
mkdir –p /opstack  
  
semanage fcontext -a -t samba\_share\_t "/ opstack (/.\*)?"  
restorecon -Rv / opstack  
  
vim /etc/samba/smb.conf  
[cluster]  
comment = opstack share  
path = /opstack  
write list = frankenstein  
writable = no  
  
useradd -s /sbin/nologin frankenstein  
useradd -s /sbin/nologin martin  
  
smbpasswd –a Frankenstein  
smbpasswd –a martin  
  
#Allow Frankenstein write access & Martin read access to the directory  
[indent=1]1) setfacl -m u:frankenstein:rwX /opstack/[/indent]  
[indent=1]2) setfacl -m u:frankenstein:r-X /opstack/

[/indent]  
  
**12) smb multiuser mount.**  
**-  Mount the samba share /opstack permanently beneath /mnt/smbspace on desktopX as a multiuser mount.**  
**-  The samba share should be mounted with the credentials of frankenstein.**

yum –y install cifs-utils samba-client  
  
mkdir –p /mnt/smbspace  
  
vim /root/smb-multiuser.txt  
  
username=frankenstein  
password= SaniTago  
  
chmod 0600 /root/multiuser.txt  
  
vim /etc/fstab  
  
//server1/cluster /mnt/smbspace cifs defaults,sec=ntlmssp,credentials=/root/smb-multiuser.txt,multiuser 0 0

**13)  Webserver.**  
**-  Implement a webserver for the site** [**http://serverX.example.com**](http://serverX.example.com)  
**-  Download the webpage from** [**http://station.network0.example.com/pub/rhce/rhce.html**](http://station.network0.example.com/pub/rhce/rhce.html)  
**-  rename the downloaded file in to index.html.**  
**-  copy the file into the document root.**  
**-  Do not make any modification with the content of the index.html.**   
**-  Clients within my22ilt.org should NOT access the webserver on your systems**

yum install httpd httpd-manual  
  
systemctl start httpd  
systemctl enable httpd  
  
firewall-cmd --permanent --add-service=http  
firewall-cmd –reload  
  
  
wget http://station.network0.example.com/pub/rhce/rhce.html  
  
mv rhce.html /var/www/html/index.html  
  
cd /etc/httpd/conf.d/  
  
vim server1.conf  
  
<VirtualHost \*:80>  
ServerAdmin webmaster@server1.example.com  
ServerName server1.example.com  
DocumentRoot /var/www/html  
CustomLog "logs/server1\_access\_log" combined  
ErrorLog "logs/server1\_error\_log"  
</VirtualHost>  
  
<Directory "/var/www/html">  
<RequireAll>  
         Require all granted  
         Require not host my22ilt.org  
</RequireAll>  
</Directory>  
  
systemctl restart httpd

**14) secured webserver**  
**-  configure the website** [**https://serverX.example.com**](https://serverX.example.com) **with TLS**  
**-  SSLCertificate file** [**http://classroom.example.com/pub/rhce/tls/certs/system1.networkX.crt**](http://classroom.example.com/pub/rhce/tls/certs/system1.networkX.crt)  
**-  SSLCertificatekeyfile** [**http://classroom.example.com/pub/rhce/tls/private/system1.networkX.key**](http://classroom.example.com/pub/rhce/tls/private/system1.networkX.key)  
**-  SSL CA certificate file** [**http://classroom.example.com/pub/example-ca.crt**](http://classroom.example.com/pub/example-ca.crt)

yum install –u mod\_ssl  
  
wget http://classroom.example.com/pub/rhce/tls/certs/system1.network1.crt  
  
wget http://classroom.example.com/pub/rhce/tls/private/system1.network1.key  
  
wget http://classroom.example.com/pub/example-ca.crt  
  
mv system1.network1.crt /etc/pki/tls/certs/  
mv system1.network1.key /etc/pki/tls/private/  
mv example-ca.crt /etc/pki/tls/certs/  
  
# Very Important, Fix the Permission on Key File  
chmod 0600 /etc/pki/tls/private/system1.network1.key  
  
vim /etc/httpd/conf.d/server1.conf  
  
(Add the following)  
  
<VirtualHost \*:443>  
ServerName server1.example.com  
DocumentRoot /var/www/html  
  
SSLEngine on  
SSLCertificateFile /etc/pki/tls/certs/localhost.crt  
SSLCertificateKeyFile /etc/pki/tls/private/localhost.key  
#SSLCertificateChainFile /etc/pki/tls/certs/server-chain.crt  
</VirtualHost>  
  
firewall-cmd --permanent --add-service=https  
firewall-cmd –reload

**15) Webpage content modification.**  
**- Implement website for** [**http://serverX.example.com/owndir**](http://serverX.example.com/owndir)  
**- Create a directory named as "owndir" under the document root of webserver Download** [**http://station.network0.example.com/pub/rhce/restrict.html**](http://station.network0.example.com/pub/rhce/restrict.html)  
**- rename the file into index.html**  
**- The content of the owndir should be visible to everyone browsing from your local system but should not be accessible from other location**

mkdir /var/www/html/owndir  
restorecon –Rv /var/www/html  
cd /var/www/html/owndir  
  
wget http://station.network0.example.com/pub/rhce/restrict.html  
mv restrict.html intex.html  
  
vi /etc/httpd/conf.d/server1.conf  
(Add this)  
  
<Directory "/var/www/html/owndir">  
AllowOverride None  
Require all Denied  
Require local  
</Directory>  
  
systemctl restart httpd

**16) Virtual hosting.**  
**- Setup a virtual host with an alternate document root.**  
**- Extend your web to include a virtual for the site** [**http://vhostsX.example.com**](http://vhostsX.example.com)  
**- Set the document root as /usr/local/vhosts**  
**- Download** [**http://station.network0.example.com/pub/rhce/vhost.html**](http://station.network0.example.com/pub/rhce/vhost.html)  
**- rename it as index.html**  
**- place this document root of the virtual host**  
**- Note: The other websites configures for your server must still accessible. vhosts.networkX.example.com is already provide by the name server on example.com**

Check that the mentioned document root exists by:  
  
cd /usr/local/vhosts  
  
If it doesn’t exist then create it:  
  
mkdir /usr/local/vhosts  
cd /usr/local/vhosts  
wget http://station.network0.example.com/pub/rhce/vhost.html  
mv vhost.html index.html  
  
semanage fcontext -a -t httpd\_sys\_content\_t "/usr/local/vhosts(/.\*)?"  
restorecon -Rv /usr/local/vhosts/  
  
Create the configuration of new virtual host:  
vim /etc/httpd/conf.d/vhosts.conf  
<VirtualHost \*:80>  
ServerAdmin webmaster@vhosts1.example.com  
ServerName vhosts1.example.com  
DocumentRoot /usr/local/vhosts  
CustomLog "logs/vhosts\_access\_log" combined  
ErrorLog "logs/vhosts\_error\_log"  
</VirtualHost>  
  
<Directory "/usr/local/vhosts">  
AllowOverride None  
# Allow open access:  
Require all granted  
</Directory>  
  
systemctl restart httpd

**17) Dynamic Webpage Configuration.**  
**-  configure website** [**http://wsgiX.example.com:8961**](http://wsgiX.example.com:8961) **on system1 with the documentroot  /var/www/scripts**  
**-  Site should executes webapp.wsgi.**  
**-  Page is already provided on** [**http://classroom.example.com/pub/webapp.wsgi**](http://classroom.example.com/pub/webapp.wsgi)  
**-  Content of the script should not be modified.**

yum install -y mod\_wsgi  
  
mkdir –p /var/www/scripts  
cd /var/www/scripts  
wget http://classroom.example.com/pub/webapp.wsgi  
restorecon –Rv /var/www/scripts  
  
vim /etc/httpd/conf/httpd.conf  
  
Listen 8961  
  
vim /etc/httpd/conf.d/wsgi1.conf  
  
<VirtualHost \*:8961>  
ServerAdmin webmaster@wsgi1.example.com  
ServerName wsgi1.example.com  
DocumentRoot /var/www/scripts # We don’t need it,only testing  
WSGIScriptAlias / /var/www/scripts/webapp.wsgi  
CustomLog "logs/wsgi\_access\_log" combined  
ErrorLog "logs/wsgi\_error\_log"  
</VirtualHost>  
  
<Directory "/var/www/scripts">  
AllowOverride None  
# Allow open access:  
Require all granted  
</Directory>  
  
firewall-cmd --permanent --add-port=8961/tcp  
firewall-cmd --reload  
  
semanage port -a -t http\_port\_t -p tcp 8961  
  
systemctl status httpd

**Verification from Server2**:

yum install -y elinks  
links --dump http://wsgi1.example.com:8961  
Should present with the desired page

**18) Script1**  
**-  create a script on serverX called /root/random with following details.**  
**-  When run as /root/random postconf, should bring the output as "postroll"**  
**-  When run as /root/random postroll, should bring the output as "postconf"**  
**-  When run with any other argument or without argument,**  
**should bring the stderr as "/root/random postconf|postroll"**

vim /root/random  
#!/bin/bash  
case $@ in  
postconf)  
         echo "postroll"  
         ;;  
postroll)  
         echo "postconf"  
         ;;  
\*)  
         echo "/root/random postconf|postroll"  
         ;;  
esac  
**chmod +x /root/random**

**20) Script2**  
**-  Create a script on serverX called /root/createusers**  
**-  When this script is called with the argument, it should add all the users from the file**  
**-  Download the file from** [**http://station.network0.example.com/pub/testfile**](http://station.network0.example.com/pub/testfile)  
**-  All users should have the login shell as /bin/false, password not required.**  
**-  When this script is called with anyother argument, it should print the message as "Input File Not Found"**  
**-  When this script is run without any argument, it should display "Usage: /root/createusers"**  
**-  NOTE:  If the users are added  no need to delete.**

cd /root  
wget [url="http://station.network0.example.com/pub/testfile"]http://station.network0.example.com/pub/testfile[/url]  
  
vim /root/createusers  
  
#!/bin/bash  
a=''  
case $@ in  
testfile)  
         for user in $(cat $1);do  
         echo "Adding this user:" $user  
         useradd -s /bin/false $user  
         done  
         ;;  
$a)  
         echo "Usage: /root/createusers"  
         ;;  
\*)  
         echo "Input File Not Found"  
         ;;  
esac  
  
chmod +x /root/createusers

**21) Configure SCSI storage.**  
**-  Create a new 1GB target on your serverX.example.com.**  
**-  The block device name should be data\_block**  
**-  The server should export an iscsi disk called iqn.2014-10.com.example:serverX.**  
**-  This target should only be allowed to desktopX**

yum install –y targetcli  
  
systemctl start target  
systemctl enable target  
firewall-cmd --permanent --add-port=3260/tcp  
firewall-cmd –reload  
  
#targetcli  
  
backstores/block/create data\_block /dev/sdb1  
iscsi/ create iqn.2014-10.com.example:server1  
cd iscsi/iqn.2014-10.com.example:server1/tpg1/  
acls create iqn.2014-10.com.example:desktop1  
luns/ create backstores/block/data\_block  
portals Server\_IP(172.25.x.11) 3260  
exit

**22) ISCSI Initiator**  
**-  The serverX.example.com provides an iscsi port(3260).**  
**connect the disk with desktopX.example.com and configure filesystem with the following requirements.**  
**-  Create 800MB partition on ISCSI block device and assign the filesystem as xfs.**  
**-  Mount the volume under /mnt/initiator at the system boot time.**  
**-  The filesystem should contains the copy of** [**http://station.network0.example.com/pub/iscsi.txt.**](http://station.network0.example.com/pub/iscsi.txt.)  
**-  The file sould be owned by root with 0644 permission.**  
**-  NOTE: content of the file should not be modified.**

yum install –y iscsi-initiator-utils  
  
vim /etc/iscsi/initiatorname.iscsi  
InitiatorName=iqn.2014-11.com.example:desktop1  
  
systemctl start iscsi  
systemctl start iscsid  
  
systemctl enable iscsi  
systemctl enable iscsid  
  
iscsiadm --mode discoverydb --type sendtargets --portal server1.example.com --discover  
iscsiadm --mode node --targetname iqn.2014-11.com.example:server1 --portal server1.example.com:3260 --login

**Verification:**

iscsiadm –m session –P 3 (it should show the State: running)  
lsblk

fdisk /dev/sdb  
Create the partition of 800M  
  
mkfs.xfs /dev/sdb1  
  
mkdir –p /mnt/initiator  
mount /dev/sdb1 /mnt/initiator  
  
blkid /dev/sdb1  
  
vim /etc/fstab  
  
UUID=c9213938-6753-4001-b939-4b5720c8ec5e /mnt/initiator xfs \_netdev 0 0  
  
cd /mnt/initiator  
wget http://station.network0.example.com/pub/iscsi.txt  
chown root iscsi.txt  
chmod 0644 iscsi.txt

**23) Mariadb**  
  
**- Restore a database on serverX from the backup file** [**http://classroom.example.com/pub/rhce/backup.mdb.**](http://classroom.example.com/pub/rhce/backup.mdb.)  
**- The database name should be Contacts. It should be access only within the localhost.**  
**- Set a password for root user as "Postroll". Other than the root user, the user andrew able to read the query from the above mentioned database. The user should be authenticated with the password as  "Postroll".**

yum groupinstall -y mariadb mariadb-client  
systemctl start mariadb  
systemctl enable mariadb  
  
(We don’t need to open firewall port because it says that only access from localhost)  
  
  
mysql\_secure\_installation  
  
wget http://classroom.example.com/pub/rhce/backup.mdb  
  
  
mysql –u root –p  
CREATE DATABASE Contacts;  
CREATE USER andrew@localhost IDENTIFIED BY 'Postroll';  
GRANT SELECT ON Contacts.\* TO andrew@localhost;  
  
mysql -u root -p Contacts < backup.mdb