RHCE7

1. Salvarsi Nameserver (DNS) + Gateway e fare un nmcli show eth0 di server e client
2. Enable/start dei servizi
3. Firewall
4. getsebool
5. semanage f-context -l & man
6. Permessi 0600 su private key e multiuser.txt
7. HTTP: yum groups install "Server web di base" oppure httpd, httpd-manual,mod\_wsgi, mod\_ssl
8. Samba Server: samba\* Samba client: samba-client, cifs-utils\*
9. NFS: Verificare ntpd (ntpq -p), nfs-secure nfs-server, nfs,
10. Mariadb: yum install mariadb\*
11. Firefox dal desktop per testare il sito
12. cappellino redhat ->pagina web domande
13. kvm -> quadrato system1 system2
14. system1 (rettangolo) ->start console - invio uguale per system2
15. systemd.unit=rescue.target
16. mandb
17. vedere versione redhat
18. fare ssh su ipv4 e ipv6

There were two systems:

– system1: system1.group3.example.com

– system2: system2.group3.example.com

**Configure SElinux**

|  |
| --- |
| Configure your systems (system1 e system2) that should be running in Enforcing.  **setenforce 1** -> in modo da lavorarci senza rebootare  # vim /etc/selinux/config  SELINUX=enforcing  After reboot and verify with this command  # getenforce |

Configure repository

|  |
| --- |
| Create a Repository for your virtual machines. The URI: http://mirror.centos.org/centos/7/os/x86\_64/  #yum-config-manager--add-repo=http://mirror.centos.org/centos/7/os/x86\_64/  gpgcheck=0  Save and Exit (:wq)  Then run this:  # yum clean all  # yum repolist |

SSH

**1. Configure the SSH Access**

|  |
| --- |
| Configure the SSH Access as required:  - Users can visit your two virtual machine systems via clients of domain group3.example.com  – Clients within my22ilt.org should NOT have access to ssh on your systems  Modify file /etc/hosts.allow  Add a line: **sshd: 172.24.11.** -----------> ricordarsi il punto alla fine  Modify file /etc/hosts.deny  Add a line: **sshd: .my22ilt.org**  --------------> ricordarsi il punto all’inizio  Both of them need to be configured  Then run this:  **systemctl restart sshd**  \*verificare se sshd è abilitato sul firewall e se sta su enable  \*per troubleshooting /var/log/secure |

**2. Create Custom Environment**

|  |
| --- |
| Create a custom command on **system1 and system2** named as qstat, this custom command will execute the following commands:  /bin/ps - Ao pid,tt,user,fname,rsz  This command is valid for all users in the system  vim /etc/bashrc // restart and keep valid  **alias qstat=’/bin/ps -Ao pid,tt,user,fname,rsz’**  **source /etc/bashrc**  alias // check if there is a qstat  qstat // execute  Both have to do |

3. Configure Port Forwarding

|  |
| --- |
| Configure port forwarding on the **system1**, as required:  - Systems in the network 172.24.11.0/24, local port 5423 will be ported to 80 when visiting  system1  - This setting must be valid permanently  # man firewalld.richlanguage | grep -i family (per la sintassi)  **firewall-cmd --add-rich-rule='rule family="ipv4" source address="172.24.1.0/24" forward-port port="5423" protocol="tcp" to-port="80"' --permanent**  #firewall-cmd --reload  \* se serve ricordarsi il masquerade  \* verificare anche se sia attivo il port forwarding  sysctl --all | grep ip\_forward -> per la sintassi  vim /etc/systcl.conf  net.ipv4.ip\_forward=1  sysctl -p -> per verificare (dopo rebootare per prenderle in modo permanente) |

**4. Configure IPV6 Address (system1 e system2)**

|  |
| --- |
| – Configure eth0 with a static ipv6 addresses as follows.  – Configure a Static IPv6 address in system1 as fddb:fe2a:ab1e::c0a8:64/64.  – Configure a Static IPv6 address in system2 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 system1:  nmcli conn show ----> per verificare la connection name (il modify si fa sulla con-name) connesso alla eth0  **nmcli conn modify eth0 ipv6.addresses fddb:fe2a:ab1e::c0a8:64/64**  **nmcli conn modify eth0 connection.autoconnect true**  **nmcli conn modify eth0 ipv6.method manual**  nmcli conn down eth0  nmcli conn up eth0  Per vedere solo IPV6 = #ip -6 a  Nel caso in  On system2:  nmcli conn show ----> per verificare la connection name (il modify si fa sulla con-name) connessa alla eth0  **nmcli conn modify eth0 ipv6.addresses fddb:fe2a:ab1e::c0a8:02/64**  **nmcli conn modify eth0 connection.autoconnect true**  **nmcli conn modify eth0 ipv6.method manual**  nmcli conn down eth0  nmcli conn up eth0  N.B. nel caso in cui non si possa fare il mod, cancellare la connessione e rifarla (in quel caso aggiungere type ethernet, con-name, ifname)  On system1 effettuare il **ping 6 -I (interface)** sull’interfaccia fisica del system2 togliendo /64  ping6 -I eth0 ddb:fe2a:ab1e::c0a8:02  On system2: effettuare il **ping 6 -I (interface)** sull’interfaccia fisica del system1, togliendo /64  ping6 -I eth0 fddb:fe2a:ab1e::c0a8 |

**5. Configure Link Aggregation (system1 & system2)**

|  |
| --- |
| man nmcli-examples (Example 7)  # yum install teamd && NetworkManager-team.x86\_64  Configure a link between system1.group3.example.com and system2.group.example.com  according to the following requirements:  (1) This link use interfaces eth1 and eth2  (2) This link still can work when one of the interfaces fails  (3) This link use the following address **172.16.3.20/255.255.255.0** on system1  (4) This link use the following address **172.16.3.25/255.255.255.0** on system2  (5) This link still keep normal after system reboot  Answer:  man teamd.conf | grep -i activebackup  system1  nmcli connection add con-name team0 type team ifname team0 config  '{"runner":{"name":"activebackup"}}'  nmcli con modify team0 ipv4.addresses **'172.16.3.20/24'**  nmcli connection modify team0 **ipv4.method manual**  nmcli connection add type team-slave con-name team0-p1 ifname eth1  master team0  nmcli connection add type team-slave con-name team0-p2 ifname eth2  master team0  teamdctl team0 state  nmcli connection up team0  nmcli con up team0-p1  nmcli con up team0-p2  **ping –I Team1 172.16.3.25** -> system2 eth0 team0 IP  system2  nmcli connection add con-name team0 type team ifname team0 config  '{"runner":{"name":"activebackup"}}'  nmcli con modify team0 ipv4.addresses '172.16.3.25/24'  nmcli connection modify team0 ipv4.method manual  nmcli connection add type team-slave con-name team0-p1 ifname eth1  master team0  nmcli connection add type team-slave con-name team0-p2 ifname eth2  master team0  teamdctl team0 state  nmcli connection up team0  nmcli con up team0-p1  nmcli con up team0-p2  ping –I Team1 172.16.3.20 -> system1 eth0 team0 IP  \*reboot  **fare la prova in ssh su ipv4 (team) e ipv6** |

**6. Configure the local mail service**

|  |
| --- |
| Configure the mail service on system1 and system2, as required:  1. These systems do not accept external send mails  2. Any mails sent locally are automatically routed to rhgls.domain11.example.com  3. Mails sent from these systems will be dispalyed from rhgls.domain11.example.com  4. You can send mail to local user ‘authur’ to test your configuration system  rhgls.domain11.example.com  5. Have alreday configured to transfer this user’s mails to the following URL  rhgls.domain11.example.com/received\_mail/11  yum install postfix systemctl enable postfix  systemctl start postfix  firewall-cmd --add-service=smtp --permanent  firewall-cmd --reload  **getsebool -a | grep mail -> postfix\_local\_write\_mail\_spool --> on**  /usr/share/doc/postfix-2.10.1/README\_FILES/STANDARD\_CONFIGURATION\_README  vim /etc/postfix/main.cf  inet\_interfaces = loopback-only  mydestination =  **-> va lasciato vuoto**  myorigin = example.com -> **su myorigin ci va il dominio**  mynetworks = 127.0.0.0/8, [::1]/128  relayhost = [station.network0.example.com]  local\_transport = error: local delivery disabled  systemctl restart postfix  test dai client ---------------------> **mail user@example.com**  Mailq (no mail)  netstat port 25 -> deve stare in ascolto ipv4 e ipv6  cat /var/log/maillog --------------------> per trobleshooting |

**7. Configure SMB Shared Folder**

|  |
| --- |
| Configure the SMB service on system1  -Your SMB server must be a member of the STAFF Working Group  -Share the folder /common and the name must be common  -Only clients of domain11.example.com can access the shared common  common must can be read  User andy must can read the content of the shared, if necessary, verification password is  redhat  **system1**:  yum -y install samba\*  systemctl enable smb  systemctl start smb  systemctl enable nmb  systemctl start nmb  firewall-cmd --add-service=samba --permanent  firewall-cmd --add-service=mountd –permanent  firewall-cmd --reload  mkdir /common  semanage fcontext -a -t **samba\_share\_t** "/common(/.\*)?"  restorecon -Rv /common  setfacl –m u:andy:r-X /common  vim /etc/samba/smb.conf  workgroup = **STAFF**  [common]  comment = common  path = /common  browseable = yes  valid users = andy  **hosts allow = 172.24.11. ------------> ricordarsi il punto (only clients of domain11.example.com)**  **smbpasswd -a andy** |

8. Configure Multi-User SMB Mounts

|  |
| --- |
| 9. Configure Multi-User Mount  Share the directory /devops through SMB on the systeml, as required:  1. Share name is devops  2. The shared directory devops just can be used by clients in domain11.example.com  3. The shared directory devop must be able to be browsed  4. User silene must can be able to access this share through read, access code is redhat  5. User akira must can be able to access this share through read and write, access code is  redhat  6. This share permanently mount to system2. domain11.example.com the user /mnt/dev,  make user silene as authentication any users can get temporary write permissions from akira  Answer:  system1：  mkdir /devops  semanage fcontext -a -t **samba\_share\_t** "/devops(/.\*)?"  restorecon -Rv /devops  setfacl -m u:silene:rwX /devops  setfacl -m u:akira:r-X /devops  smbpasswd -a silene  smbpasswd -a akira  vim /etc/samba/smb.conf  [devops]  path = /devops  hosts allow = 172.24.11. #(ifconfig and get your ip and only use the 3 octets)  browseable = yes  **write list = akira**  valid users = akira,silene  systemctl restart smb  systemctl restart nmb  **smbpasswd -a silene**  **smbpasswd -a akira**  system2：  **yum install –y cifs-utils samba-client**  mkdir /mnt/dev  **smbclient -L /system1/ -U silene**  vim /etc/fstab  //system1/devops /mnt/dev cifs  defaults,**multiuser**,username=silene,password=redhat,**sec=ntlmssp** 0 0  df –hT  Testing：  Switch to user akira **on system2**, enter /mnt/dev and view the file:  su akira  cd /mnt/dev  ls  cifscreds add system1  touch 1 |

09. Implement a web server

|  |
| --- |
| Configure a site http://systeml.group3.example.com/ on the system1, then executes the  following steps:  1. From http://rhgls.group3.example.com/materials/rhce.html  2. Download a file, rename the file as index.html, do not modify the file content  3. Copy the file index.html to the DocumentRoot directory of your web server  4. Clients from the domain group3.example.com can access the web service  5. Clients from my133t.org refuse to access the web service  #yum -y install httpd httpd-manual oppure yum groups install "Server web di base"  #firewall-cmd --permanent --add-service=http  #firewall-cmd –reload  #systemctl enable httpd  #systemctl start httpd  **#cd /var/www/html**  #wget -O index.html <http://station.group3.example.com/materials/rhce.html>  p.s. fare un cat dell’Index, può servire se si punta a quello giusto  tail -n 11 /usr/share/doc/httpd-2.4.6/httpd-vhosts.conf > /etc/httpd/conf.d/system1.group3.example.com.conf -> ricordati estensione .conf  <VirtualHost \*:80>  ServerAdmin webmaster@system1.group3.example.com  DocumentRoot "/var/www/html"  ServerName system1.group3.example.com  ErrorLog "/var/log/httpd/system1.group3.example.com-error\_log"  CustomLog "/var/log/httpd/system1.group3.example.com-access\_log" common  </VirtualHost>  <Directory "/var/www/html"> -**>quando lo copi da relax di aggiungere html**  <RequireAll>  Require all granted  Require not host my133t.org  </RequireAll>  </Directory>  **# apachectl configtest**  # systemct restart httpd  \*fare troubleshooting sui log |

10. Configure security web service

|  |
| --- |
| – Configure the website https://systeml.group3.example.com with TLS  – SSLCertificate file http://classroom.example.com/pub/rhce/tls/certs/system1.networkX.crt  – SSLCertificatekeyfile http://classroom.example.com/pub/rhce/tls/private/system1.networkX.key  – SSL CA certificate file http://classroom.example.com/pub/example-ca.crt  #firewall-cmd --permanent --add-service=https  #firewall-cmd –reload  cd /etc/pki/tls/certs/ -> wget http://classroom.example.com/pub/rhce/tls/certs//etc/pki/tls/certs/system1.group3.example.com.crt  cd /etc/pki/tls/private/ -> wget http://classroom.example.com/pub/rhce/tls/private/system1.group3.example.com.key  cd /etc/pki/tls/certs/ -> wget http://classroom.example.com/pub/example-ca.crt  -rw-------. root root unconfined\_u:object\_r:**cert\_t**:s0 system1.g  # Very Important, Fix the Permission on Key File  **chmod 0600 /etc/pki/tls/private/system1.network1.key**  vim /etc/httpd/conf.d/server1.conf  cat /etc/httpd/conf.d/ssl.conf | grep SSLE  cat /etc/httpd/conf.d/ssl.conf | grep SSLC  (Add the following)  <VirtualHost \*:443>  SSLEngine on  SSLCertificateFile /etc/pki/tls/certs/system1.group3.example.com.crt  SSLCertificateKeyFile /etc/pki/tls/private/system1.group3.example.com.key  **SSLCertificateChainFile /etc/pki/tls/certs/server-chain.crt**  </VirtualHost> |

11. Configure the Virtual Host

|  |
| --- |
| Expand your web server on the system1, create a virtual host for site  http://**www.group3.example.com**  Then Executes the following steps:  1. Setting the DocumentRoot to /var/www/virtual  2. From http://rhgls.domain11.example.com/materials/www.html  3. Download a file, rename as index.html, don’t modify file index.html content  4. Put the file index.html under the directory DocumentRoot of Virtual Host  5. Ensure that user andy can create files under directory /var/www/virtual  Note: original site http://system1.group3.example.com/ must still be able to be accessed,  Name server group3.example.com provide the domain name resolution for host name  #mkdir –p /var/www/ virtual  #cd /var/www/ virtual  #wget –O index.html  <http://rhgls.domain11.example.com/materials/www.html>  **Add in virtual host:**  <VirtualHost \*:80>  ServerAdmin webmaster@group3.example.com  DocumentRoot "**/var/www/virtual**"  ServerName www.group3.example.com  ErrorLog "/var/log/httpd/group3.example.com-error\_log"  CustomLog "/var/log/httpd/group3.example.com-access\_log" common  </VirtualHost>  #<Directory "/var/www/virtual">  # AllowOverride None  # # Allow open access:  # Require all granted  #</Directory>  **#setfacl -m u:andy:rwx /var/www/virtual**  #su andy  touch /var/www/virtual/11.html |

12. Configure Web Content Access

|  |
| --- |
| Create a directory private under directory DocumentRoot of web server on the system1, as  required:  1. Download a file copy to this directory from  http://rhgls.domain11.example.com/materials/private.html and rename it as index.html.  2. Dont make any changes to this file content  3. Any users from the system1 can browse the content of the private, but can not access this directory content from other systems  **mkdir /var/www/virtual/private**  -> document root virtual  cd /var/www/virtual/private  wget -O index.html http://rhgls.domain11.example.com/materials/private.html  cd /var/www/html/private  **mkdir /var/www/html/private** -> document root default  cd /var/www/html/private  wget -O index.html http://rhgls.domain11.example.com/materials/private.html  cd /var/www/html/private  **<Directory "/var/www/html/private">**  AllowOverride none  Require all denied  Require local  </Directory>  **<Directory "/var/www/virtual/private">**  AllowOverride none  Require local  Require all denied  </Directory> |

**13. Configure NFS Service**

|  |
| --- |
| Configure NFS service on **system1**, requirements are as follow:  (1) Share the directory /public in read access and only can be accessed by systems in  domian domain11.example.com  (2) Share the directory /protected in read-write access and can be accessed by systems in domain11.example.com  (3) Access /protected through secure encryption from kerberos, you can use the key from the following URL http://host.domain11.example.com/materials/nfs\_server.keytab  (4) The directory /protected should contain a subdirectory named project and owner is deepak  (5) User deepak can read-write access /protected/project  system1:  **yum install –y nfs\***  systemctl enable nfs  systemctl enable nfs-secure  systemctl enable nfs-secure-server ->only in 7.0  systemctl start nfs  systemctl start nfs-secure  systemctl start nfs-secure-server  firewall-cmd --add-service=**nfs** –permanent  firewall-cmd --add-service=**rpc-bind** –permanent  firewall-cmd --add-service=**mountd**–permanent  firewall-cmd --reload  mkdir -p /public  **chmod 0777 /public**  mkdir -p /protected/project  **chmod 0777 /protected/project**  chown deepak /protected/project/  setfacl -m u:deepak:rwx /protected/project/  semanage fcontext -l | grep public  semanage fcontext -a -t **public\_content\_t** "/public(/.\*)?"  semanage fcontext -a -t **public\_content\_rw\_t** "/protected/project/(/.\*)?"  restorecon -Rv /public  restorecon -Rv /protected/project  vim /etc/exports  /public **172.24.11.0/24**(ro,sync) oppure -----> /public oppure \*.example.com(ro,sync)  /protected **172.24.11.0/24**(rw,sync,sec=krb5p) oppure ------> /protected \*.example.com(rw,sec=krb5p,sync)  **wget -O /etc/krb5.keytab**  http://host.domain11.example.com/materials/nfs\_server.keytab  vim /etc/sysconfig/nfs  RPCNFSDARGS="-V 4.2 "  systemctl restart nfs  systemctl start nfs-secure-server  systemctl enable nfs-secure-server  exportfs –rav  showmount –e localhost  \*verificare ntp se è in sync con l’IPA Server  \*disabilitare e stoppare chronyd  \*klist -k |

**14. Mount an NFS Share**

|  |
| --- |
| Mount an NFS share system1.domain11.example.comon system2, and meet the following  requirements:  (1) Mount /public to the directory /mnt/nfsmount  (2) Mount /protected to the directory /mnt/nfssecure and use a safe mode. Key can download from the following URL: http://host.domain11.example.com/materials/nfs\_client.keytab  (3) User deepak can creat files in /mnt/nfssecure/project  (4) These file systems mount automatically when system start  system2:  yum install nfs-utils\* & **yum groups install “Network File System Client”**  showmount -e system1  mkdir -p /mnt/nfsmount  vim /etc/fstab  system1:/public /mnt/nfsmount nfs defaults 0 0 mount –a → **se non va provare con IP system1**  df –h  mkdir -p /mnt/nfssecure  **wget -O /etc/krb5.keytab**  http://host.domain11.example.com/materials/nfs\_client.keytab  vim /etc/fstab  /protected:/mnt/nfssecure nfs defaults,**sec=krb5p,v4.2** 0 0  mount –a  Verification from system2:  ssh harry@localhost  cd /secure/protected  echo “Is it writeable?” >> test.txt  \*se la versione è 7.0 = systemctl enable nfs-secure && systemctl start nfs-secure  \*fix by setting use\_fully\_qualified\_names = false in sssd.conf  \*systemctl restart rpc.svgssd ? |

**15. Create a user Adding Script**

|  |
| --- |
| Create a script named /root/batchusers, this script can achieve to create local user for  system1, and these users names will come from a file which contains a list of user names.  Meet the following requirements at the same time:  (1) This script required to provide a parameter, this parameter is a file which contains a list of user names  (2) If dont provide the parameter, this script should give the following message Usage:  /root/batchusers then quit and return to the corresponding value  (3) If provide a file name whihc does not exist, this script should give the following message input file not found then quit and return to the corresponding value  (4) Login shell for created user is /bin/false  (5) This script does not need to set password for users  You can get the list of usersnames for testing from the following URL:  http://rhgls.domain11.example.com/materials/userlist  **/root/createusers (senza argomenti) = Usage: /root/createusers**  **/root/createusers testfile = crea utenti**  **/root/createusers sfavd = Input File Not Found**  #!/bin/bash  users=$(cat /root/testfile)  if [[ $# == 0 ]]  then  echo "Usage: /root/createusers"  elif [[ $1 == testfile ]]  then  for i in $users  do  useradd -s /bin/nologin $i  done  else  echo "Input File Not Found"  fi |

**16. Configure a Database**

|  |
| --- |
| - Restore a database on serverx from the backup file  - The database name should be contactos (onyl for local access)  - Set a password for root user as "postroll"  - User andrew able to read the query  - user should be authenticated with the password "postroll"  # yum install mariadb\*  # systemctl enable mariadb && systemctl start mariadb  # mysql\_secure\_installation -> set postroll password  #mysql -u root -p  postroll  show databases;  create database contactos;  use contactos;  **create user andrew@localhost identified by "postroll";**  **grant select on contactos.\* to andrew@localhost;**  # mysql -u root -p contactos < /root/db01\_contact\_book.dump  Enter password:  mysql -u andrew -p  select \* from book;  select \* from book where name="fidel"; |

**18. Configure iSCSI Server**

|  |
| --- |
| Configure the system1 provide a ISCSI service disk named  iqn.2014-09.com.example.domain11:system1 and meet the following requirements at the same time:  1. Server Port 3260  2. Use iscsi\_store as its back-end volume, its size is 3G  3. This service just can be accessed by system2.domian11.example.com  Solutions:  yum install -y targetcli\*  **systemctl enable target ------------------> fondamentale**  systemctl start target  **firewall-cmd --add-port=3260/tcp --permanent**  firewall-cmd --reload  cat /proc/partitions o fdisk ------> per vedere dischi da partizionare  pvcreate /dev/sdc  vgcreate vgsan /dev/sdc  lvcreate -n lvsan1 -L 1G vgsan  mkfs.ext4 /dev/vgsn/lvsan  targetcli---->  cd backstores/  block/ create **block1** (nome blocco casuale) **/dev/vgsan/lvan** ----> path LVM creata  cd /iscsi  create iqn.2014-09.com.example.domain11:system1  cd iqn.2014-09.com.example.domain11:system1/  cd tpg1/ ----> questo te lo crea in automatico  poi creare l’acl dove sul wwn va messo il nome che dell’initiator (client)  **acls**/ create iqn.2014-09.com.example.domain11:**client -----------> questo andrà inserito solamente sotto /etc/iscsi/initiatorname.iscsi (ACL)**  luns/ create /backstores/block/block1 -> nome del block creato backstores  Nel portals di defualt c'è questa configurazione 0.0.0.0 3260 va cancellata (del 0.0.0.0 3260)  portals/ create x.x.x.x 3260 (ip e porta del system1 ovvero del target)  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.>  – 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.i686**  **vim /etc/iscsi/initiatorname.iscsi**  InitiatorName=iqn.2014-11.com.example:desktop1 -------->acl creato  systemctl start iscsi  systemctl start iscsid  **systemctl enable iscsi ----------------> fondamentale**  **systemctl enable iscsid ----------------> fondamentale**  **man iscsiadm** (sta tutto in fondo sotto EXAMPLES)--->  1° esempio  **iscsiadm --mode discoverydb --type sendtargets --portal 172.24.11.10 -discover** ------->portal=ip del server = mi da il nome del target  2° esempio  **iscsiadm --mode node --targetname iqn.2014-09.com.example.domain11:system1 --portal 172.24.11.10:3260 –login** ->> inserire nome del target  lsscsi o fdisk -l **---------------------> vedere se ora il client vede il disco condiviso**  poi ti chiedono di fare und isco da 2000MB con fdisk  fdisk –l  fdisk /dev/sdb  mkfs.ext4 /dev/sdb1  partprobe  mkdir /mnt/data  vim /etc/fstab  **UUID=c9213938-6753-4001-b939-4b5720c8ec5e** **/mnt/initiator** ext4 **\_netdev** 0 0 ---> utilizzare **blkid**  cd /mnt/initiator  wget http://station.network0.example.com/pub/iscsi.txt  chown root iscsi.txt  chmod 0644 iscsi.txt  Troubleshooting:  1) sul server verificare che sia in **enable target.service**  2) sul cliente entrare **systemd.unit=rescue.target**  e commentare fstab e poi mettere in disable iscsi e iscsid |

Se servisse…

Dynamic webpage configuration

|  |
| --- |
| http://wsgi1.example.com:8961  documentroot /home/lello/site  page webapp.wsgi  Installare pacchetto:  yum search all wsgi  *#yum -y install mod\_wsgi*  Aggiungere la porta custom:  #Listen 12.34.56.78:80  Listen 80  *Listen 8961*  Aggiungere Porta alle Policy Selinux :  # man semanage-port | grep http  *#semanage port -a -t http\_port\_t -p tcp 8961*  Aggiungere porta sul Firewall:  *# firewall-cmd --add-port=8961/tcp --permanent*  *# firewall-cmd --reload*  mkdir /home/lello/site -p  vim myapp.wsgi (or wget)  ls -lZ  # ls -ldZ /var/www/cgi-bin/  *drwxr-xr-x. root root system\_u:object\_r:httpd\_sys\_script\_exec\_t:s0 /var/www/cgi-bin/*  man semanage-fcontext | grep web  *semanage fcontext -a -t httpd\_sys\_script\_exec\_t "/home/lello/site(/.\*)?"*  *restorecon -Rv /home/lello/site*  cat /usr/share/doc/mod\_wsgi-3.4/README | grep -i alias  WSGIScriptAlias /myapp /usr/local/wsgi/scripts/myapp.wsgi  Sostituire con:  SGIScriptAlias / /home/lello/site/myapp.wsgi  <VirtualHost \*:8961>  ServerAdmin webmaster@dummy-host2.example.com  DocumentRoot "/home/lello/site/"  ServerName wsg1.example.com  ErrorLog "/var/log/httpd/wsg1.example.com-error\_log"  CustomLog "/var/log/httpd/wsg1.example.com-access\_log" common  WSGIScriptAlias / /home/lello/site/myapp.wsgi  </VirtualHost>  <Directory "/home/lello/site">  AllowOverride None  # Allow open access:  Require all granted  </Directory>  # getsebool httpd\_enable\_cgi |

|  |
| --- |
| vim /etc/hosts -> [alternative@example.com](mailto:alternative@example.com)  mkdir -p /usr/local/alternative  semanage fcontext -a -t httpd\_sys\_content\_t "/alternative(/.\*)?"  restorecon -Rv /alternative  <VirtualHost \*:80>  ServerAdmin alternative@example.com  DocumentRoot "/usr/local/alternative"  ServerName alternative.example.com  ErrorLog "/var/log/httpd/alternative.example.com-error\_log"  CustomLog "/var/log/httpd/alternative.example.com-access\_log" common  </VirtualHost>  <Directory /usr/local/alternative>  AllowOverride None  Require all granted  </Directory>  systemctl restart httpd |

Script Postroll/Postconf

|  |
| --- |
| #!/bin/bash  if [[ $1 == postroll ]]  then  echo "postconf"  elif [[ $1 == postconf ]]  then  echo "postroll"  else  echo "/root/random postroll|postconf"  fi |