

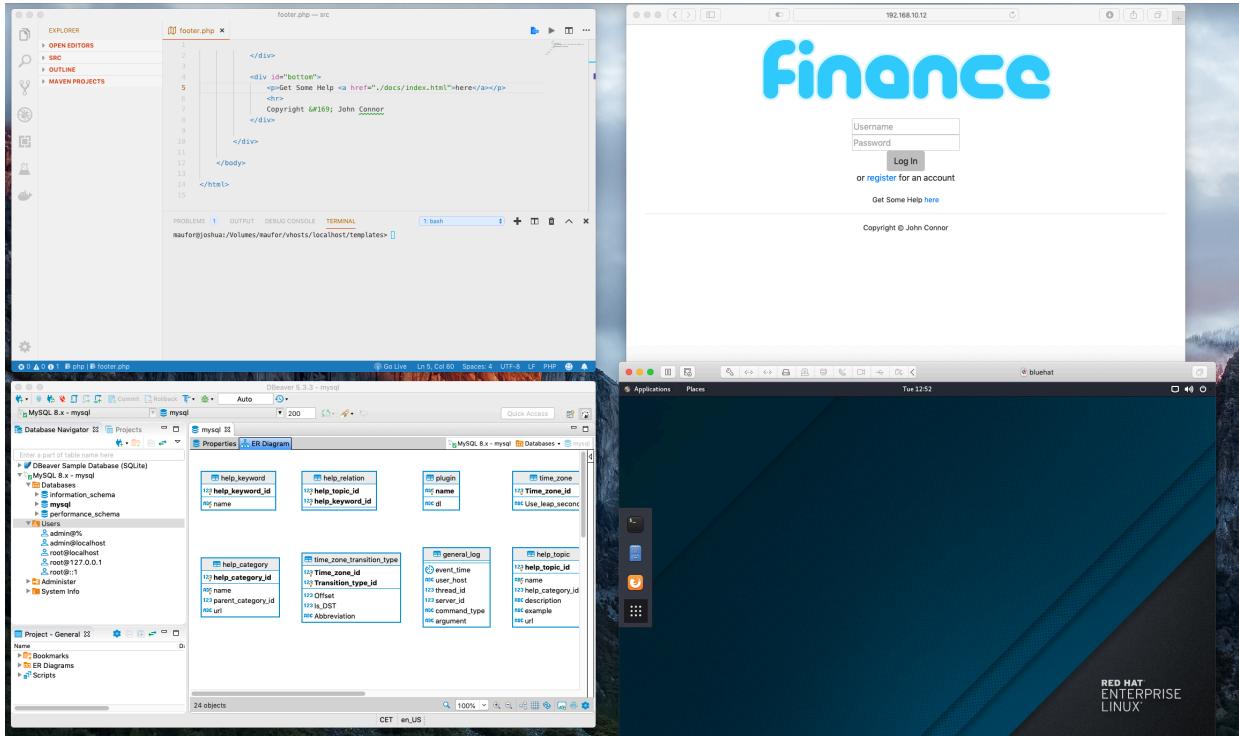
Finance

Disclaimer

Il progetto è un adattamento di **pset7**, [CS50X](#), David J. Malan (Copyright BSD, 2012)

Ambiente di Sviluppo

Scopo di questo documento è configurare tutti i servizi e le componenti necessarie per installare un'*appliance* di sviluppo Linux (CentOS 7.6).



Amministrazione del Sistema

Il sistema operativo sul quale è installata l'*appliance* è assunto essere **CentOS**, release [7.6.1810](#), disponibile all'[URL](#).

L'installazione del sistema è in *virtual machine*, indifferentemente

- [VMWare Player](#) — È necessario installare anche [VMware Tools](#)
- [Oracle VirtualBox](#) — È necessario installare anche [VirtualBox Extension Pack](#). Le versioni dell'hypervisor e dell'extension pack devono corrispondere.

Configurazione Minimale del Sistema

Il sistema operativo **Guest** (CentOS, nel nostro caso), deve essere configurato come segue.

Tipo	Conf	Note
CPU	>=1	<i>Nota:</i> per CPU si intende core . Non superare il 50% del numero di core del sistema Host
RAM	>= 1 GB	<i>Nota:</i> Non superare il 50% della RAM del sistema Host
HD	Si consiglia: formato VMDK con dimensione >= 8 GB , tipo Dinamico	
Networking	Due schede di rete: 1) NAT , 2) Bridged	<i>Nota:</i> la scheda secondaria non è abilitata per default al boot. Inoltre, essa dipende dalla scheda di rete del sistema Host (i.e., se la macchina virtuale è clonata e/o spostata da un PC ad un altro, è necessario configurare la scheda di rete secondaria — una sola volta — <i>prima di avviare</i> la virtual machine per la prima volta sul nuovo sistema)
Shared Folders		<i>Nota:</i> non strettamente necessaria. È possibile, tuttavia, creare un folder nel sistema operativo Host che funga da directory di "scambio" tra i sistemi Host e Guest

Installazione

L'installazione del (o dei) server è, in generale, una fase critica della messa in produzione (**delivery**) di un'applicazione che usi un *ecosistema* (hardware + software + stack di protocolli di rete + infrastruttura). Sono richieste competenze sistemistiche diversificate ...

Per i nostri scopi, tuttavia, il tipo di filesystem, il loro partizionamento, la scelta dei *mount point* è ininfluente e pertanto

durante la fase di installazione confermare le impostazioni di default: gestione dei dischi **LVM** (Logical Volume Manager), filesystem **XFS** (di Silicon Graphics), mount point (partizioni) proposti dal sistema: tipicamente una partizione per il **root file system** (/), una per lo **swap**, una per **home** (/home).

Hostname

Il nome host va scelto in modo tale da essere facilmente identificabile. Il dominio di default è **Localdomain** e va lasciato così com'è (non faremo uso di DNS in questo progetto).

hostname.localdomain

Amministrazione

Nei sistemi di derivazione **UNIX** (Unix-like), l'utente con i privilegi di **amministrazione** è chiamato **root**. Non è sempre installato per default in tutte le distribuzioni Linux (e.g., in Ubuntu non è disponibile all'avvio ma può essere creato successivamente).

Durante la fase di installazione di CentOS sono richieste due utenze:

- **user** — è l'utente che normalmente usa il sistema per compiti non necessariamente di amministrazione. Lo **user** può svolgere temporaneamente compiti di amministrazione (a patto di essere abilitato a farlo) tramite alcuni comandi (**su** e **sudo**);
- **root** — l'amministratore unico del sistema (a cui possono essere affiancate diverse altre utenze in seguito).

È necessario che le utenze siano diverse (per i nostri scopi possono anche avere la stessa password sebbene, in generale, ciò sia altamente sconsigliato).

Post-Install

Il sistema operativo va aggiornato dopo l'installazione:

1. Avviare la shell (**gnome-terminal**)
2. digitare **su** - e la **password** di **root**
3. digitare **yum -y update** ed aspettare che il sistema esegua il download dei pacchetti software da aggiornare (solo il delta tra ciò che è già installato e quanto deve essere modificato), effettui il test dei pacchetti, li decomprima, li installi, e ripulisca il sistema dopo l'aggiornamento. **Nota:** è possibile che il sistema installi un nuovo **kernel** in questa fase. Dopo l'installazione del kernel è necessario riavviare l'host (fatto più unico che raro). Non è necessario farlo subito... aspettare di aver effettuato tutti i passaggi sottostanti
4. installare l'editor **nano** (se non è già presente): **yum -y install nano**
5. sempre dall'account di **root**, disabilitare **SELinux** (sebbene, anche questo non sia consigliato in ambienti di produzione...): **nano /etc/sysconfig/selinux**
6. cercare la riga: **SELINUX=enforcing** e sostituirla con **SELINUX=disabled**
7. salvare il file con la sequenza di tasti **CTRL+O**
8. uscire da **nano** premendo **CTRL+X**
9. optionalmente, disabilitare la *bell* (campanella fastidiosa) del terminale: **nano /etc/inputrc**
10. cercare (prima riga): **#set bell-style none** e rimuovere il commento **#** lasciando la riga come **set bell-style none**. Salvare (**CTRL+O**, **CTRL+X**)
11. riavviare: è possibile farlo dall'interfaccia grafica (tecnicamente è un **Desktop Environment** chiamato **GNOME**), in alto a destra, col simbolo del *power-off*

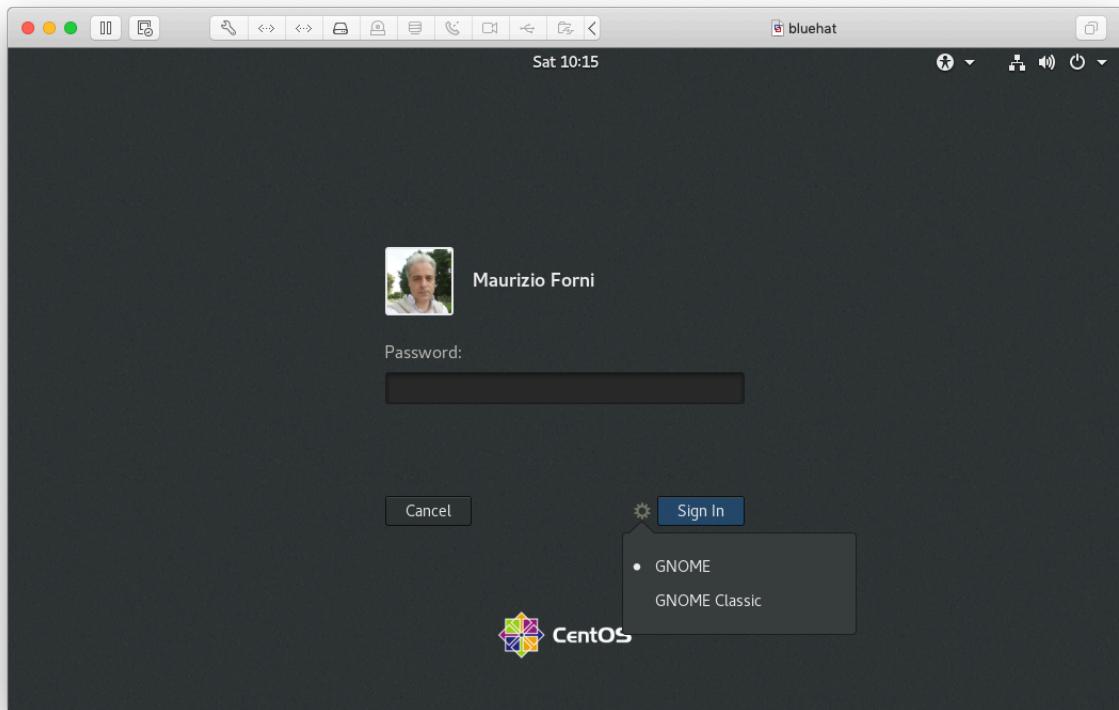
Nota: è possibile che, prima del riavvio, appaia l'azione **install pending software**: confermare e riavviare. Durante il riavvio saranno installati i pacchetti lasciati in sospeso prima dell'aggiornamento. Attendere il completamento.

Addons

Per chi lo desideri, è possibile personalizzare (molto pesantemente) l'aspetto del proprio sistema. Alcune impostazioni sono 'leggere', altre sono invasive.

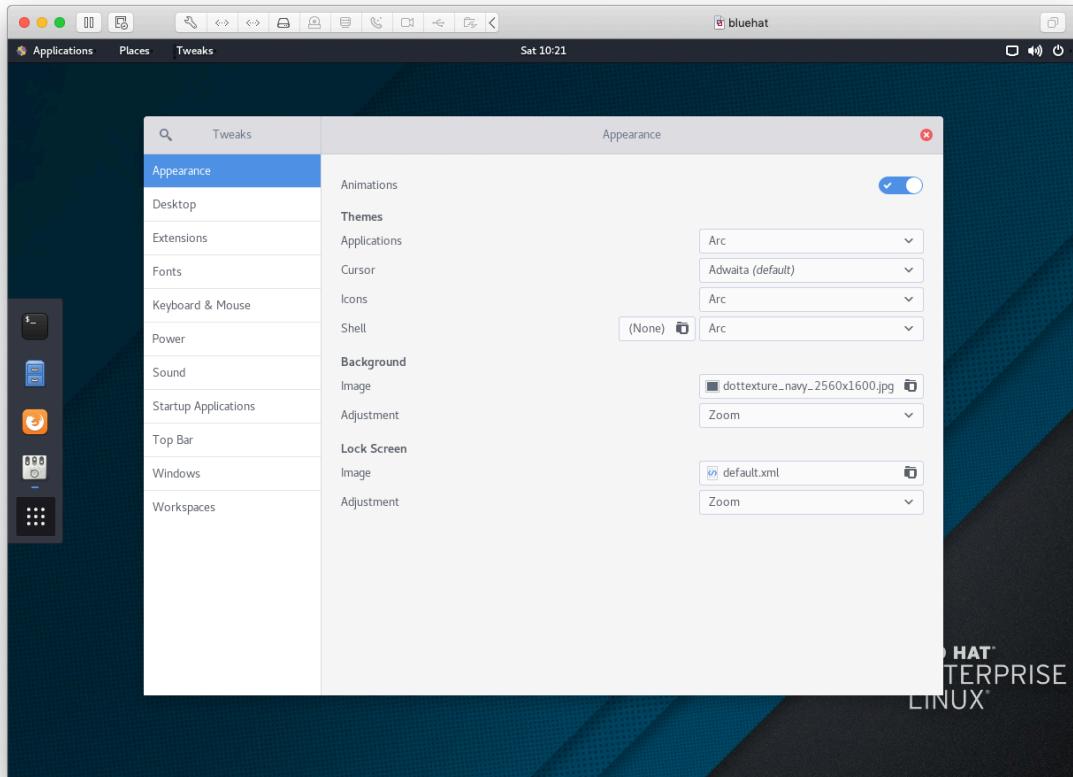
Impostazioni consigliate

1. **Desktop-Environment di default** — CentOS è abbastanza 'conservativo' per cui parte per default con "GNOME Classic". È possibile scegliere, in alternativa, **GNOME** dalla finestra di login (`login manager`)



2. `Gnome Tweak Tools` permette di aggiungere e modificare le impostazioni dell'interfaccia. Per CentOS sono disponibili alcune (non tutte quelle presenti in Fedora e/o Ubuntu) personalizzazioni. Riferirsi alle varie guide presenti sul web. L'aspetto delle finestre e delle icone può essere personalizzato consultando alcuni siti, tra i quali:

- o [gnome-look](#)



3. `.bashrc` e `.bash_profile` — i file (nascosti) controllano le impostazioni della shell. È consigliato modificare il file `.bashrc` ed aggiungere quanto segue alla fine del file:

```
alias l='ls -la'
alias ..='cd ..'
alias ...='cd ../../'

export PS1='\u@\h:\w$ '
```

Si definiscono alcuni comodi `alias` per i comandi più usati e si personalizza l'aspetto del `prompt`.

Nota: se si personalizza il file dell'utente `root`, attenzione al prompt:

```
export PS1='\u@\h:\w# '
```

4. È possibile installare software aggiuntivo (ufficialmente non presente nella distribuzione di default) tramite il repository **epel** (una specie di AppStore, per intenderci) di **Fedora** ed aggiungere software proprietario (e molto altro ancora) tramite il repository **rpmfusion**. Consultare il web per ulteriori informazioni.

Utenze e Gruppi

Ogni utente UNIX dispone di un numero (intero) che identifica la sua utenza: lo **UID** o **User ID**. Un utente, inoltre, può appartenere ad uno o più gruppi identificati dal **GID** o **Group ID**. Di tutti i gruppi ai quali appartiene un utente, il primo è suo gruppo di default (o **primary Group**). Di seguito poche opzioni utili che possono essere comode.

gruppi di un utente

Per visualizzare la lista dei gruppi di un utente:

```
groups user
```

dove **user** è il nome dell'utente (*login name*).

Le informazioni sugli utenti e i gruppi (nonché sulle password) sono racchiusi in alcuni file:

- **/etc/passwd** — lista delle login. Il file riporta alcune informazioni, tra le quali
 - **login**
 - **UID**
 - **GID** del gruppo primario
 - Nome esteso dell'utenza
 - **home**, ossia la directory di default dell'utente. In generale, la directory di default dell'utente **user** è **/home/user** e su di essa l'utente ha i privilegi di accesso completi (lettura, scrittura, esecuzione).
 - **shell**, ossia la shell di default: in generale è **/bin/bash** per le utenze "umane" o **/sbin/nologin** per le utenze di sistema (alle quali si impedisce di disporre di una shell di login per questioni di sicurezza)
- **/etc/group** — lista dei gruppi. Ogni riga riporta:
 - il nome del gruppo
 - il **GID**
 - la lista degli utenti appartenenti al gruppo, separati da virgola
- **/etc/shadow** — file locale delle password (visibile solo da **root**). Le password sono memorizzate come **hash**. L'hash è una funzione **one-way** che genera un fingerprint (o digest) a partire da una sequenza di bit. Avendo a disposizione il digest non è possibile (o molto difficile) ricavare la sequenza di bit che lo ha prodotto. Inoltre, per sequenze iniziali diverse, i digest sono diversi (non ci sono collisioni di digest, o meglio, ve ne sono ma restano casi limitati).

L'hash della password è calcolato aggiungendo alla password scelta dall'utente un valore pseudocasuale detto **salt**. Ciò evita di generare digest uguali per password uguali. Per come è concepito il sistema, la password **non è mai** memorizzata nel file. Se un utente dimentica la propria password è necessario che **root** provveda a cambiarla: nemmeno **root** può ricavare la password originaria dell'utente.

modificare la password

- Da utente (propria password): **passwd**
- da **root**: **passwd user** (per modificare la password di **user**) oppure **passwd** per

modificare la password di `root`

aggiungere utente a gruppo

```
usermod -aG group user
```

dove `group` è il nome del nuovo gruppo del quale `user` deve far parte

Permessi File

Ogni file appartiene ad un utente e ad un gruppo. Ha quindi impostata per default la coppia `UID : GID`. Inoltre, **ogni file ha permessi di accesso:**

- Lettura (`r` - read)
- Scrutture (`w` - write)
- Esecuzione (`x` - execute)

Il listato successivo mostra come leggere le impostazioni dei file.

```
ls -la
```

```
drwxr-xr-x@ 3 maufor staff      96 Jan 15 12:55 dwhelper
lwxr-xr-x  1 maufor staff      7 Jul 14 2017 local -> /local/
-rw-----  1 maufor staff   1024 Nov 13 06:28 .rnd
```

I permessi sono elencati a sinistra e sono composti da 10 campi.

- Primo campo (solitamente `-`, `d`, `l`, etc).
 - `-` file
 - `d` directory
 - `l` soft link (symbolic link)
- seguono 9 campi `rw-`:
 - la prima tripletta è relativa a `UID`
 - la seconda a `GID`
 - La terza a tutti gli altri (resto del mondo)

Pertanto:

```
-rw-----  1 maufor staff   1024 Nov 13 06:28 .rnd
```

È un file (`-`) cui permessi di accesso sono:

- lettura e scrittura per `UID`, in questo caso `maufor`
- nessun permesso per `UID` (`staff`)
- nessun permesso per gli altri

si tratta di un file **nascosto**: tutti i file o directory che iniziano con un punto sono nascosti.

cambiare la proprietà di un file

```
chown user:group file
```

dove `user` è il nome di un utente, `group` il gruppo e `file` è il nome del file. Oppure

```
chown -R user:group directory
```

per cambiare la proprietà della directory e di tutto il suo contenuto (ricorsivamente)

cambiare i permessi di un file

```
chmod NNN file
```

dove `NNN` è un numero decimale. Ogni cifra `N` del numero si ricava da un numero binario. Si associa

- 0 se il permesso non è abilitato
- 1 altrimenti

Si converte il numero binario ottenuto in decimale, ottenendo un numero decimale compreso tra 0 e 7. Ad esempio, dato il file `esempio.text`, il comando `chmod 755 esempio.text` assegna

- a `UID` il permesso 7, ossia `rwx`
- a `GID` il permesso 5, ossia `r-x`
- agli altri 5, cioè `r-x`

Come prima, per cambiare i permessi di una directory (inclusi i suoi file): `chmod -R NNN directory`.

Installazione Samba (SMB/CIFS)

Al fine di agevolare lo scambio di dati tra la virtual machine ed il sistema operativo Windows è possibile installare e configurare il servizio `samba` (attiva il protocollo di Microsoft per l'accesso a directory condivise in rete e/o configura un dominio Windows Active Directory).

1. **Creazione directory condivisa** (chiamata a piacere *scambio* o *share*): con l'utente `root`:

```
su -
mkdir /local/share
chown -R root:users share
chmod -R 775 share

yum -y install samba-client samba-common

firewall-cmd --permanent --zone=public --add-service=samba
firewall-cmd reload

cd /etc/samba
cp smb.conf.example smb.conf
```

Nota: tutti le utenze che potranno accedere alla directory condivisa dovranno far parte del gruppo qui chiamato `users`. Altrimenti non sarà possibile accedere (**access denied**).

I comandi creano la cartella condivisa `share` in `/local`, installano Samba e configurano correttamente il firewall. L'ultimo comando copia il file di configurazione di esempio (`smb.conf.example`) nel file di configurazione vero e proprio (`smb.conf`) che **andrà editato** come segue:

```
[global]
    workgroup = ITIS-SMB
    security = user

    interfaces enp0s8 lo
    bind interfaces only = yes
    networks 192.168.0.0/20

    passdb backend = tdbsam

    printing = cups
    printcap name = cups
    load printers = yes
    cups options = raw

[homes]
    comment = Home Directories
    valid users = %S, %D%w%S
    browseable = No
    read only = No
    inherit acls = Yes

[printers]
    comment = All Printers
    path = /var/tmp
    printable = Yes
    create mask = 0600
    browseable = No

[print$]
    comment = Printer Drivers
    path = /var/lib/samba/drivers
    write list = @printadmin root
    force group = @printadmin
    create mask = 0664
    directory mask = 0775

[share]
    comment = share on bluehat
    path = /local/share
```

```
browsable = yes  
writeable = yes
```

Particolare attenzione va posta alla sezione globale. Si assume che il servizio possa funzionare **solo** sull'interfaccia `enp0s8` che deve essere avviata preventivamente. Inoltre, la riga `networks 192.168.0.0/20` permette di accedere al servizio unicamente dalla rete LAN dell'ITIS. Eventuali altre reti domestiche e/o interfacce vanno elencate esplicitamente. Ad esempio

```
[global]  
workgroup = ITIS-SMB  
security = user  
  
interfaces enp0s8 en34 lo  
bind interfaces only = yes  
networks 192.168.0.0/20 192.168.10.0/24
```

consente di accedere al server Samba dalle interfacce:

- loopback (`lo`)
- LAN Bridged (`enp0s8`)
- LAN Bridged (`en34`)

dalle reti:

- `192.168.0.0/20`
- `192.168.10.0/24`

La riga:

```
bind interfaces only = yes
```

evita che il server possa essere accessibile da interfacce diverse (e.g. `wifi`) mentre la riga

```
workgroup = ITIS-SMB
```

crea un server standalone in workgroup nella rete dell'ITIS. Infine, la riga di commento dello share (`[share]`):

```
[share]  
comment = share on bluehat
```

va modificata sostituendo il nome `bluehat` con il nome del proprio server.

2. **Creazione utenze Samba:** prima di accedere al server è necessario dare i permessi agli utenti. Supponendo che l'utente che usa il PC sia `user`, digitare da `root` quanto segue:

```
smbpasswd -a user  
>>>PASSWORD
```

Si ribadisce che `user` deve far parte del gruppo degli `users`, altrimenti va dato il comando:

```
usermod -aG users user
```

3. **Test e riavvio del server samba:** una volta configurato il servizio, è necessario verificare che la sintassi del file di configurazione sia corretta:

```
testparm
```

Se tutto funziona correttamente, abilitare ed avviare il servizio Samba:

```
systemctl enable smb nmb  
systemctl start smb nmb
```

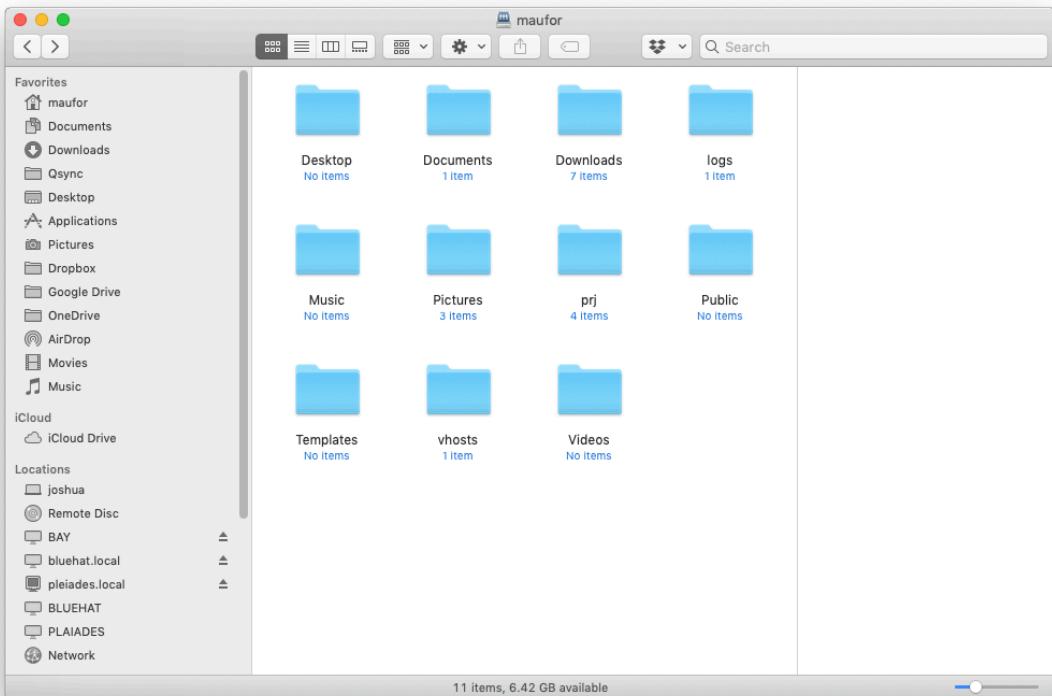
Attendere qualche secondo e provare a connettersi da qualsiasi host Windows tramite:

```
\\\ipaddr
```

dove `ipaddr` è l'indirizzo IP del server Samba.

Per connettersi direttamente alla proprio `home`:

```
\\\ipaddr\~\user
```



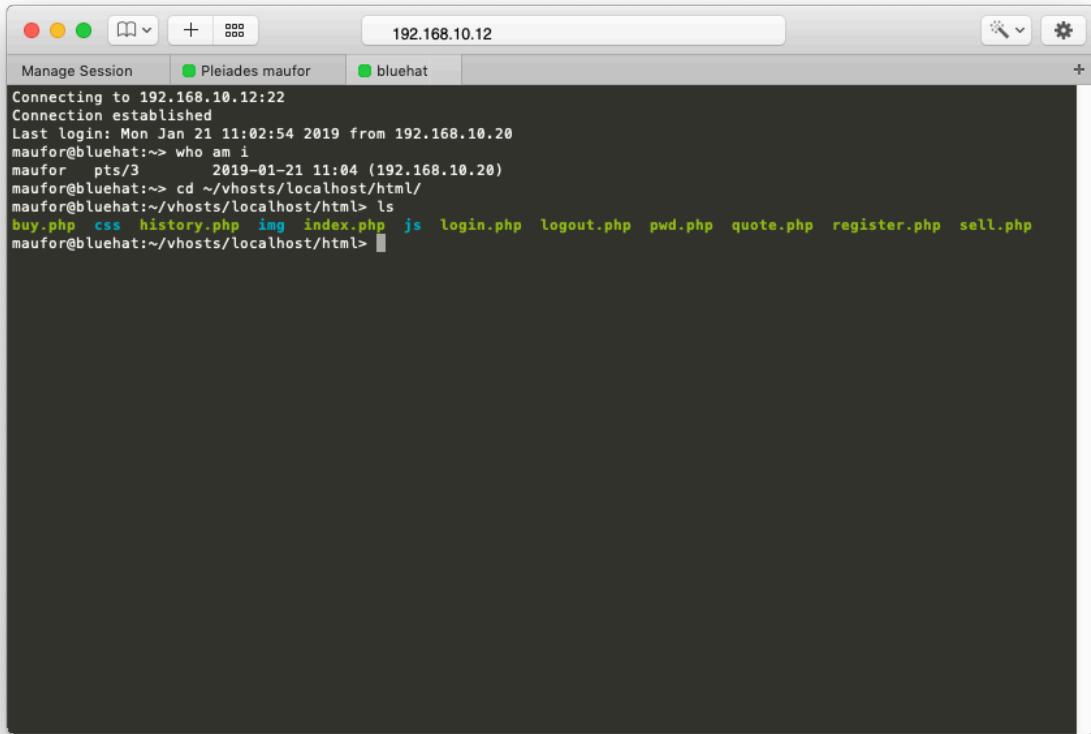
Accesso Remoto con SSH

È possibile accedere al proprio host tramite **Secure Shell** (`ssh`) oppure utilizzando uno strumento come `PUTTY` (nota: quest'ultimo supporta soltanto l'accesso da shell testuale).

Presumendo che il servizio sia attivo (`systemctl status sshd`), la sintassi del comando (per l'accesso solo testuale) è:

```
ssh user@server
```

dove `user` è il nome di login e `server` è l'hostname oppure l'indirizzo IP del server. La sessione termina con `CRTL+D`, oppure con `Logout`.



```
Connecting to 192.168.10.12:22
Connection established
Last login: Mon Jan 21 11:02:54 2019 from 192.168.10.20
maufor@bluehat:~$ who am i
maufor pts/3      2019-01-21 11:04 (192.168.10.20)
maufor@bluehat:~$ cd ~/vhosts/localhost/html/
maufor@bluehat:~/vhosts/localhost/html$ ls
buy.php  css  history.php  img  index.php  js  login.php  logout.php  pwd.php  quote.php  register.php  sell.php
maufor@bluehat:~/vhosts/localhost/html$
```

Installazione Apache

Operazioni di Pre-Configurazione

Assicurarsi che la scheda di rete **bridged** sia avviata. Generalmente è l'interfaccia con nome `enp0s8`

```
ip addr show
```

Se l'interfaccia non è attiva, digitare quanto segue:

```
su -
nano /etc/sysconfig/networking-scripts/ifcfg-enp0s8
```

e quindi cercare la linea `ONBOOT` ed assicurarsi di abilitarla al boot: `ONBOOT=YES`. Salvare e avviare l'interfaccia con

```
ifup enp0s8
```

Installare Apache:

```
yum -y install httpd
systemctl enable httpd
systemctl start httpd
```

Configurare il firewall:

```
firewall-cmd --permanent --add-port=80/tcp  
firewall-cmd --permanent --add-port=443/tcp  
firewall-cmd --permanent --add-port=9090/tcp  
firewall-cmd --reload
```

La porta 9090 è utile per l'amministrazione del server da remoto tramite il tool cockpit. Esso può essere installato come segue:

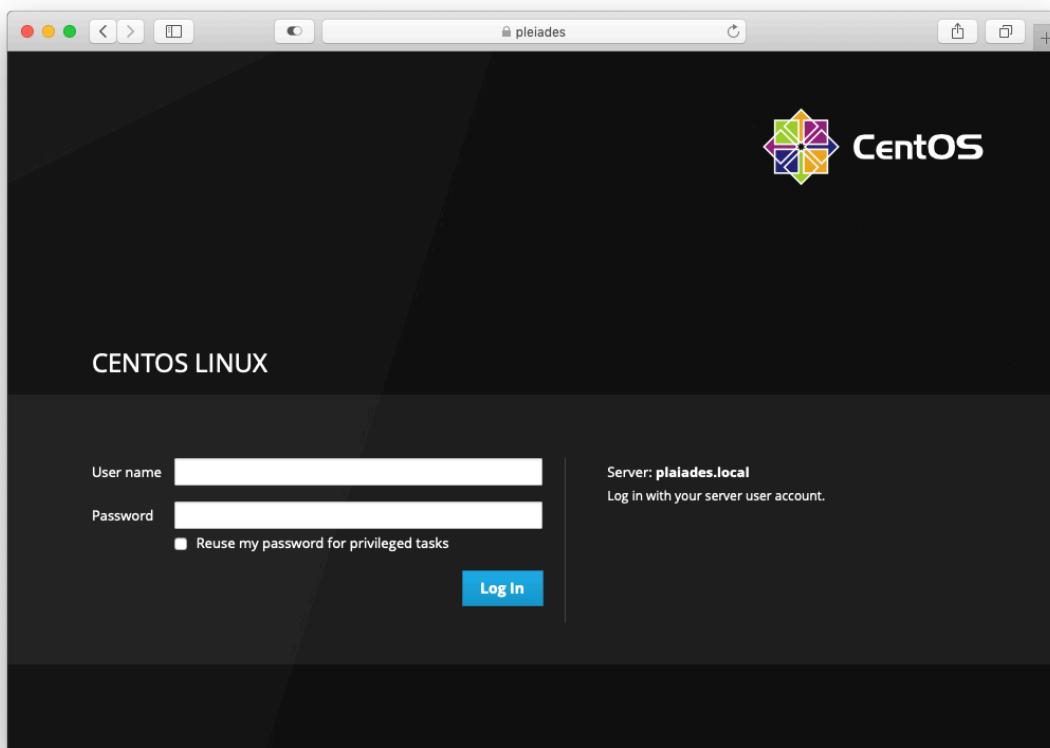
```
yum -y install cockpit  
systemctl start cockpit
```

Avviare un browser e digitare per verificare che tutti funziona normalmente:

<http://localhost>, oppure <http://localhost:9090> per cockpit

Se possibile, verificare che il server risponda anche da un host diverso:

- leggere l'indirizzo IP associato alla scheda: ifconfig oppure ip addr show
- aprire un browser e digitare <http://ipaddr> dove ipaddr è l'indirizzo IP del passo precedente



Configurazione Directory di Sviluppo

```
su -  
cd /home  
chmod -R 711 user
```

dove `user` è la login dell'utente (corrispondente alla home).

Creazione VirtualHost di sviluppo

Sempre dall'utente `root`:

```
nano /etc/httpd/conf.d/localhost.conf
```

e digitare quanto segue:

```
<Directory /home/USER/vhosts/localhost/html/>  
    Options Indexes Multiviews  
    AllowOverride All  
    Require all granted  
</Directory>  
<virtualHost *:80>  
    ServerName      SERVERNAME  
    ServerAlias     SERVERALIAS  
    DocumentRoot   /home/USER/vhosts/localhost/html/  
    CustomLog      /home/USER/logs/httpd/access_log combined  
    ErrorLog       /home/USER/logs/httpd/error_log  
</virtualHost>
```

Nell'esempio vanno sostituite le stringhe `USER`, `SERVERNAME` e `SERVERALIAS` con i nomi opportuni.

Come utente digitare:

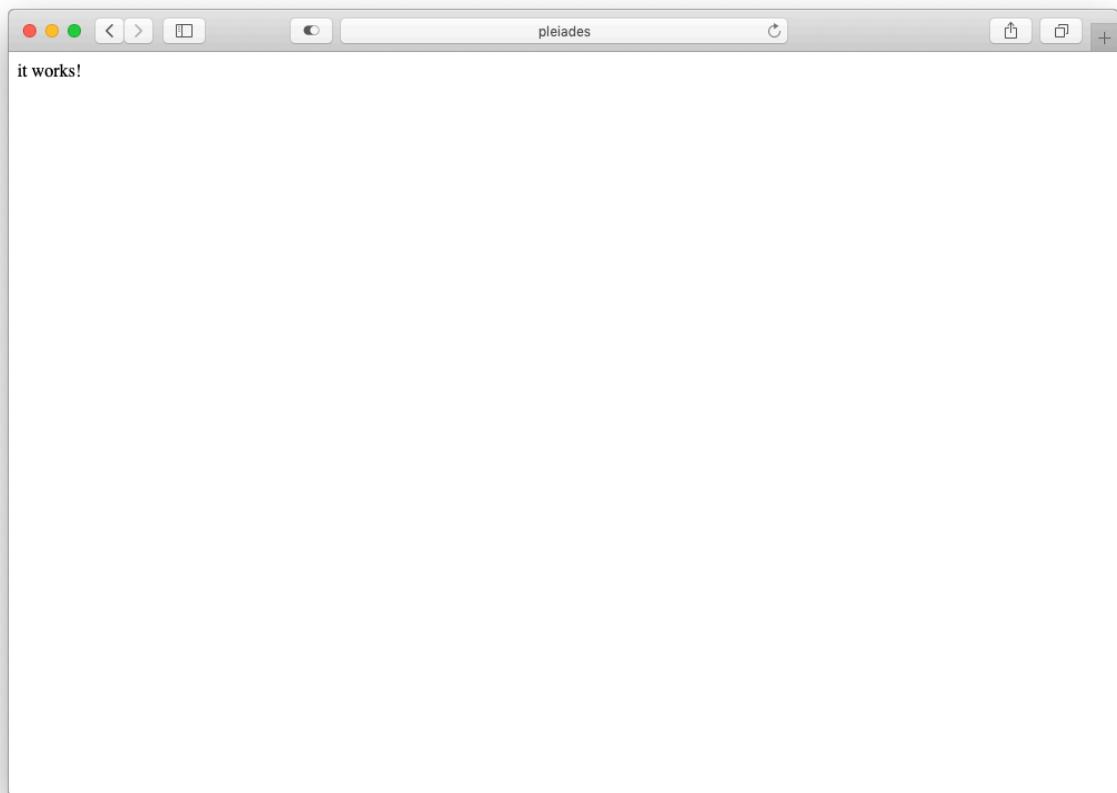
```
cd  
mkdir vhosts vhosts/localhost logs logs/httpd  
chmod -R 755 vhosts  
systemctl restart httpd.service
```

index.html

Verificare il corretto funzionamento dalla directory di sviluppo. Da `USER` creare un file in `~/vhosts/localhost/html` con nome `index.html`

```
<html>  
    it works!  
</html>
```

Assicurarsi che i permessi siano corretti... e provare a digitare `http://ipaddr` nel browser (oppure l'hostname se il server è in un dominio):



Installazione MariaDB

Dall'utente `root`:

```
yum -y install mariadb-server mariadb  
systemctl enable mariadb  
systemctl start mariadb
```

Configurazione Iniziale

```
mysql_secure_installation  
Set root password? [Y/n] Y  
New password: Enter your password here  
Re-enter new password: repeat your password  
Remove anonymous users? [Y/n] Y  
Disallow root login remotely? [Y/n] Y  
Remove test database and access to it? [Y/n] Y  
Reload privilege tables now? [Y/n] Y
```

Riavviare il DBMS:

```
systemctl restart mariadb
```

Abilitare l'utente `admin` ad accedere al database anche da remoto

Nota: nella realtà, questa impostazione è sconsigliata... andrebbe creato un db-administrator per database.

Dall'utente `root` digitare quanto segue:

```
mysql -u root -p
```

Dopo aver inserito la password:

```
CREATE USER admin@localhost;
CREATE USER admin@'%';
GRANT ALL PRIVILEGES ON *.* TO admin@localhost IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON *.* TO admin@'%' IDENTIFIED BY 'password';
FLUSH PRIVILEGES;
```

Installazione PHP

Dall'utente `root`

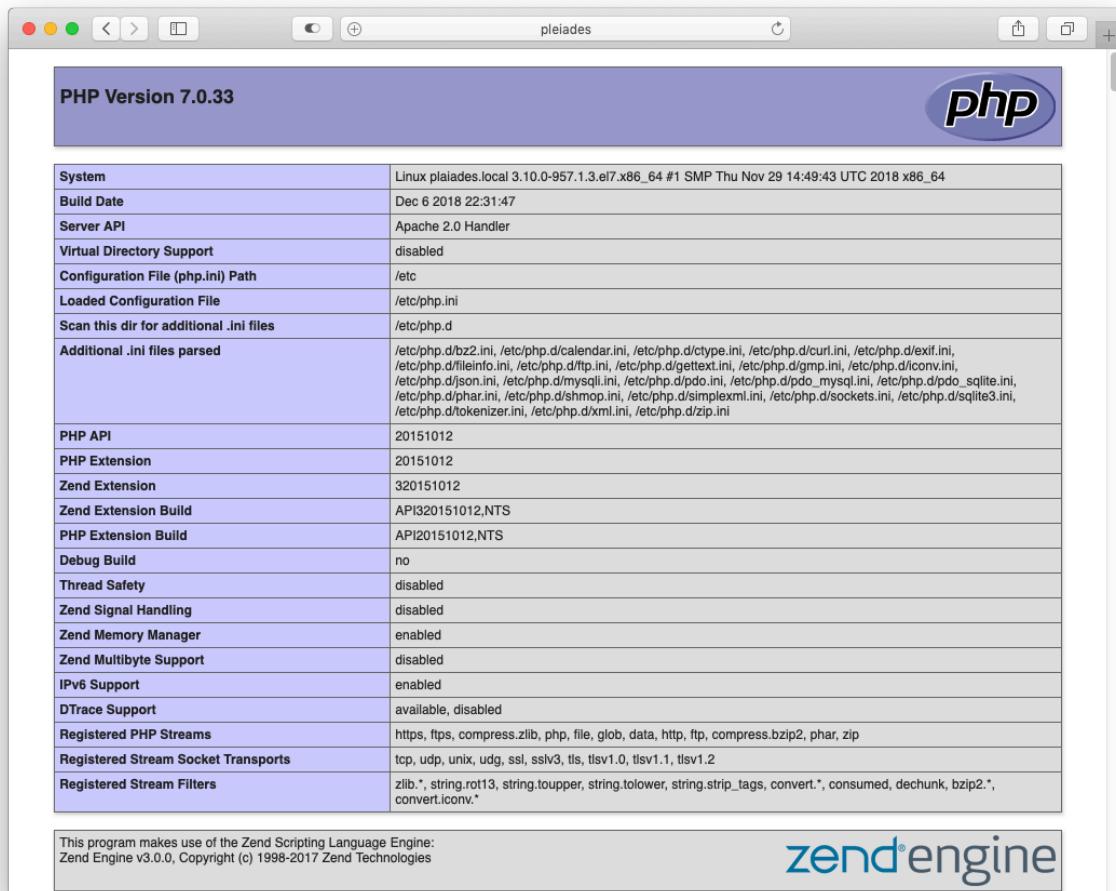
```
yum -y install php php-mysql
systemctl restart httpd
```

Testare PHP

Una volta installato, creare la prima la prima pagina PHP utilizzando `user`. Creare la pagina `~/vhhosts/localhost/html/info.php` contenente:

```
<?
phpinfo();
?>
```

Salvare e provare ad aprire la pagina con il browser: `http://localhost/info.php`



Installazione phpMyAdmin

Se non è stato precedentemente installato, è necessario includere **epel** nella lista dei repository e quindi installare phpMyAdmin:

```
yum -y install epel-release
yum -y install phpmyadmin
systemctl restart httpd.service
```

Nota: L'installazione di default consente di accedere a phpMyAdmin solo da `localhost`.

Quindi è necessario utilizzare un browser nella macchina virtuale (oppure eseguire una sessione remota con `ssh`).

Per abilitare l'accesso da remoto è necessario conoscere l'indirizzo IP della subnet alla quale si è connessi (e preferibilmente disporre di un server con IP statico). Bisogna modificare il file `/etc/httpd/conf.d/phpmyadmin.conf` come segue. Nelle due sezioni relative ad `Apache 2.4`, aggiungere la propria subnet all'opzione `Require ip`:

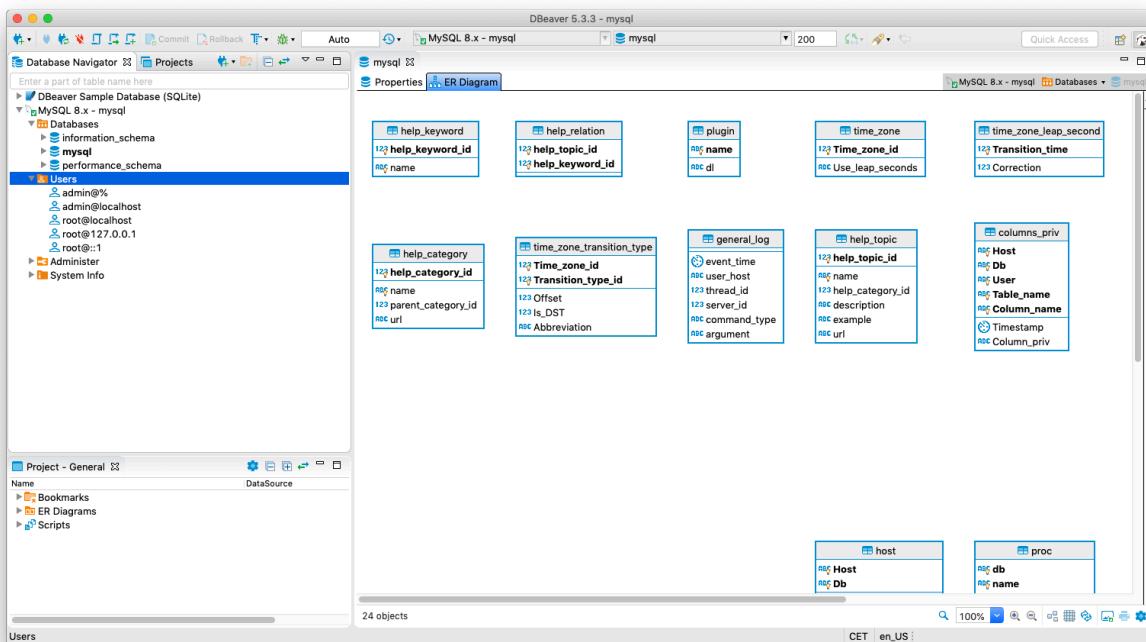
```
# Apache 2.4
<RequireAny>
    Require ip SUBNET/MASK
    Require ip 127.0.0.1
    Require ip ::1
</RequireAny>
```

Accesso a MariaDB da Client Esterni

Per utilizzare strumenti di sviluppo di database anche su altri host è necessario aprire la porta `tcp\3306` sul firewall:

```
firewall-cmd --permanent --add-port=3306/tcp
firewall-cmd reload
```

Successivamente è possibile installare ed usare (da qualsiasi host) un prodotto di sviluppo quale, ad esempio, DBeaver, reperibile (anche nella versione portable) al seguente [URL](#).



Installazione Strumenti per Documentazione

Le specifiche ed il file di help dovranno essere scritti in markdown e renderizzati in HTML. Vi sono diversi strumenti, il più semplici dei quali è `pandoc`:

```
yum -y install pandoc
```

Dall'utente `user` digitare:

```
curl 'https://raw.githubusercontent.com/ryangrose/easy-pandoc-  
templates/master/copy_templates.sh' | bash
```

Il comando crea una directory nascosta nella `home` in cui vi è un singolo template (personalizzabile):

```
cd ~/.pandoc/templates
```

Creare directory Documentazione in WebServer

Da utente `user` :

```
cd ~/vhosts/localhost/html  
mkdir docs  
chmod -R 755 docs
```

all'interno della quale andranno inseriti i file della documentazione scritti in Markdown.

Per renderizzarli in HTML ed ottenere il file `index.html` :

```
pandoc FILE.md -f markdown -o index.html --template=easy_template --toc
```

(Sostituire `FILE.md` con il nome del proprio file)