By Falko Timme

Published: 2006-10-21 20:35

Managing Xen With Xen-Tools, Xen-Shell, And Argo

Version 1.0
Author: Falko Timme <ft [at] falkotimme [dot] com>
Last edited 10/21/2006

This guide describes how to install and use **xen-tools**, **xen-shell**, and **Argo** on a Debian system. All three packages provide useful tools for the administration of virtual Xen machines. Xen-tools is a collection of Perl scripts that allow you to easily create, update, and delete Xen guest domains. The xen-shell provides a command-line interface to owners of Xen domains so that they can manage their Xen domains without the help of the server administrator. And with Argo, you can control Xen domains through a web interface or through a menu on the command line. All three packages were developed for Debian systems, but might work on other distributions as well.

I want to say first that this is not the only way of setting up such a system. There are many ways of achieving this goal but this is the way I take. I do not issue any guarantee that this will work for you!

1 Preliminary Note

I assume you have already installed Xen 3 on your Debian system (it doesn't work with Xen 2!), for example **from the sources** as shown in **chapter 4** of this tutorial: **http://www.howtoforge.com/debian_sarge_xen_3.0.3**. I couldn't get xen-tools to work with the binary Xen installation (shown in chapter 5), but it might work for you. But xen-tools work like a charm for me on a Xen installation from the sources.

The hostname of my Xen *dom0* domain is *server1.example.com* in this tutorial, its IP address is *192.168.0.100*, and the gateway is *192.168.0.1*. I want to store my guest domains in the */vserver/images* directory. These values might be different for you.

Please make sure you have a line like this one in */etc/hosts*:

```
vi /etc/hosts
```

```
[...]
192.168.0.100   server1.example.com server1
[...]
```

2 Xen-Tools

Although there are Debian packages for xen-tools and xen-shell available on **http://www.steve.org.uk/apt/**, I decided to install xen-tools and xen-shell from the sources because the xen-tools Debian package wrote invalid Xen configuration files for me which didn't happen when I used the sources.

First we install some prerequisites for xen-tools:

```
apt-get install libtext-template-perl perl-doc
```

Then we install xen-tools like this:

```
cd /tmp

wget http://xen-tools.org/software/xen-tools/xen-tools-2.7.tar.gz

tar xvfz xen-tools-2.7.tar.gz

cd xen-tools-2.7

make install
```

Next we edit */etc/xen-tools/xen-tools.conf*. This file contains the default values that are used by the *xen-create-image* script unless you specify other values on the command line. I changed the following values and left the rest untouched:

```
vi /etc/xen-tools/xen-tools.conf
```

```
[...]
gateway   = 192.168.0.1
netmask   = 255.255.255.0

passwd = 1

kernel = /boot/vmlinuz-2.6-xenU
#initrd = /boot/initrd.img-2.6.16-2-xen-686

mirror = http://ftp2.de.debian.org/debian/
[...]
```

The *passwd = 1* line makes that you can specify a root password when you create a new guest domain. In the *kernel* line you must specify the *domU* kernel that you want to use for your guest domains. If your guest domains don't need a ramdisk to boot up, comment out the initrd line (I had to do this on my installation), otherwise specify the correct ramdisk. In the *mirror* line specify a Debian mirror close to you.

Make sure you specify a gateway and netmask. If you don't, and you don't specify a gateway and netmask on the command line when using *xen-create-image*, your guest domains won't have networking even if you specified an IP address!

Now let's create our first guest domain, *vm03.example.com*, with the IP address *192.168.0.103*:

```
xen-create-image --hostname=vm03.example.com --ip=192.168.0.103 \

--netmask=255.255.255.0 --gateway=192.168.0.1 --dir=/vserver/images \

--dist=sarge --debootstrap
```

The *--netmask* and *--gateway* switches are unnecessary here because we specified the same details in */etc/xen-tools/xen-tools.conf* but it shows that you can specify the desired settings either on the command line or in */etc/xen-tools/xen-tools.conf*.

This command will now create the guest domain *vm03.example.com* with Debian Sarge in it by using Debian's *debootstrap* tool. This will take some minutes so be patient. In the meantime you can open up another shell window and run

```
tail -f /var/log/xen-tools/vm03.example.com.log
```

to see what's happening behind the scenes.

After the new guest domain has been created, check */etc/xen/vm03.example.com.cfg*. It should now look like this (I've stripped out the comments here):

```
vi /etc/xen/vm03.example.com.cfg
```

```
kernel  = '/boot/vmlinuz-2.6-xenU'
memory  = '128'
root    = '/dev/sda1 ro'
disk    = [ 'file:/vserver/images/domains/vm03.example.com/disk.img,sda1,w', 'file:/vserver/images/domains/vm03.example.com/swap.img,sda2,w' ]
name    = 'vm03.example.com'
vif  = [ 'ip=192.168.0.103' ]
on_poweroff = 'destroy'
on_reboot   = 'restart'
on_crash    = 'restart'
```

We can start our new virtual machine like this:

```
xm create -c /etc/xen/vm03.example.com.cfg
```

With the *xen-create-image* command, you can specify almost all settings on the command line. Have a look at

```
man xen-create-image
```

to find out more. Here's another example:

```
xen-create-image --size=2Gb --swap=128Mb --hostname=vm04.example.com \

--ip=192.168.0.104 --netmask=255.255.255.0 --gateway=192.168.0.1 --dir=/vserver/images \

--dist=sarge --debootstrap --fs=ext3 --kernel=/boot/vmlinuz-2.6-xenU --memory=32Mb \

--image=sparse --mirror=http://ftp2.de.debian.org/debian/ --passwd
```

This will create the virtual domain *vm04.example.com*. */etc/xen/vm04.example.com.cfg* should look like this afterwards (again, I've stripped out the comments):

```
vi /etc/xen/vm04.example.com.cfg
```

```
kernel  = '/boot/vmlinuz-2.6-xenU'
memory  = '32'
root    = '/dev/sda1 ro'
disk    = [ 'file:/vserver/images/domains/vm04.example.com/disk.img,sda1,w', 'file:/vserver/images/domains/vm04.example.com/swap.img,sda2,w' ]
name    = 'vm04.example.com'
vif  = [ 'ip=192.168.0.104' ]
on_poweroff = 'destroy'
on_reboot   = 'restart'
on_crash    = 'restart'
```

You can start the new virtual machine like this:

```
xm create -c /etc/xen/vm04.example.com.cfg
```

There are also three other scripts that come with xen-tools:

xen-update-image lets you update the packages in guest domains, e.g. like this:

```
xen-update-image --dir=/vserver/images vm03.example.com
```

xen-list-images shows all guest domains that were created with *xen-create-image*:

```
xen-list-images
```

The output looks like this:

```
server1:~# xen-list-images
  Name: vm03.example.com
  Memory: 128
IP: 192.168.0.103

Name: vm04.example.com
  Memory: 32
  IP: 192.168.0.104
```

With *xen-delete-image* you can remove guest domains that were created with *xen-create-image*:

```
xen-delete-image --dir=/vserver/images vm03.example.com
```

3 Xen-Shell

The xen-shell provides a special command-line interface that allows users to connect to the host system (*dom0*) via SSH and to control their own virtual Xen machine. This shell knows only commands which are needed to manage the virtual machine, so it cannot be abused for other tasks. It also allows the owner to reimage his virtual machine (if he messed up his current one), i.e., reset it to a pristine/new state.

To allocate a virtual machine to a user, it must have the same name as the user. For example, if you have a user *bob*, then his virtual machine must also be named *bob* instead of *vm03.example.com*, etc.

To demonstrate this, I first create a user *bob* and give him a password:

```
useradd -d /home/bob -m -g users -s /bin/bash bob

passwd bob
```

Now I create a virtual machine called *bob*:

```
xen-create-image --hostname=bob --ip=192.168.0.105 --netmask=255.255.255.0 \

--gateway=192.168.0.1 --dir=/vserver/images --dist=sarge --debootstrap
```

Then we boot up the guest domain to see if it works:

```
xm create -c /etc/xen/bob.cfg
```

and shut it down again:

```
xm shutdown bob
```

Next we install some prerequisites for xen-shell:

```
apt-get install sudo libterm-readline-gnu-perl
```

Afterwards, we install xen-shell like this:

```
cd /tmp

wget http://xen-tools.org/software/xen-shell/xen-shell-0.5.tar.gz

tar xvfz xen-shell-0.5.tar.gz

cd xen-shell-0.5

make install
```

Now we must change `bob`'s login shell from `/bin/bash` to `/usr/bin/xen-login-shell`:

```
chsh -s /usr/bin/xen-login-shell bob
```

(

Next, I change Debian's default text editor to `vi`:

```
update-alternatives --config editor
```

<-- 3 (/usr/bin/nvi)

)

Now we edit */etc/sudoers* by using *visudo*. We have to allow *bob* to use the commands */usr/sbin/xm* and */usr/bin/xen-create-image* which require root privileges:

```
visudo
```

```
[...]

User_Alias   XENUSERS = bob
Cmnd_Alias   XEN      = /usr/sbin/xm
Cmnd_Alias   XENIMG   = /usr/bin/xen-create-image
XENUSERS     ALL      = NOPASSWD: XEN,XENIMG
```

This is necessary because otherwise *bob* won't be able to use the *reimage* function of the xen-shell.

To use the *reimage* function, we also need a shell script called *image.sh* in *bob*'s home dir */home/bob*. This script must contain the commands to be executed to reset or create a new guest domain for *bob*. The contents is totally up to you. For example, it could look like this:

```
vi /home/bob/image.sh
```

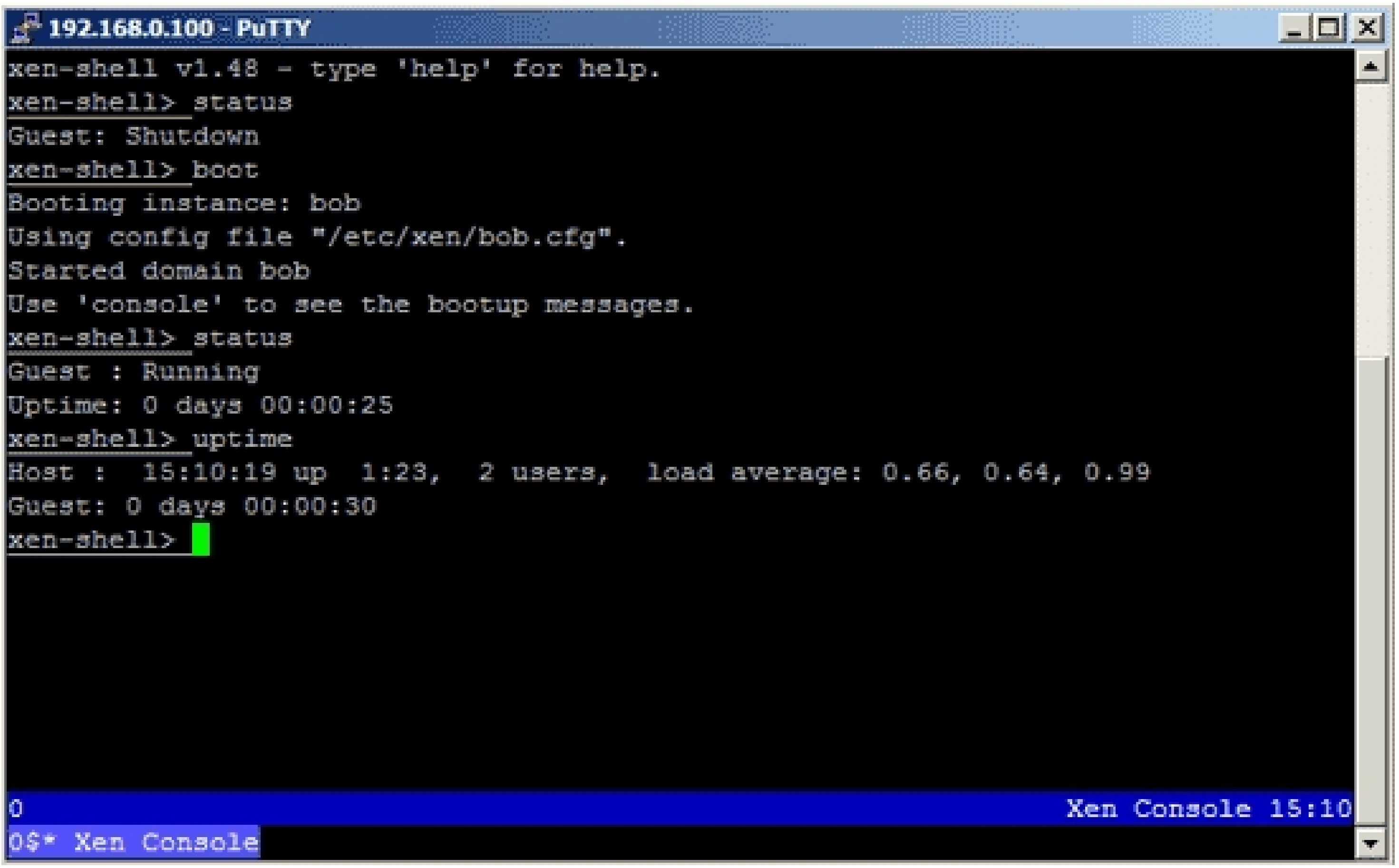
```
#!/bin/sh
/usr/bin/sudo /usr/bin/xen-create-image --hostname=bob --ip=192.168.0.105 \
--netmask=255.255.255.0 --gateway=192.168.0.1 --dir=/vserver/images \
--dist=sarge --debootstrap --force
```

(You should use full paths in the script, and you must invoke the *xen-create-image* command with */usr/bin/sudo*, otherwise *bob* isn't allowed to run *xen-create-image*.)

We must make the script executable:

```
chmod 755 /home/bob/image.sh
```

Now `bob` can use his favourite SSH client (like **PuTTY** for Windows) and connect to `dom0` (`192.168.0.100`). If all goes well, `bob` will see the xen-shell:



```
xen-shell v1.48 - type 'help' for help.
xen-shell> status
Guest: Shutdown
xen-shell> boot
Booting instance: bob
Using config file "/etc/xen/bob.cfg".
Started domain bob
Use 'console' to see the bootup messages.
xen-shell> status
Guest : Running
Uptime: 0 days 00:00:25
xen-shell> uptime
Host :  15:10:19 up  1:23,  2 users,  load average: 0.66, 0.64, 0.99
Guest: 0 days 00:00:30
xen-shell>
```

The following commands are available on the xen-shell:

- `boot` - This allows you to boot your Xen guest if it is currently shutdown.
- `console / serial` - These, identical, commands allow you to access your Xen guest's console, allowing you to login if your networking is broken, or if you've chosen to disable the OpenSSH server.
- `exit / quit` - These commands both exit the shell, saving any commands you might have entered into the ~/.xen-shell history file.
- `help [command1 command2]` - When invoked with no arguments the help command will show a list of available commands and a one-line description of them all. If you wish help on a specific command you can use "help command" to see more details.

- `passwd` - If the client is connecting to the host system via OpenSSH using a plain text password then this command will allow them to change that password. The command is disabled/removed if the users home directory contains the file ~/.ssh/authorized_keys - since it is assumed this means key-based authentication is in force.
- `rdns [ip hostname]` - The rdns command allows you to either view your current reverse DNS settings, or update them. Run rdns with no arguments to view your current setup, or run "rdns 1.2.3.4 foo.bar.com" to set the reverse DNS for the IP 1.2.3.4 Note: you can only set the reverse DNS for IPs which you control, which you'll find listed in the output of rdns. See the installation page for what is required to implement this fully.
- `reboot` - This allows you to reboot your running Xen guest.
- `reimage` - After prompting for confirmation, and counting down for ten seconds this command will wipe your Xen guest, and perform a fresh installation via the use of xen-tools.
- `shutdown` - This command will shutdown your currently running Xen guest. It will remain shutdown until you issue a boot command.
- `status` - This command shows you the current status of your Xen guest, which will either be "running/booted", or "shutdown". For information purposes it will also show you the uptime of the host system.
- `uptime` - This command is similar to the status command and will show you the uptime of your Xen guest along with the uptime of the host system.
- `version` - This shows you the version of the Xen shell which is installed, which is the CVS revision number.

4 Argo

Argo has a client written in PHP which provides a web interface to Xen. To run these PHP scripts, we need a web server with PHP:

```
apt-get install apache2 apache2-doc
```

```
apt-get install libapache2-mod-php4 libapache2-mod-perl2 php4 php4-cli php4-common php4-curl php4-dev php4-domxml php4-gd php4-imap php4-ldap php4-mcal php4-mhash php4-mysql php4-odbc php4-pear php4-xslt
```

You will be asked the following question:

`Continue installing libc-client without Maildir support?` <--

There are Debian packages for Argo which we can install. Therefore, we edit `/etc/apt/sources.list` and add these two lines:

```
vi /etc/apt/sources.list
```

```
[...]
deb     http://www.steve.org.uk/apt sarge main non-free contrib
deb-src http://www.steve.org.uk/apt sarge main non-free contrib
[...]
```

Then we run

```
apt-get update
```

and install Argo:

```
apt-get install argo-client-dialog argo-client-php argo-server
```

To connect to Argo, we must provide a username and password. The default username is *admin*, the default password is *password*. Both are specified in */etc/argo-server/argo-server.conf*, and if you like you can change them:

```
vi /etc/argo-server/argo-server.conf
```

```
[...]
admin : verysecretpassword, *
[...]
```

Next, we must change the default Apache vhost so that we can access Argo using a web browser. Therefore, we replace the default Apache vhost like this:

```
cp /etc/apache2/sites-available/default /etc/apache2/sites-available/default_orig

cat /dev/null > /etc/apache2/sites-available/default

vi /etc/apache2/sites-available/default
```

```
<VirtualHost *>
        DocumentRoot /usr/share/argo/
    ServerName   xen
    ServerAlias  xen.my.flat

        DirectoryIndex index.php

    # Logfiles
        ErrorLog  /var/log/apache2/xen.log
        CustomLog /var/log/apache2/xen.log combined

</VirtualHost>
```

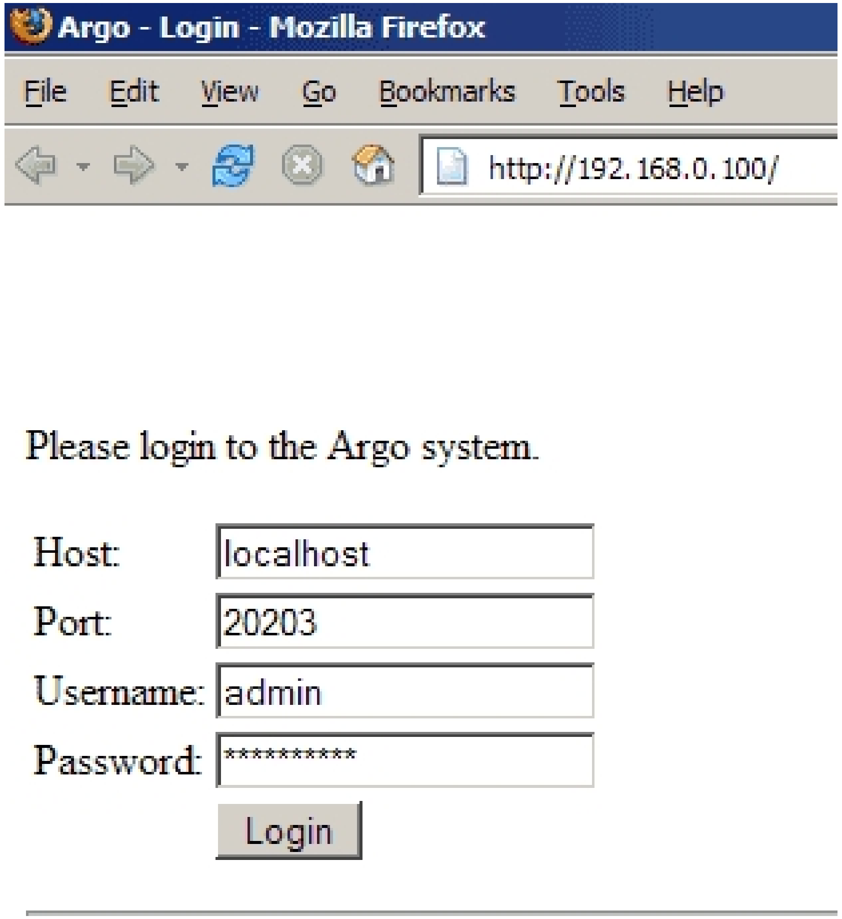
and restart Apache:

```
/etc/init.d/apache2 restart
```

Then we start Argo:

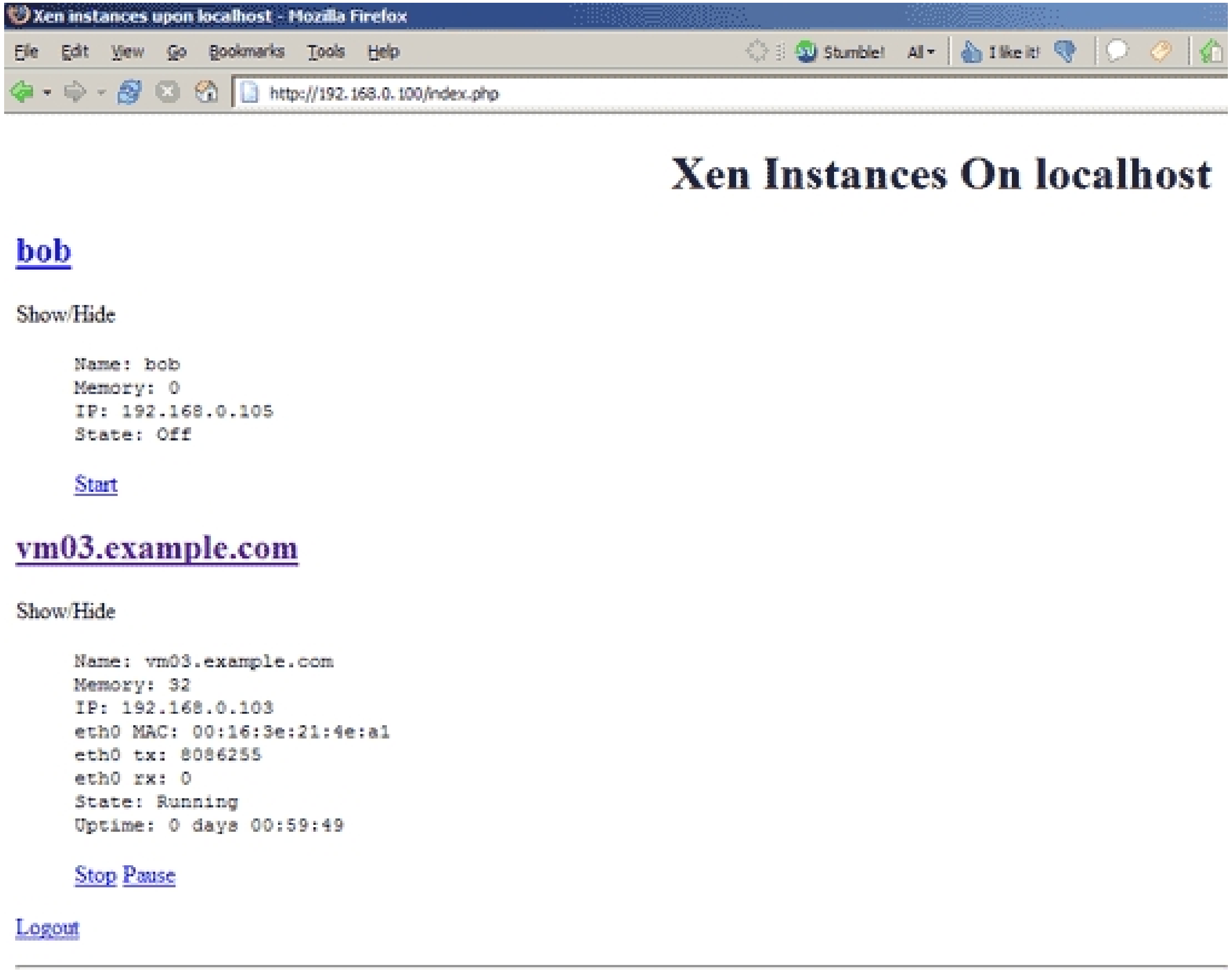
```
/etc/init.d/argo-server start
```

Now you can direct your browser to `http://192.168.0.100/` and log in. The web interface is pretty simple and straightforward:


Argo - Login - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://192.168.0.100/

Please login to the Argo system.

Host:	localhost
Port:	20203
Username:	admin
Password:	**********


Xen instances upon localhost - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
Stumble! All I like it!
http://192.168.0.100/index.php

Xen Instances On localhost

bob

Show/Hide

```
Name: bob
Memory: 0
IP: 192.168.0.105
State: Off
```

Start

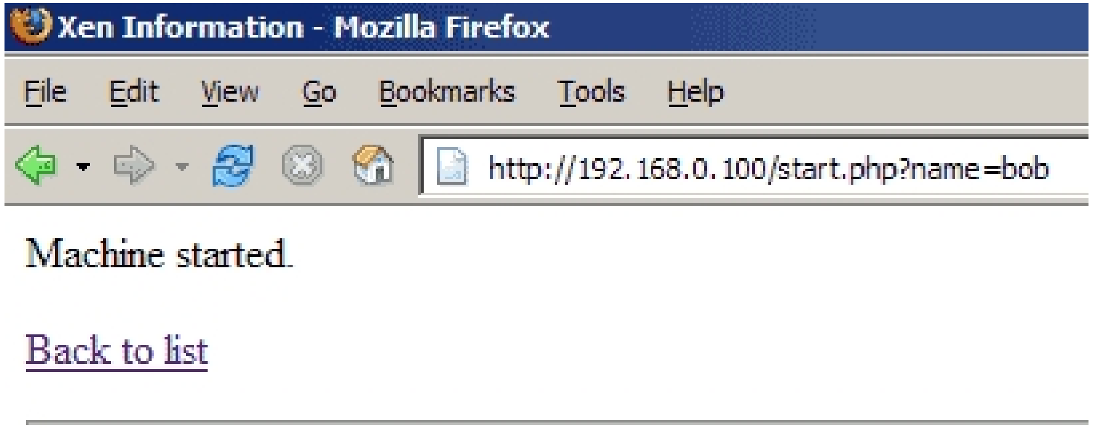
vm03.example.com

Show/Hide

```
Name: vm03.example.com
Memory: 32
IP: 192.168.0.103
eth0 MAC: 00:16:3e:21:4e:a1
eth0 tx: 8086255
eth0 rx: 0
State: Running
Uptime: 0 days 00:59:49
```

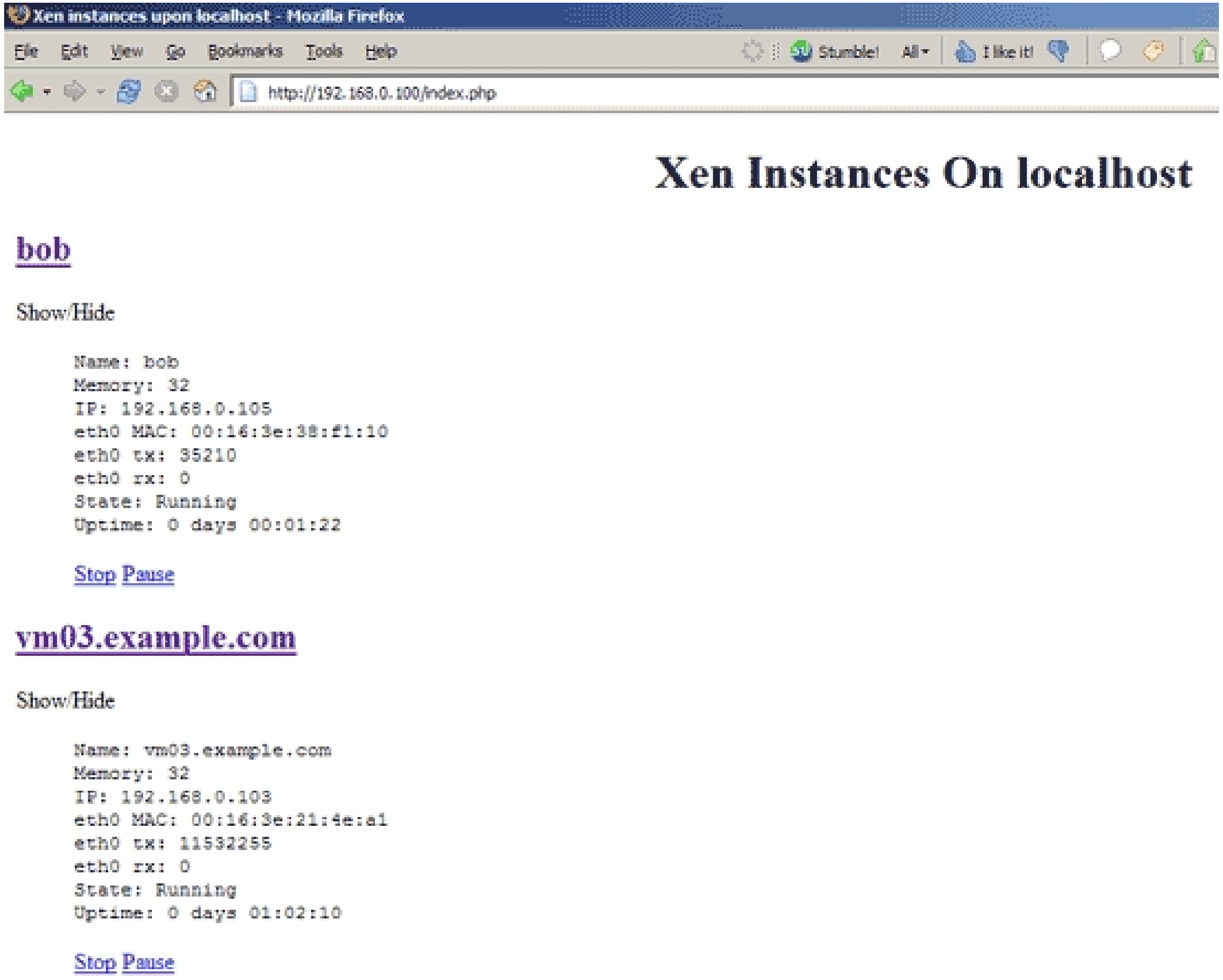
Stop Pause

Logout


Xen Information - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://192.168.0.100/start.php?name=bob

Machine started.

Back to list


Xen instances upon localhost - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
Stumble! All
I like it!
http://192.168.0.100/index.php

Xen Instances On localhost

bob

Show/Hide

```
Name: bob
Memory: 32
IP: 192.168.0.105
eth0 MAC: 00:16:3e:38:f1:10
eth0 tx: 35210
eth0 rx: 0
State: Running
Uptime: 0 days 00:01:22
```

Stop Pause

vm03.example.com

Show/Hide

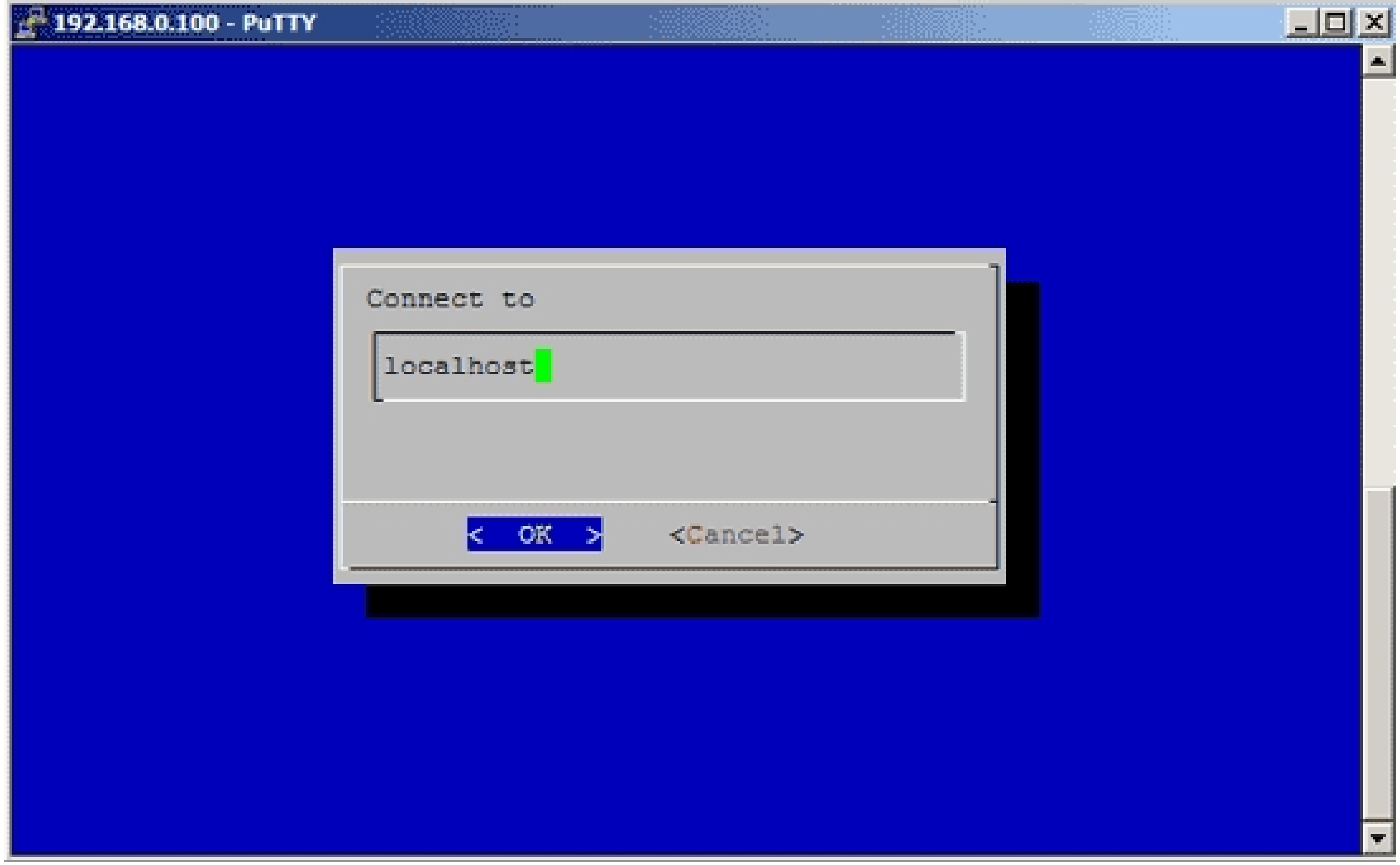
```
Name: vm03.example.com
Memory: 32
IP: 192.168.0.103
eth0 MAC: 00:16:3e:21:4e:a1
eth0 tx: 11532255
eth0 rx: 0
State: Running
Uptime: 0 days 01:02:10
```

Stop Pause

You can also invoke the Argo interface in the shell by running

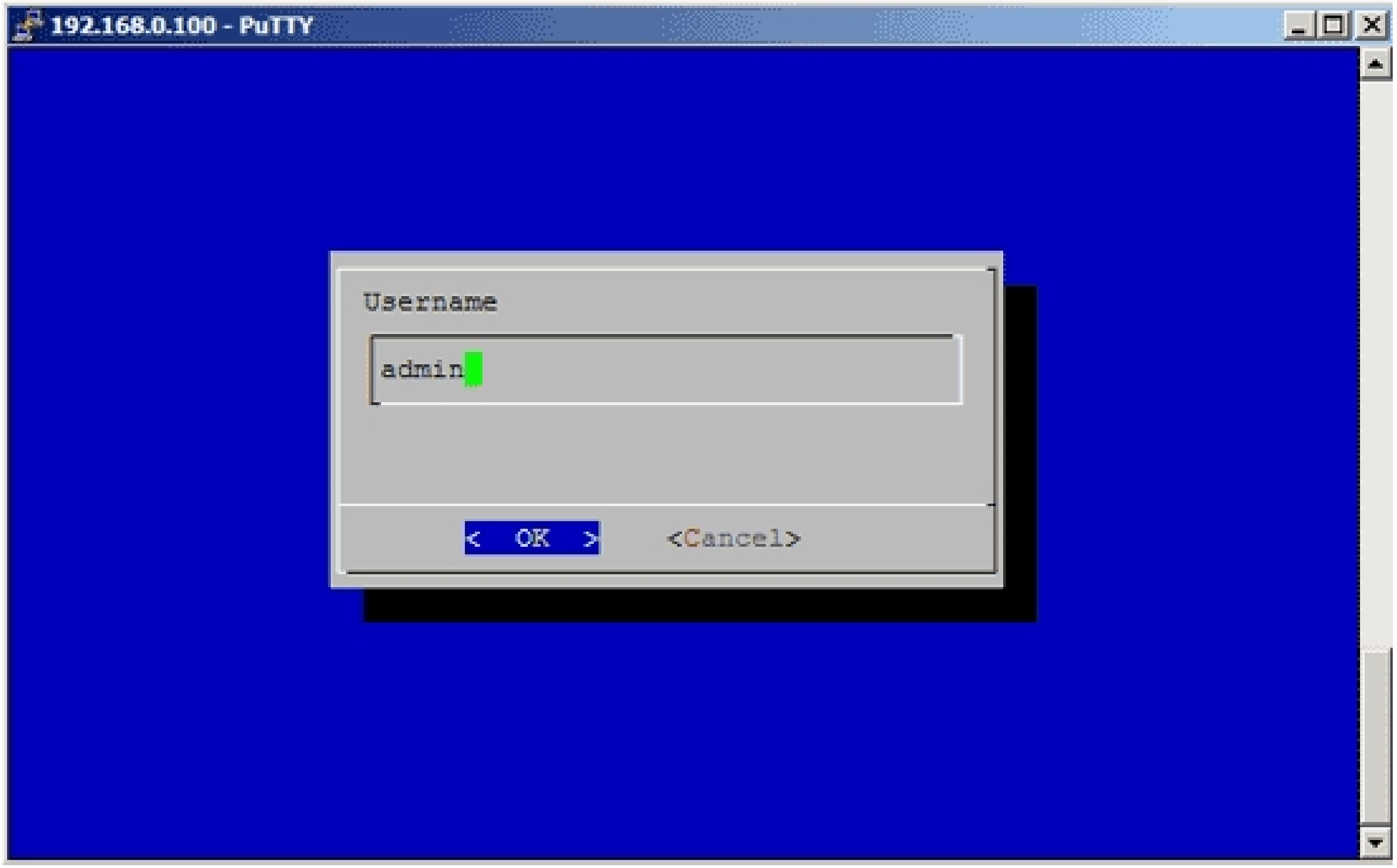
```
argo-dialog
```

The interface looks like this:



Connect to

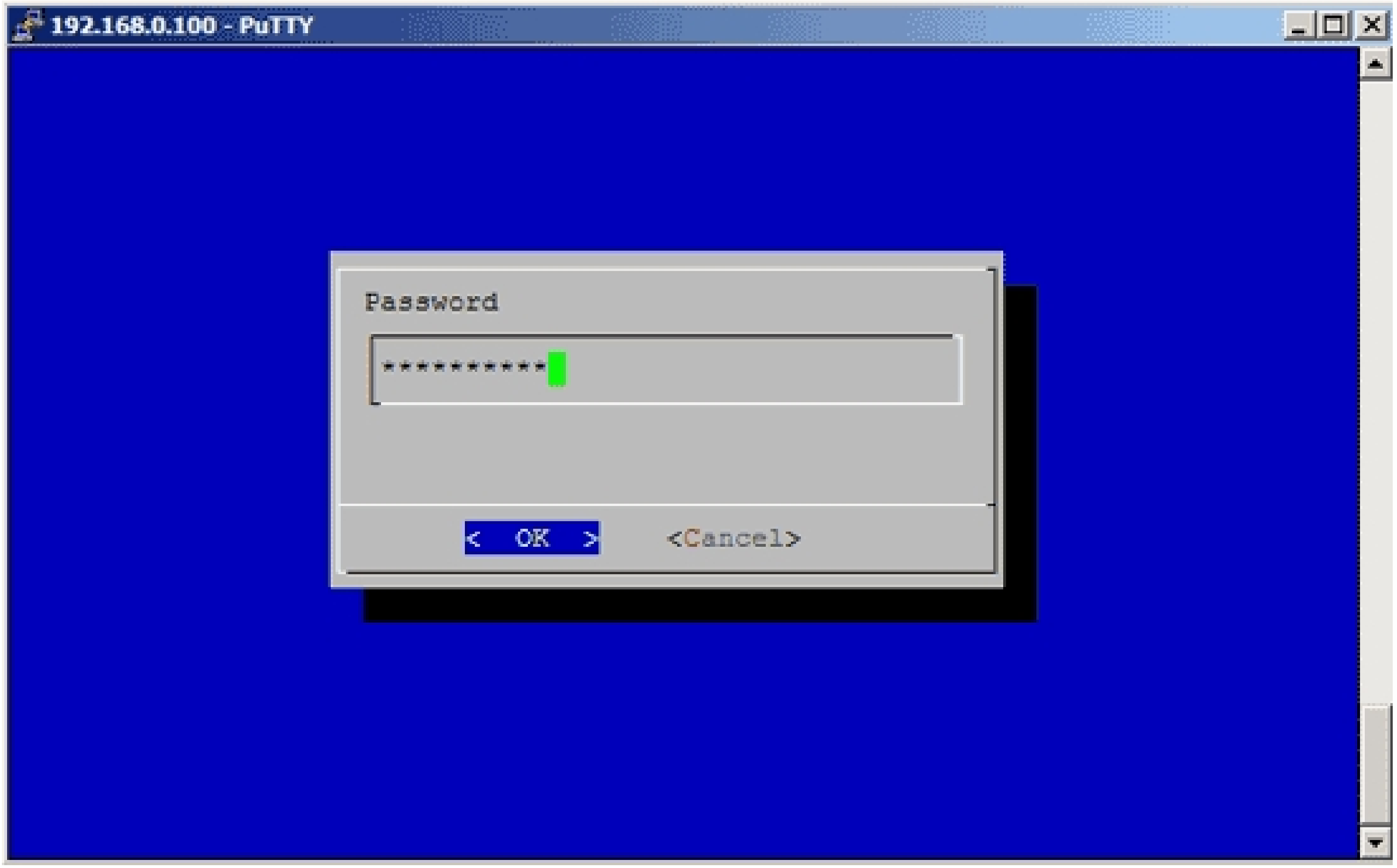
localhost

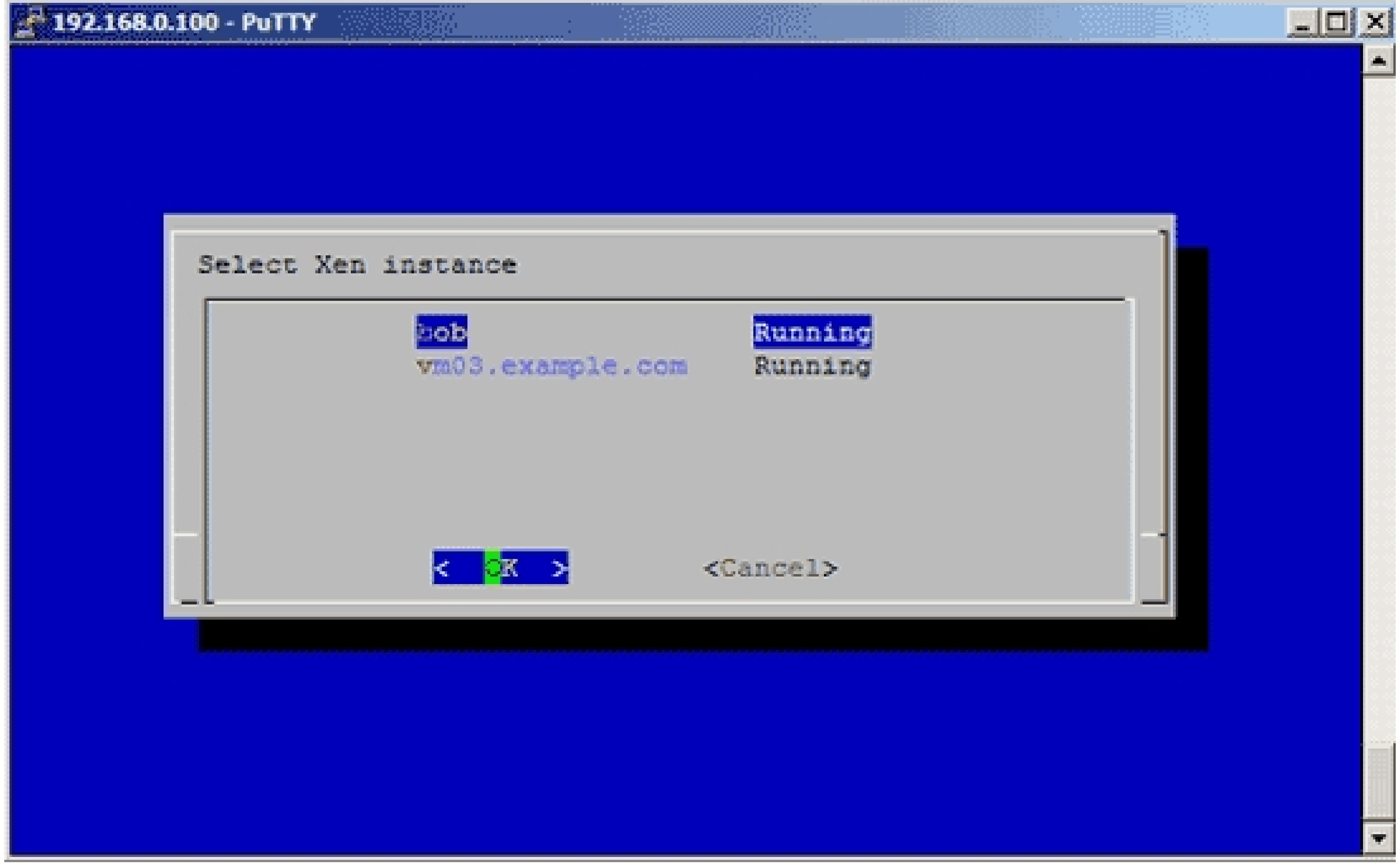


```
Username

admin

< OK >     <Cancel>
```

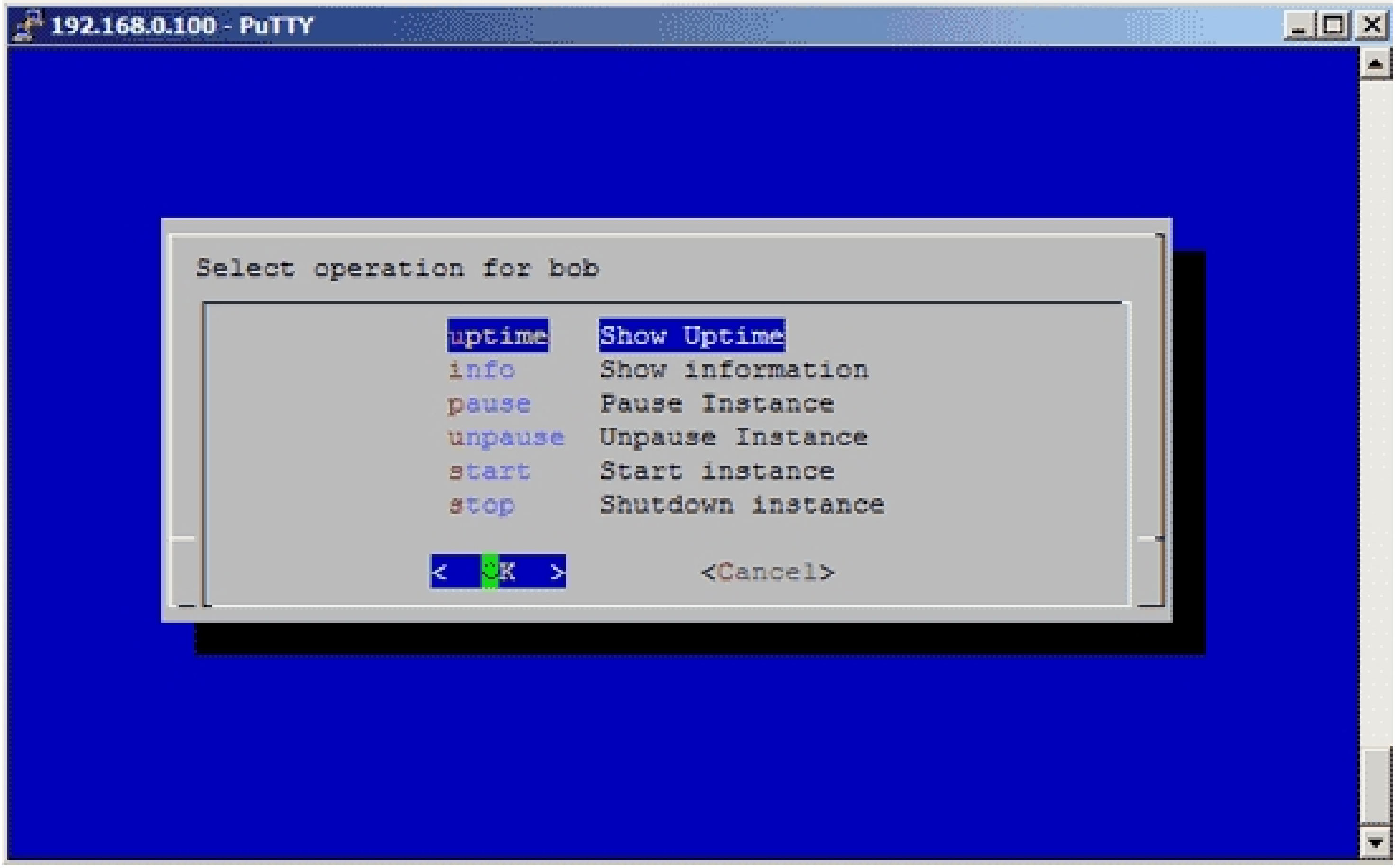

192.168.0.100 - PuTTY
Password
****
< OK >
<Cancel>



```
Select Xen instance

          bob                      Running
          vm03.example.com         Running

          <  OK  >          <Cancel>
```



```
Select operation for bob

uptime      Show Uptime
info        Show information
pause       Pause Instance
unpause     Unpause Instance
start       Start instance
stop        Shutdown instance
```

Have fun!

5 Links

- xen-tools: **http://xen-tools.org/software/xen-tools**
- xen-shell: **http://xen-tools.org/software/xen-shell**
- Argo: **http://xen-tools.org/software/argo**

- Xen: **http://www.xensource.com/xen**
- Debian: **http://www.debian.org**