Ecole d'ingénieurs et d'architectes de Fribourg
Hochschule für Technik und Architektur Freiburg

ViSaG - Virtual Safe GRID

POP-C++ Virtual-Secure (VS)

POP-C++ User and Installation manual : add-on

Author:
Valentin Clément

Date: April 7, 2011
Revision: 1.0.1

Contents

1 Introduction

POP-C++ Virtual Secure (abbreviated POP-C++ VS) is a version of the POP-C++ middle-ware able to run parallel objects into virtual machines with a certain level of security for the communications between the nodes. The development of this version is the core of the ViSaG project. A prototype version of Virtual - POP-C++ has been developed during a bachelor thesis project in the summer 2010. This version is called VirtualPOPC-1[1].

This document is an add-on of the official POP-C++ User and Installation Manual[2] to help the end user to set up a node with POP-C++ VS.

This document is structured as follows:

- The next chapter explains the confidence link used between the nodes in a POP-C++ network.
- The third chapter aims to help the user to set the Admin Virtual Machine (Admin VM) and the original Worker Virtual Machine (Worker VM).
- The fourth chapter explains the installation process of the original Worker Virtual Machine (Worker VM).
- The fifth chapter is a glossary of the terms used in POP-C++ VS.

2 Developer Notice

POP-C++ VS must be used only with the Global Services. This forces tu use the resource discovery to create parallel object. We can't use object description like **od.url** as used in the POP-C++ standard version.

3 The confidence link

In order to be able to communicate between two POP-C++ nodes in secure mode, we need to set up the confidence link. The confidence link means that two POP-C++ nodes agree to communicate with each other with a certain level of security.

For the moment, the POP-C++ confidence link is set up with the exchange of the SSH Public Key. To set this confidence link between the computer A and the computer B, we should do the following actions :

1. Generate the SSH Public and Private keys on each computer (command ssh-keygen). We will use the default option to generate the SSH keys.
2. Send the SSH Public Key of A to B (use scp or another way to transfer the file from the computer A to the computer B). The SSH Public is located under $HOME/.ssh/id_rsa.pub
3. Send the SSH Public Key of B to A.
4. Write the SSH Public Key of A in the authorized_keys file of B (located under $HOME/.ssh/authorized_keys)
5. Write the SSH Public Key of B into the authorized_keys file of A.

Be sure to check your confidence link to avoid any problems during the execution of a POP-C++ application.

4 Set up a virtual node

A node using POP-C++ VS is composed by minimum two virtual machines over an ESXi virtualization platform. These two virtual machines are called as follow :

- **Admin VM** : This virtual machine will run the POP-C++ Global Services for the whole ESXi node. A node should have only one Admin VM (see Section 4.1 to set up this VM).
- **Worker VM** : This virtual machine will be managed by the Admin VM and will execute POP-C++ jobs (see Section 4.2 to set up this VM).

The following of this chapter explains how to set up these two VM on a ESXi virtualization platform.

4.1 Set up the Admin VM

The first VM to set up in the POP-C++ VS environement is an Admin VM. The Admin VM will interact with the ESXi hypervisor to manage the Worker VM. To set up this particular VM, the following steps must be done:

- Install ESXi on the computer (see Appendix E)
- Create a VM in vSphere (see Appendix F)
- Install an OS in the created VM.
- Install necessary packages for POP-C++ (see Section 4.1.1)
- Configure the DNS hostname (see Section 4.1.2)
- Configure the SSH server (see Section 4.1.3)
- Configure, Compile and Install POP-C++ (see Section 4.1.4)

4.1.1 Install necessary packages for POP-C++

In order to install POP-C++ VS, we need to install some additional packages. The following packages need to be installed and the default values can be used :

1. libvirt 0.8.5 or later (see Appendix A)
2. VMware tools (see Appendix B)
3. VMware CLI 4.1 or later (see Appendix C)
4. VIX 10.1 or later (see Appendix D)
5. A C++ compiler, Zlib

For the point 5, the following command can be used on a Debian/Ubuntu based OS.

```
sudo apt-get install build-essential zlib1g-dev
```

4.1.2 Set the ESXi DNS hostname

If your network does not have a DNS server or if the hostname of the ESXi platform is not registered in it, we need to add this hostname in the host file name of the Admin VM. On a linux based OS, we will modify the file located under **/etc/hosts** and add the following line (replace with your values):

```
160.98.20.140    esxivisag01.sofr.hefr.lan
```

4.1.3 Configure SSH Server

As the Admin VM will communicate with worker VM and these Worker will be created dynamically, we need to configure the SSH server to avoid a strict host key identification. To do that, we need to modify the file **/etc/ssh/ssh_config** and have the following line:

```
StrictHostKeyChecking no
```

We need to generate the SSH public/private keys to be able to run Virtual Secure POP-C++. To generate the SSH keys, we need to run the following command (use the default parameters):

```
ssh-keygen
```

4.1.4 Configure and compile POP-C++ in its Virtual Secure version

POP-C++ needs to be configured and compiled before its installation. During the configuration, we can choose the version of POP-C++ that we want to install. For the ViSaG project, we want to install POP-C++ VS. To have this version, use the following options with the configure script:

```
./configure --enable-virtual --enable-secure
```

NOTE: If you already have compiled POP-C++ before, use "make clean" to clean the workspace before compiling the new version.

Once the configuration process is done, we can compile POP-C++. For this, use the following command:

```
make
```

If the make command exit without errors, POP-C++ is compiled in its VS version and ready to be installed.

To install POP-C++, we just need to launch the following command:

```
make install
```

The installation script will ask us some question to configure our installation. Here are the different questions and the answer we could provide:

Part 1:Select the right installation

```
DO YOU WANT TO CONFIGURE  POP–C++ SERVICES? (y/n)
y
...
DO YOU WANT TO MAKE A SIMPLE INSTALLATION? (y/n)
n
```

Part 2: Configure the parameters of the node
The first question of this part asks us to enter the "full qualified master host name". This means the hostname or IP address of the node which have the confidence link with this node.

```
Enter the full qualified master host name (POPC gateway):
```

Not used in POP-C++ VS for the moment (let blank)

```
Enter the child node:
```

Enter the number of processors available on the whole node.

```
Enter number of processor available (default:1):
```

Set the number of jobs that can be concurrently executed on the node for all VM.

```
Enter the maximum number of POP–C++ jobs that can run concurrently
(default:100):
20
```

Set the available RAM for the whole node.

```
Enter the available RAM for job execution in MB (default: 1024):
4096
```

Set the user you want to use to execute the jobs (usually the same as the one who installed POP-C++, let blank)

```
Which local user you want to use for running POP–C++ jobs?
```

```
CONFIGURE THE RUNTIME ENVIRONMENT
Enter the script to submit jobs to the local system:

Communication pattern:
```

Part3: Setting up the virtual environment
It's time to set up the virtual environment for POP-C++. We need the information about the ESXi platform and the worker VM. If the worker VM is not set up yet, these information can be edited later in the file POPC_LOCATION/etc/virtual.conf.

```
SETTING UP VIRTUAL ENVIRONMENT INFORMATION NOW
ESX(i) hypervisor IP address : e.g. 160.98.20.140
160.98.20.141
ESX(i) user name with admin rights (default: root)
root
ESX(i) password:
*****
ESX(i) datastore name (default: datastore1)

ESX(i) maximum worker (default: 4)

ESX(i) worker name (default: popc_worker_guest1)
visag04_worker1
ESX(i) worker OS username
visag
ESX(i) worker OS password
*****
ESX(i) clean snapshot name (default: popc_clean)
popc_clean
```

Part 4: runtime environment variables
We can set up some environment variables to be used in the runtime of POP-C++. No particular variables needs to be set for POP-C++ VS.

```
SETTING UP RUNTIME ENVIRONMENT VARIABLES
Enter variable name:

Enter variable value:
```

Part 5: Installation end
It's important to generate the startup script because it will be used to run the POP-C++ Global Services on the Admin VM.

```
==================================
CONFIGURATION POP-C++ SERVICES COMPLETED
==================================
Do you want to generate the POP-C++ startup scripts? (y/n)
y
Enter the service port[2711]:

Enter the domain name:

Enter the temporary directory for intermediate results:

==================================
CONFIGURATION DONE!
==================================

IMPORTANT : Do not forget to add these lines to your .bashrc file or
equivalent:

POPC_LOCATION=/home/visag/popc
PATH=$PATH:$POPC_LOCATION/bin:$POPC_LOCATION/sbin
```

4.1.5 Run the POP-C++ services on the Admin VM

After the preparation and installation of POP-C++ VS on the Admin VM, we can now start the Global Services on the node. Use this command to start the POP-C++ Global Services :

```
SXXpopc start
```

You will see the followings lines if the startup is successful (the IP addresses and ports will be different):

```
Starting POP-C++ [Virtual Secure Version] Global Services
VSPSN Started [socket://160.98.21.195:49391]
VPSM Started [socket://160.98.21.195:37958]
POPCloner Started [socket://160.98.21.195:52035]
VSJM created [socket://160.98.21.195:2711]
```

If an exception occurs at start up, please check one of the following log files to find the problem:

- /tmp/popc_node_log
- /tmp/popc_security_log
- /tmp/popc_clone_log

NOTE: The /tmp directory is the default log files directory you may have changed it with another directory

4.2 Set up the original Worker VM

As POP-C++ VS will clone workers as needed, we just need to set up the first "original" Worker VM. To set this particular worker VM, the following steps must be done :
WARNING: Read carfully the restriction imposed to the name, username and password. These restrictions are strict and cannot be worked around.

- Create VM in vSphere (see Appendix F). The VM name **must** have the suffix **_worker1** (other suffixes are not supported for the moment).
- Install an OS on the create VM.
- Install necessary packages for POP-C++ (see Section 4.2.1)
- Install VMware tools on the worker VM (see Appendix B)
- Configure SSH for the worker VM (see Section 4.2.2)
- Configure, Compile and Install POP-C++ Standard (see Section 4.2.3)
- Create a snapshot of the worker VM (see Appendix G)

4.2.1 Necessary packages

To be able to run application using POP-C++ VS on the worker VM, we need to install some packages. Here are the list of the needed packages :

- A C++ compiler
- An SSH server
- Zlib compression library

On a Debian/Ubuntu based OS, we can use the following command to install those packages.

```
sudo apt-get install build-essential openssh-server zlib1g-dev
```

4.2.2 Configure SSH server on the worker VM

As POP-C++ VS uses the secure mode of POP-C++ to communicate between the nodes, we need to set up the SSH parameters. First, we need to generate the SSH public/private keys pair. Use the following command and keep the default values.

```
ssh-keygen
```

We also need to let the SSH server accept host key without a strict check. For this, we need to modify the file **/etc/ssh/ssh_config** and have the following line :

```
StrictHostKeyChecking no
```

4.2.3 Configure, Compile and Install POP-C++

On the Worker VM, we just need a standard version of POP-C++. Here are the commands to execute to configure and compile POP-C++ in its standard version.

```
./configure
make
```

Once POP-C++ is compiled, we can install it. Use the following command :

```
make install
...
DO YOU WANT TO CONFIGURE  POP-C++ SERVICES? (y/n)
y
DO YOU WANT TO MAKE A SIMPLE INSTALLATION? (y/n)
y
...
=================================
CONFIGURATION DONE!
=================================

IMPORTANT : Do not forget to add these lines to your .bashrc file or
equivalent:

POPC_LOCATION=/home/visag/popc
PATH=$PATH:$POPC_LOCATION/bin:$POPC_LOCATION/sbin
```

NOTE: Don't forget to add the lines in your .bashrc or equivalent file if you changed the default installation location otherwise the node will not run normally.

5 Compatibility issues

As POP-C++ can be compiled in 4 different versions, there are some compatibility issues between those versions. The table below indicates which version can be used together.

Version/Version	Standard	Secure	Virtual	Virtual-Secure
Standard	OK	NO OK	OK	NO OK
Secure	NO OK	OK	NO OK	OK
Virtual	OK	NO OK	OK	NO OK
Virtual-Secure	NO OK	OK	NO OK	OK

NOTE: Be aware that an application compiled with POP-C++ Secure or Virtual-Secure will not run with POP-C++ Standard or Virtual. You need to recompile the application before running it.

6 Functional test procedure

This chapter will help you to set up an infrastructure with one or several POP-C++ node and test it.

6.1 Test a single node

The first thing to test is the single node installation. This test is done just to make sure that the Admin VM and the Worker VM are configured correctly to work together.

Start the Global Services on the Admin VM
First we need to start the POP-C++ Global Services on the Admin VM to be able to run a POP-C++ application. For this, use the following command:

```
SXXpopc start
```

You should have a result as follows:

```
Starting POP-C++ [Virtual Secure Version] Global Services
VSPSN Started [socket://160.98.21.238:54314]
VPSM Started [socket://160.98.21.238:44603]
POPCloner Started [socket://160.98.21.238:34485]
VSJM Started [socket://160.98.21.238:2711]
```

If you already have an error at this step, check the virtual configuration file located under *POPC_LOCATION* /etc/virtual.conf and be sure to have set the right parameters for your ESXi hypervisor and VMs.

Once the POP-C++ Global Service are started, we are ready to run a POP-C++ application on the node. For this, go to *POPC_RELEASE_FOLDER*/demos/demopopc. We need to compile the code.

```
make
```

Once the code is compiled, we need to provide a way to find this executable code. The easiest way is to put the code on a Web Server or FTP Server. The second way is to have a NFS. But for the second solution, we have to configure the Admin VM and Worker VM with the same NFS.

The most important is that the Worker VM can download the code to execute it. In our demo example, the code is **demopopc.obj**. This file must be accessible for the Worker VM.

Once the executable is accessible from somewhere, we need to edit the object map file. In our example, this file is named **obj.map**. Depending on the architecture, and location of the executable, this file could look like this.

```
POPCobject i686-pc-Linux http://www.popcwebserver.com/example1/demoobj.obj
```

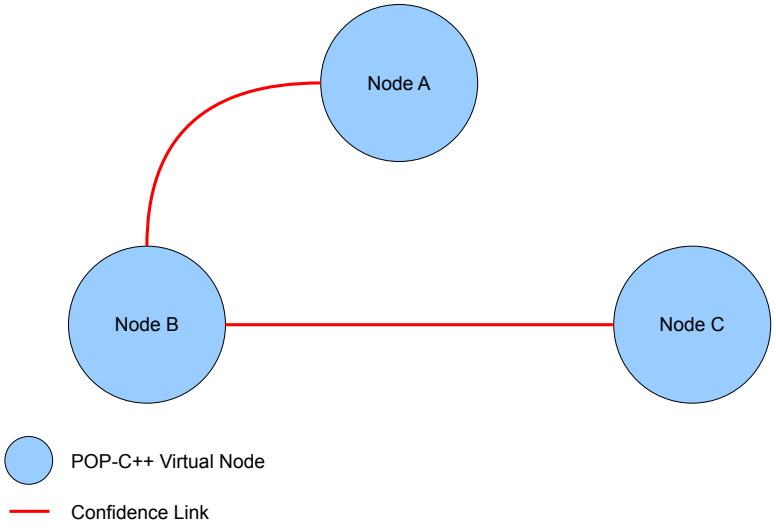
6.2 Test several node

To test the execution on several nodes, we first need to set up an infrastructure of POP-C++ nodes. We will set up 3 POP-C++ nodes to test two different aspects of POP-C++ VS.

6.2.1 Object reference passing test

Infrastructure schema
Here is the connection schema of the infrastructure we would like to test. It's important to have two node that doesn't know each other.

Figure 1: Test Case 1: Logical Connection



Infrastructure configuration

- Node A has no master nodes. Node A has the PKI of Node B. The number of job is limited to 1 during the installation.
- Node B has Node A as master node. Node B has the PKI of Node A and Node C. The number of job is limited to 1 during the installation.
- Node C has Node B as master node. Node C has the PKi of Node B. The number of job is limited to 2 during the installation.

The Node A must be started first, Node B second and Node C after.

Compilation and execution of test program
To test this specific case, we will use the program located under *POPC_RELEASE_FOLDER*/example/multiobj/. To compile this program, use the following command:

```
make
```

As for the single node test, we need to provide a way to find the executable code. The file "demopopc.obj" must be uploaded on a Web or FTP server or located on a NFS drive. Once the file is accessible by and VM in the network, we must edit the obj.map file. Here is a sample of this file.

```
POPCobject i686-pc-Linux http://www.popcwebserver.com/example1/demoobj.obj
```

Once the program is compiled and the executable accessible, use this command to run the program:

```
popcrun obj.map ./main
```

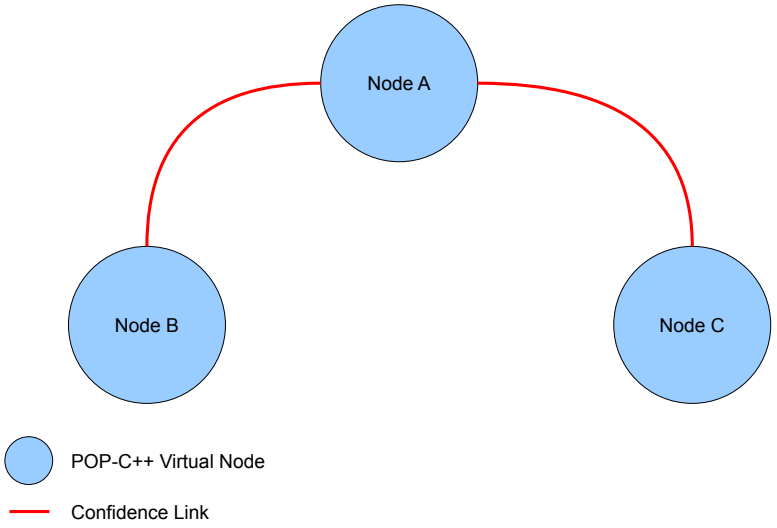
Three Worker VM must be started on the three nodes. The program should end successfully.

6.2.2 Decentralized object creation test

In this first test, we will check the creation of object by different nodes.

Infrastructure schema
Here is the connection schema of the infrastructure we would like to test. It's important to have one node between the first node (Main Node) and the last node.

Figure 2: Test Case 2: Logical Connection



Infrastructure configuration

- Node A has no master nodes. Node A has the PKI of Node B and Node C. The number of job is limited to 1 during the installation.
- Node B has Node A as master node. Node B has the PKI of Node A. The number of job is limited to 1 during the installation.
- Node C has Node A as master node. Node C has the PKi of Node A. The number of job is limited to 1 during the installation.

The Node A must be started first and Node B and C after.

Compilation and execution of test program
To test this specific case, we will use the program located under *POPC_REALEASE_FOLDER*/demos/demopopc/. To compile this program, use the following command:

```
make
```

Once the program is compiled, use this command to run the program:

```
popcrun objmap ./main 3 - - -
```

Three Worker VM must be started on the three nodes. The program should end successfully.

6.3 Test conclusion

If the three tests have been executed successfully, we are now sure that our POP-C++ infrastructure is ready to execute our own program.

7 New options

The SXXpopc has two new options. The first one is "kill" to kill all the POP-C++ Global Services. The second one is "vmstate" to give the state of the VM on a specific node.

kill
The option "kill" does not take any parameters:

```
SXXpopc kill
```

vmstate
The option "vmstate" takes three parameters:

- IP address: The IP address of the ESXi hypervisor
- Username: The username used to connect to the hypervisor
- Password: The password use to connect to the hypervisor

```
SXXpopc vmstate 160.98.20.142 root myPassword
```

8 Glossary

- **POP-C++** : refers to POP-C++ in general or to the standard version of POP-C++.
- **POP-C++ S** : refers to POP-C++ Secure version.
- **POP-C++ V** : refers to POP-C++ Virtual version.
- **POP-C++ VS** : refers to POP-C++ Virtual-Secure version.
- **Admin VM** : refers to the Admin Virtual Machine managing the POP-C++ Virtual Node.
- **Worker VM** : refers to the Worker Virtual Machine managed by the Admin VM.
- **PSN** : POP-C++ Search Node.
- **VPSN** : Virtual POP-C++ Search Node.
- **SPSN** : Secure POP-C++ Search Node.
- **VSPSN** : Virtual Secure POP-C++ Search Node.
- **PSM** : POP-C++ Security Manager.
- **VPSM** : Virtual POP-C++ Security Manager.
- **JM** : Job Manager.
- **VJM** : Virtual Job Manager.
- **SJM** : Secure Job Manager.
- **VSJM** : Virtual Secure Job Manager.

9 Table of figures

10 References

[1] Adrian Wyssen, *VirtualPOPC-1 : Project Report*. EIA-FR, Switzerland, June-August 2010.

[2] The POP-C++ Team, *Parallel Object Programming C++, User and installation Manual*. Grid and Cloud Computing Group, EIA-FR, Fribourg, Switzerland, 2010.

[3] Valentin Clément, *POP-C++ over SSH Tunnel*. EIA-FR, Switzerland, September-November 2010.

A libvirt 0.8.7 installation

POP-C++ Virtual Secure uses libvirt to interact with the ESXi platform. We need to install libvirt before compiling and installing POP-C++.

The latest version of libvirt is available at : http://www.libvirt.org

To install libvirt, we need first to install some packages. On a Debian/Ubuntu Os we can use apt-get to install the following packages:

```
sudo apt-get install libxml2 libxml2-dev libgnutls-dev libdevmapper-dev
libcurl4-openssl-dev libvirt-dev libnl-dev
```

After the installation of these packages, we need to install the libvirt distribution downloaded from the official libvirt website. To configure libvirt with the ESXi driver use the following command:

```
./configure --with-esx=yes
```

We can make sure that the ESX driver support is enabled by checking the end of the configure script output. We should have the following line:

```
configure: ESX: yes
```

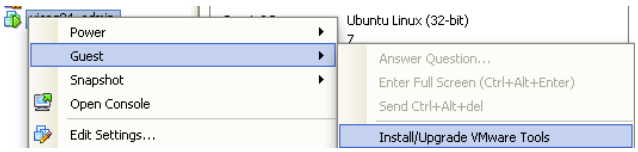
Once the configure process is done, we can compile and install libvirt. Use the following commands to do that.

```
make
sudo make install
```

B VMware Tools 8.3.2 installation

To install the latest VMware tools on the VM, we need to use the vSphere Client. On the left side, we need to right-click on the VM and select Guest > Install/Upgarde VMware Tools (see Figure 3). This action will connect a virtual CD-ROM on the VM.

Figure 3: VMware Tools Install (Step 1)



Once the CD-ROM is connected to the VM, we need to mount it. Use the following command to mount it and to launch the installation process. We do a default installation.

```
sudo mount /dev/cdrom /media/cdrom
cp /media/cdrom/VMwareTools-8.3.2-257589.tar.gz ./
tar -xvf VMwareTools-8.3.2-257589.tar.gz
cd vmware-tools-distrib
sudo ./vmware-install.pl
```

C VMware CLI 4.1 installation

The VMware CLI are used by POP-C++ to do some action that are not currently possible with libvirt such as the cloning process.

The latest version of the VMware CLI can be found at : http://www.vmware.com/support/developer/vcli/

To install the VMware CLI, we need first to install some packages. On a Debian/Ubuntu Os we can use apt-get to install the following packages:

```
sudo apt-get install libcrypt-ssleay-perl libssl-dev libxml-libxml-perl
```

After the installation of these packages, we can install the VMware CLI itself. To do it, use the following commands:

```
tar -xvf VMware-vSphere-CLI-4.1.0-254719.i386.tar.gz
cd vmware-vsphere-cli-distrib
sudo ./vmware-install.pl
```

Just follow the installation script and enter default values.

D VIX 1.10.2 installation

The VMware VIX API is used by POP-C++ for some action that are not currently available in libvirt such as sending file to a VM or executing command on VM.

The latest version of VIX can be found at : http://www.vmware.com/support/developer/vix-api/
To install the VIX API, we just run the bundle with the following command:

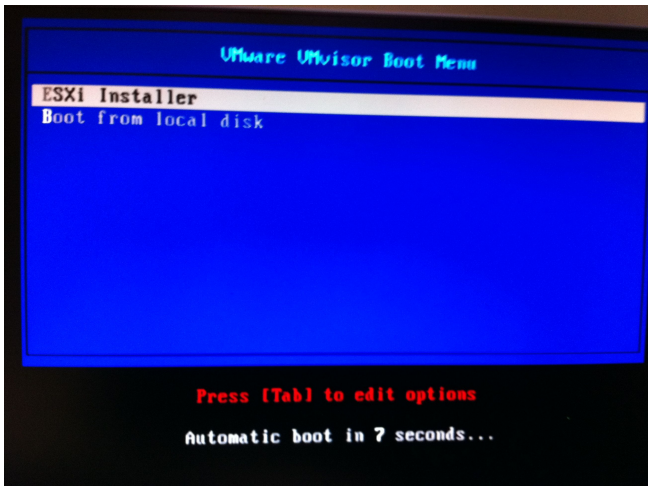
```
sudo sh VMware-VIX-1.10.2-331863.i386.bundle
```

E VMware ESXi 4.1

The first step is to install the ESXi hypervisor on your hardware if it's not done already. Be sure that your hardware is listed in the "Hardware Compatibility Guide" (http://www.vmware.com/resources/compatibility/search.php)

The latest version of VMware ESXi can be found at: http://www.vmware.com/products/vsphere-hypervisor/
Step 1:
We first need a bootable CD-ROM of ESXi 4.1 or later. Once the CD-ROM is in the computer and the computer is started, the "VWware ESXi Boot Menu" will appear. At this step, we need to choose to boot from the ESXi Installer.

Figure 4: VMware ESXi Boot Menu

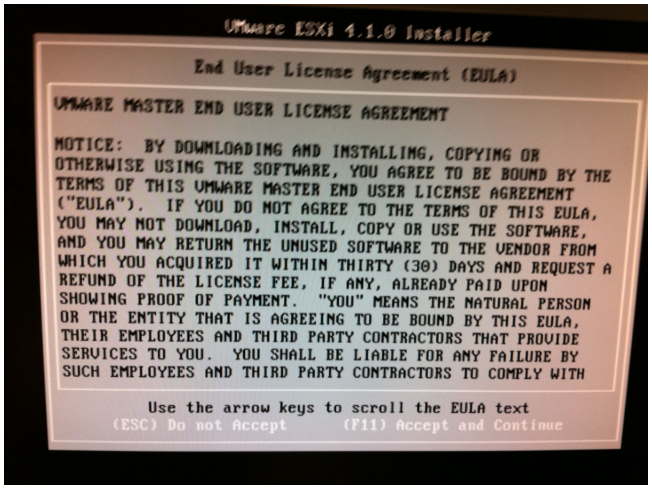


Step 2:
Once the installer has started, the welcome screen will be displayed. On this screen, we would choose the "Install" option by pressing the "Enter" key.

Figure 5: VMware ESXi Installation Welcome screen

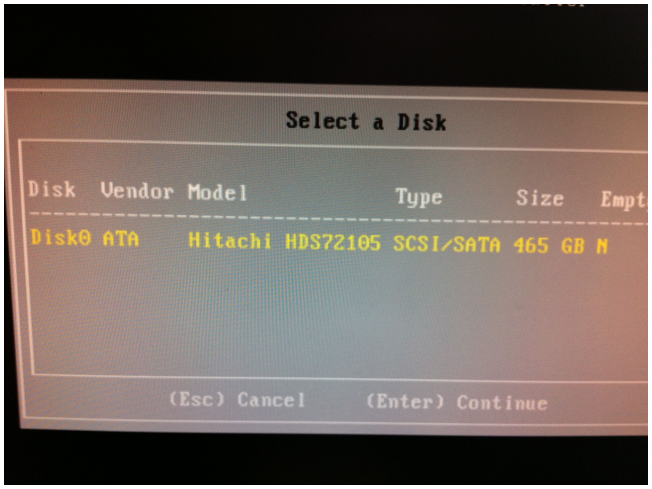
Step 3:
We will need to agree the licence of VMware ESXi by pressing the "F11" key.

Figure 6: VMware ESXi licence agreement



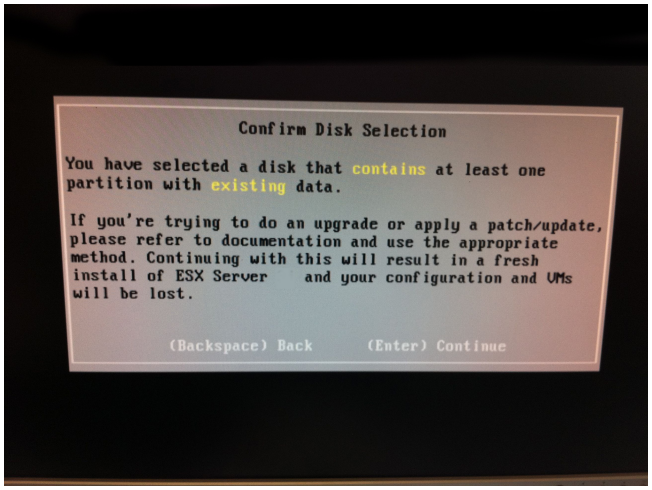
Step 4:
In this step, we will need to choose the disk to install VMware ESXi. We choose the correct destination disk and press "Enter"

Figure 7: VMware ESXi disk selection



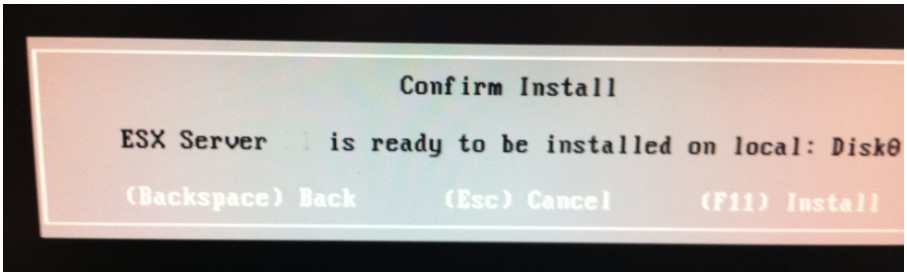
Step 5: (optional)
The installation program may ask if we want to erase the data containing on the destination disk. We will erase the data by pressing the "Enter" key.

Figure 8: VMware ESXi disk selection (confirmation)



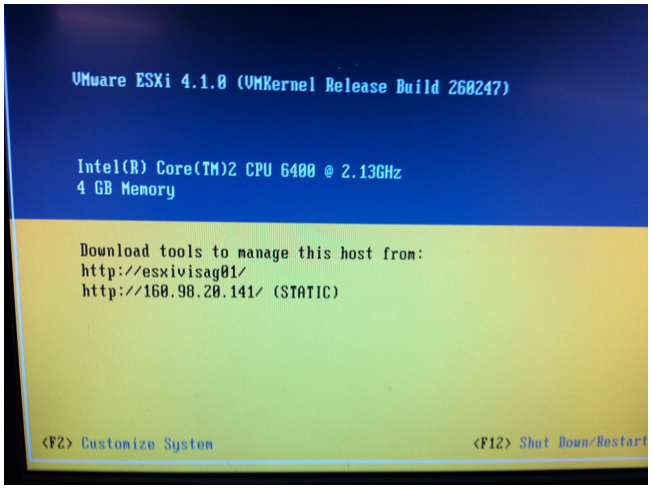
Step 6:
We are ready to launch the installation of VMware ESXi on the computer. The installation program will ask us to confirm the installation by pressing the "F11" key.

Figure 9: VMware ESXi Installation confirmation



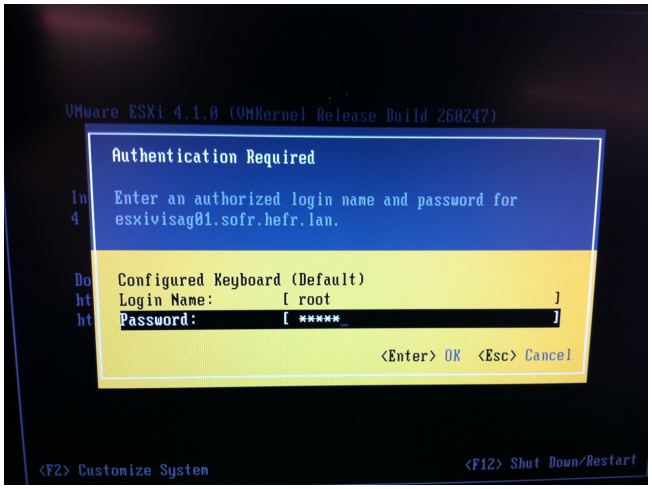
ESXi configuration As the ESXi platform will act as a server, we will need to configure the network parameter and the credentials. When the ESXi system is started, the first screen looks like Firgure 10. To be able to change any system parameters, we need to press the "F2" key.

Figure 10: VMware ESXi Startup screen



If a password is already set up, a login screen will appear like in Figure 11. Just enter the password and press the "Enter key".

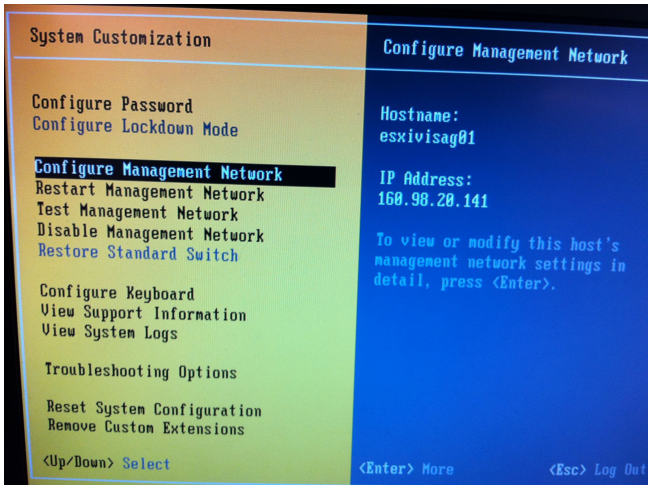
Figure 11: VMware ESXi Login screen



Network configuration

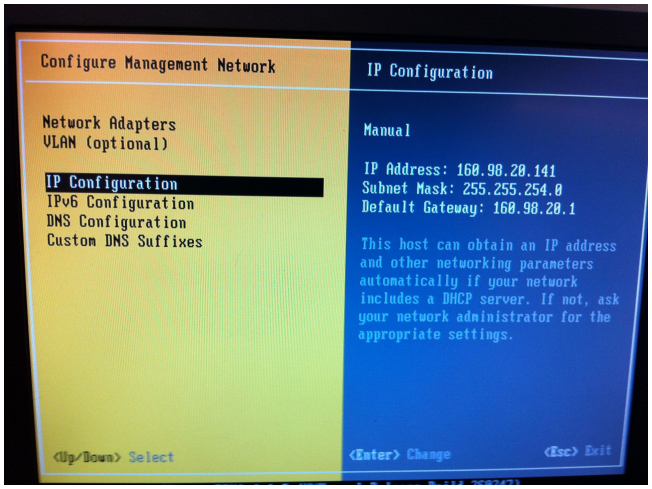
The ESXi platform needs a static IP address to work well with POP-C++ Virtual. Once logged into the ESXi system, we can modify the IP parameter in the "Configure Management Network" section (see Figure 12).

Figure 12: VMware ESXi System configuration



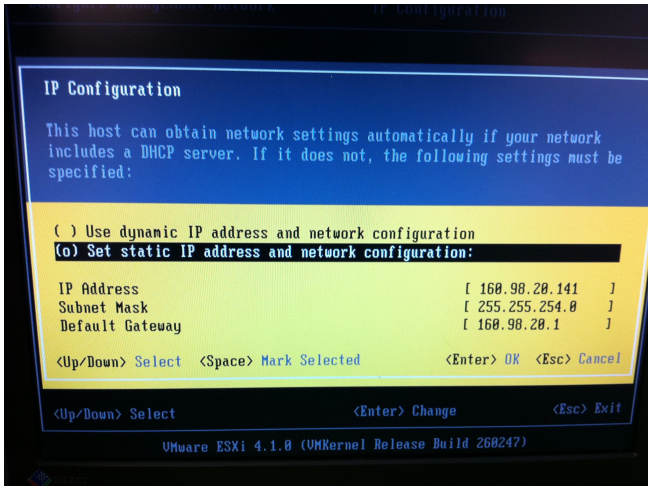
In the "Configure Management Network" section, we can select the "IP Configuration" subsection to modify the IPv4 parameter (see Figure 13).

Figure 13: VMware ESXi Network Management Configuration



Once in the IP Configuration subsection, we can select "Set static IP address and network configuration" and fill the fields below (see Figure 14). When the information are set, we can save them by pressing the "Enter" key.

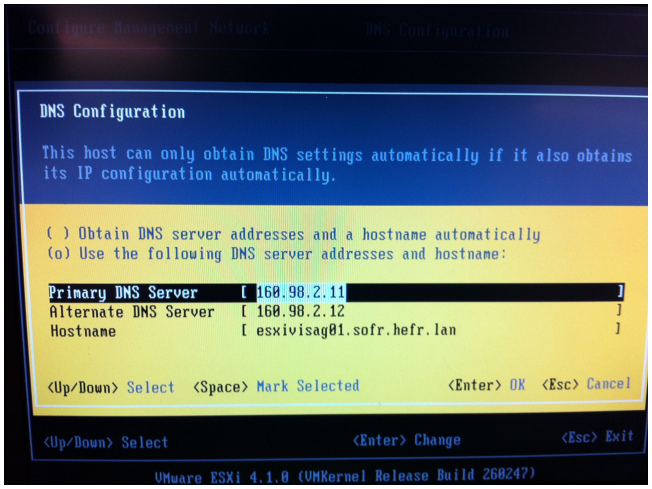
Figure 14: VMware ESXi IPv4 Configuration



After setting the IPv4 parameters, we need to set up the DNS parameter. In the "Configure Management Network", we will choose "DNS configuration" instead of "IP configuration" (see Figure 13). Like in Figure 15, we select "Use the following DNS server addresses and hostname" and fill the fields.

NOTE: This step is not mandatory if you have a unique DNS name attributed by IP address. But your ESXi host must be reachable by its DNS hostname.

Figure 15: VMware ESXi System configuration



Password configuration

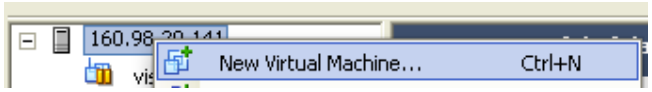
To set or change the administrator password, we will choose "Configure Password" in the ESXi System Configuration (see Figure 12). A screen will be prompted (Figure 16) and we just need to fill it.

Figure 16: VMware ESXi System configuration

F Create a VM with vSphere 4.1

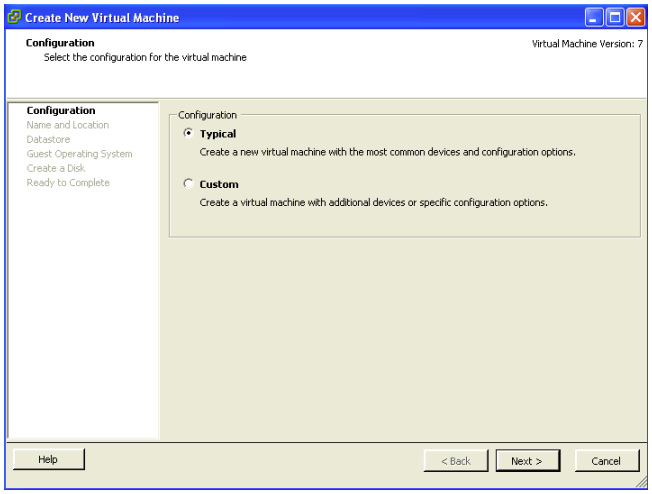
Before installing a VM, we need to create it with the vSphere Client. On the item symbolising the ESXi node just right-click and choose "New Virtual Machine...".

Figure 17: vSphere Client - Create VM (Step 1)



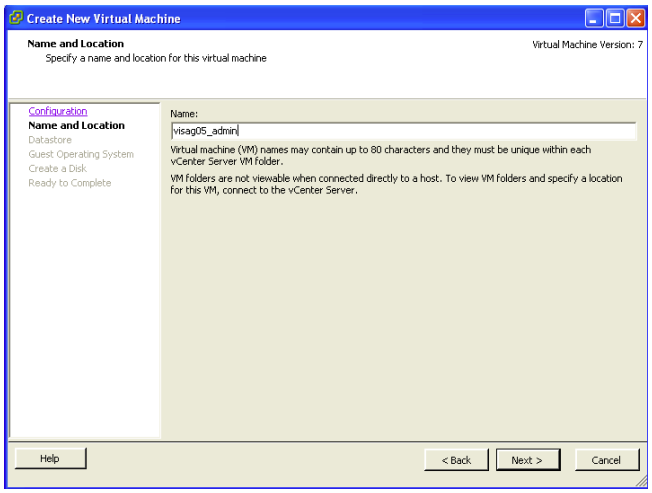
Choose "Typical" configuration and click "Next".

Figure 18: vSphere Client - Create VM (Step 2)



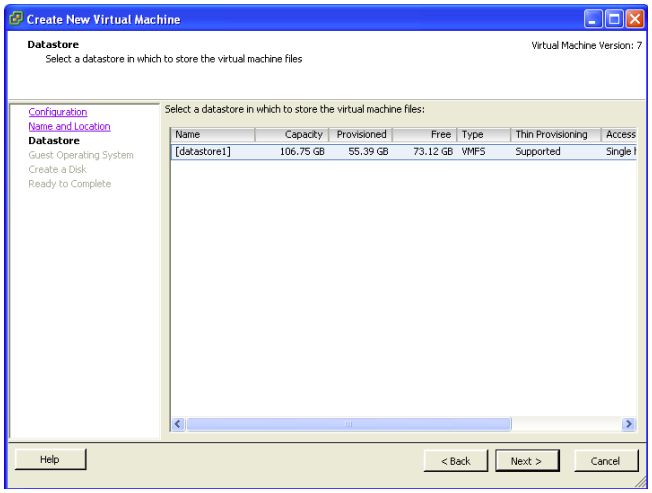
Give a name to the VM and click "Next".

Figure 19: vSphere Client - Create VM (Step 3)



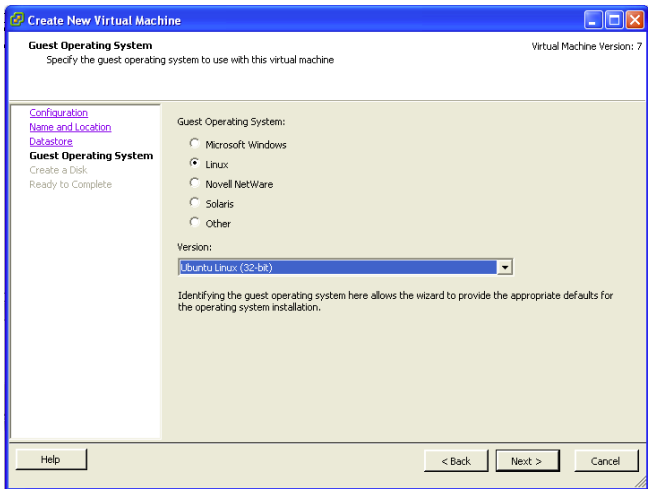
Choose the datastore of destination and click "Next".

Figure 20: vSphere Client - Create VM (Step 4)



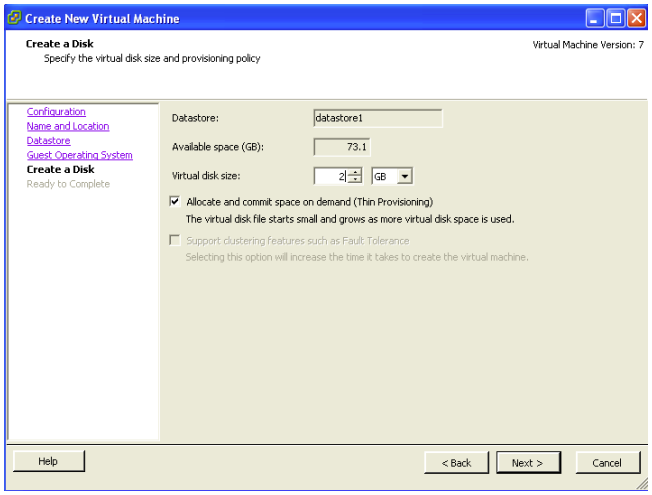
Choose the "Operating System" and click "Next".

Figure 21: vSphere Client - Create VM (Step 5)



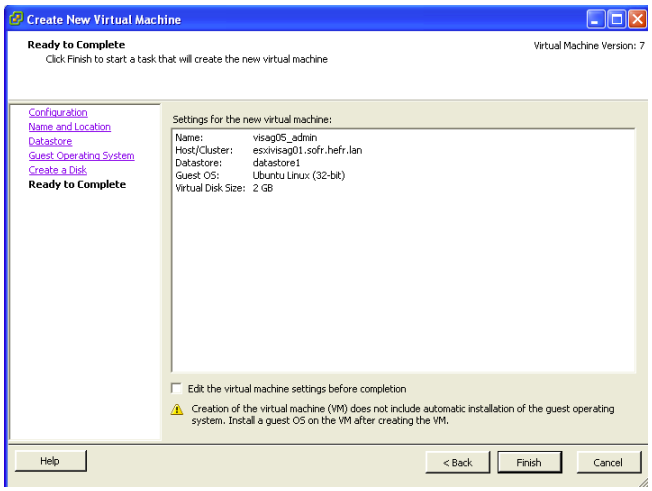
Choose the size of the disk and select "Allocate and commit ..." and click "Next".

Figure 22: vSphere Client - Create VM (Step 6)



Click "Finish", the VM is created and appears in the left-side of the vSphere client.

Figure 23: vSphere Client - Create VM (Step 7)



G Create a snapshot of a VM

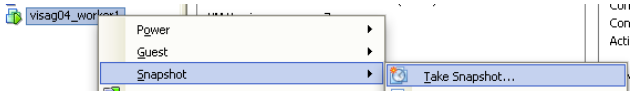
The worker VM will be reverted to a snapshot before executing a job. To be able to do this action, we need to create the snapshot of our worker VM in the good state. After installing all the needed elements, you should release the dhcp lease and shutdown the VM. On a Ubuntu OS, use the following command:

```
sudo dhclient -r && sudo shutdown -h now
```

This will remove any IP address information from the snapshot. It is very important to have to shortest revert time possible. Once we have to clean Vm, we can create the snapshot.

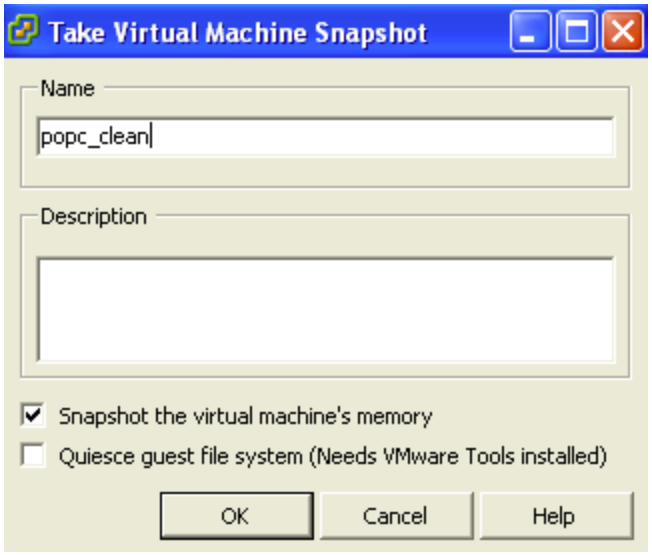
To create the snapshot, make a right-click on the VM in vSphere Client and select "Snapshot > Take a snapshot" (see Figure 24).

Figure 24: vSphere Client - Create Snapshot (Step 1)



Give a name to the snapshot (this name will be used in the virtual configuration of POP-C++), an optional description and click "Ok" (see Figure 25).

Figure 25: vSphere Client - Create Snapshot (Step 2)



The snapshot will be created (it can take a while to create the snapshot) and the worker will be ready to be used.