



A BEGINNER'S GUIDE TO

---

# Getting Started with QVD

---

Rowan PUTTERGILL

<rowan.puttergill@qindel.com>

July 4, 2011

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Components and Architecture	1
1.1.1	QVD Client	1
1.1.2	QVD Server	2
1.1.3	QVD Administration	2
1.1.4	PostgreSQL DBMS	3
1.1.5	Shared Storage	3
1.2	High-Level Architecture Diagrams	4
<b>2</b>	<b>Planning your QVD Solution</b>	<b>6</b>
2.1	General Requirements	6
2.2	System Hardware Requirements	7
<b>3</b>	<b>Installing the QVD Demo Metapackage</b>	<b>8</b>
<b>4</b>	<b>Installing and Configuring QVD DB</b>	<b>9</b>
4.1	Installation	9
4.2	Creating a user account	9
4.3	Creating the QVD database	10
4.4	Changing PostgreSQL configuration	10
4.5	Installing the QVD tables	10
4.6	Testing access	11
<b>5</b>	<b>Installing and Configuring QVD Server</b>	<b>12</b>
5.1	Installation	12
5.2	Base Configuration	12
5.3	Configuring network	13
5.3.1	Set dnsmasq to be controlled by QVD	13
5.3.2	Configure IP forwarding	13
5.3.3	Configure a Network Bridge	14
5.4	Configuring SSL	14
5.4.1	Creating a self-signed certificate	14
5.4.2	Configure QVD to use the SSL Certificates	15

<b>6</b>	<b>Installing and Configuring QVD-WAT</b>	<b>17</b>
6.1	Installation	17
6.2	Running	17
6.3	Add Your QVD Server Node	18
6.4	Installing Your First Image	19
6.4.1	Download a demonstration OSI	19
6.4.2	Load the OSI into QVD	19
6.5	Adding Your First User	21
6.6	Attaching a Virtual Machine To A User	22
<b>7</b>	<b>Installing and Configuring QVD Client</b>	<b>26</b>
7.1	Downloading and Installing the Windows Client	26
7.2	Downloading and Installing the Ubuntu Client	28
7.3	Connecting to your Virtual Desktop	28
<b>8</b>	<b>Additional Notes about Server Side Administration</b>	<b>31</b>
8.1	Creating Your Own Image	31
8.2	Making Changes To An Image	31
8.3	Authenticating Users Against LDAP	32
8.4	Managing A Virtual Machine As An Administrator	32
<b>9</b>	<b>Conclusion</b>	<b>33</b>

# List of Figures

1.1	Client and Server Interaction in the QVD Architecture	4
1.2	QVD-WAT and Server Node Interactions in the QVD Architecture	5
6.1	The Node link in the QVD-WAT Navigation bar	18
6.2	Adding a New QVD Server Node in the QVD-WAT	18
6.3	The Images link in the QVD-WAT Navigation bar	19
6.4	Adding an Image to QVD using the QVD-WAT	20
6.5	The Users link in the QVD-WAT Navigation bar	21
6.6	Adding a User in the QVD-WAT	21
6.7	After Adding a User in the QVD-WAT	22
6.8	Editing a User in the QVD-WAT to Attach a Virtual Machine	23
6.9	The New Virtual Machine link	23
6.10	Adding a Virtual Machine to a User in the QVD-WAT	24
6.11	The Virtual Machine added to a User in the QVD-WAT	24
6.12	The Virtual Machine has started and is running	25
7.1	The Windows QVD Client Installer Wizard	27
7.2	The Windows QVD Client	28
7.3	Enter the details for your QVD connection	29
7.4	A Gnome desktop loaded under QVD	30

# Chapter 1

## Introduction

In this guide we will provide an introduction to QVD and will present the basic steps that you will need to take in order to set up and configure QVD server and the various auxiliary components that you will make use of. This guide is intended to help administrators understand how to set QVD up from scratch in a demonstration environment. It will provide the most simple installation and configuration instructions that can be used to get started. It will not go into any detail about additional configuration steps or build functionality, as these details will be covered in the QVD Administration Manual.

QVD is under continuous development. While we try to keep all of our documentation up to date with the current version, it is possible that some new functionality is provided before the documentation has been updated. If there are sections in this document which are have become obsolete, or if you find that some of the instructions provided do not work as expected, please do not hesitate to contact us.

### 1.1 Components and Architecture

In this guide, we will not spend a long time discussing the different components that make up QVD in great detail, however it is important to understand the basic architecture and how each of the components interacts so that you are aware of what needs to be installed and configured in order to get your QVD environment up and running.

QVD 3.0.0 is composed of several parts:

- QVD client,
- QVD server,
- Administration tools,
- PostgreSQL DBMS,
- Shared Storage.

These components can be installed on the same system for testing and evaluation. Naturally, if you are installing all components on the same system, you will not need to make use of shared storage, however in a production environment you would usually make this available over NFS. For production use we recommend using separate machines for the database, administration tools, and QVD servers. In order to reduce the complexity of your host networking requirements, we certainly recommend that when testing and evaluating QVD, you run the client from a separate workstation. This will also help you to see QVD functioning in the way that it is supposed to work.

#### 1.1.1 QVD Client

The QVD Client is the application that the User will run in order to access his or her own desktop. QVD provides client applications in a variety of package formats and for a selection of base operating systems, including Linux *.deb* and *.rpm* packages for Ubuntu and SUSE Linux distributions and a Windows executable for Microsoft Windows environments.

The QVD Client is a modified NX client that will connect to the QVD Server node, where it will be authenticated before loading the user's Desktop environment. The client comes with a GUI that makes connecting to a Server a little more user-friendly, but the GUI is not required. In order for the client to connect, it must be configured to provide a user name and password, along with the server node's FQDN or IP address. There is also an option to control the connection type. For this, there are three possibilities:

- *Local* - For connections with high bandwidth, such as a LAN
- *ADSL* - For connections with broadband speed bandwidth, such as an ADSL connection
- *Modem* - For low bandwidth connections

Changing the connection type controls the amount of compression and caching that is performed by the client. This means that for High Bandwidth connections, less compression and caching is performed and the quality of the desktop display is much better than for lower bandwidth connections.

### 1.1.2 QVD Server

The QVD Server is also known as a Node, and represents the physical location where a QVD server daemon is running. In an enterprise environment, it is possible that you may have multiple QVD Server nodes to handle high load etc.

QVD Server is responsible for accepting requests from the QVD Client and for loading a Virtual Machine to serve the client request. In order to facilitate authentication requests, and to determine which image to load within the Virtual Machine, QVD Server makes use of a PostgreSQL database.

In actual fact, the QVD Server is comprised of three separate core components:

- **L7R** - A level-7 router, responsible for routing client connections to the correct virtual IP address that is configured for the Virtual Machine that has been created for the connecting user. Also responsible for authenticating the user prior to connection.
- **HKD** - The *House Keeping Daemon*, responsible for starting and stopping virtual machines, and updating status within the PostgreSQL database
- **Node** - Responsible for managing the L7R and HKD

In order for QVD Server to work correctly, it must have access to the PostgreSQL database, and it must have a properly configured network bridge in place. Configuring the Server can be relatively simple, but there are many options that apply to the L7R and HKD to fine tune behaviour. We will not cover these additional options in this document, but will provide the most basic configuration required in order to get QVD Server running.

QVD Server is responsible for loading the image used for the base operating system that will be served to the client, and connecting this with the user home directory. The QVD Server also makes use of *overlays* that can be used to store temporary data such as log and temporary files. In general, overlays are locally hosted and are destroyed when the Virtual Machine is shutdown. It is possible to make these persistent, by mounting the overlay store on an NFS share. But we will cover these topics a little later.

### 1.1.3 QVD Administration

QVD comes with a set of administration tools that help with the configuration of QVD Server, and which allow you to easily add users.

For ease of use, the QVD Web Administration Tool (WAT) should be installed and set up, so that you are easily able to administer the QVD infrastructure. The QVD-WAT provides a simple user interface to monitor the status of QVD, including the Virtual Machines that are running or that have stopped; the sessions that are currently open; and the number of users, virtual machines, nodes and images that are used within the solution.

The QVD-WAT also provides tools to manage users, set up Virtual Machines, and upload new images.

In order to function properly, the QVD-WAT needs access to the PostgreSQL database and should have access to the `/var/lib/qvd/storage` directory. Often, this is mounted across NFS so that images can be shared with the QVD Server Nodes that require them.

Like most of the QVD components, QVD-WAT is coded in Perl. QVD-WAT makes use of the Perl Catalyst framework to facilitate much of its code-base, and includes its own web-server, which runs on TCP port 3000 by default.

It is possible to administer your QVD Solution without installing the QVD-WAT, by using the command line interface. To do this, you will need to install the `qvd-admin` tool. QVD-admin is a perl script that is capable of performing any of the administration tasks available through the QVD-WAT from the command line. It also includes some additional functionality that may be useful to an administrator, such as the option to perform a backup of the PostgreSQL database.

You will need at least one of these two tools to configure your QVD Solution properly.

### 1.1.4 PostgreSQL DBMS

QVD stores information about users, virtual machines, nodes and images within a PostgreSQL database. This database is a central component that is used by each QVD Server Node, as well as the QVD Administration tools. It is used to determine the status of the overall solution and to manage users, sessions and all of the individual components in the solution.

The PostgreSQL database is installed using the `qvd-db` package, which includes all of the required dependencies as well as the tools needed to create and configure the database for use within the QVD solution.

### 1.1.5 Shared Storage

In a production level deployment, the QVD components are usually installed on separate systems in order to provide better running stability and to ensure that adequate system resources are available for each logically independent task. Nonetheless, the QVD Administration Tools and the QVD Server Nodes need access to shared storage components in order to manage and load system images, user profiles and user data.

QVD keeps all commonly used files in the directory location:

```
/var/lib/qvd/storage
```

Within this location there are four subdirectories:

- **homes** - location of the home directories for each user serviced by QVD. These should be accessible to all QVD Server Nodes usually on an NFS share.
- **images** - location of the OSIs (Operating System Images) that are loaded by the nodes for each Virtual Machine that is created. These need to be accessible to QVD Server Nodes and to the QVD-WAT. This directory might be stored on an NFS share, but in a very simple configuration where the QVD-WAT is either not used or is hosted on the same system as the QVD Server Node, it can be hosted locally which will help to improve performance.
- **overlays** - location used to store overlays for temporary files etc. Usually this folder can be hosted locally, but for more persistent behaviour in your virtual machines, you can choose to store these on an NFS share and configure QVD to make your virtual machines persistent.
- **staging** - temporary location for OSIs that you want available in the QVD-WAT for the purpose of loading as an image. Files located here are available within QVD-WAT when you select to add an image. When you enable an image, the image file is moved to the `images` directory and is updated to include additional information such as the available memory to allocate the system image, and the memory that should be allocated to user space. This folder can either be hosted locally or on NFS

While we will not be making use of shared storage in this guide, since all of the components will be installed on the same system and will have access to the same storage directories, you should be aware that in a normal environment you will have to configure NFS mountpoints for some of the directories listed above. If you are interested in setting your environment up in this way, please refer to the *QVD Administration Manual*, where the chapter titled *Design Considerations and Integration* contains information on configuring NFS for QVD.

## 1.2 High-Level Architecture Diagrams

In this section we will provide some simplistic diagrams explaining the architecture of a typical QVD deployment to show how the different components interact. Please note that these are very high-level diagrams and do not cover many of the internal processes involved. More detailed architecture diagrams including flow diagrams are included in the *QVD Administration Manual*.

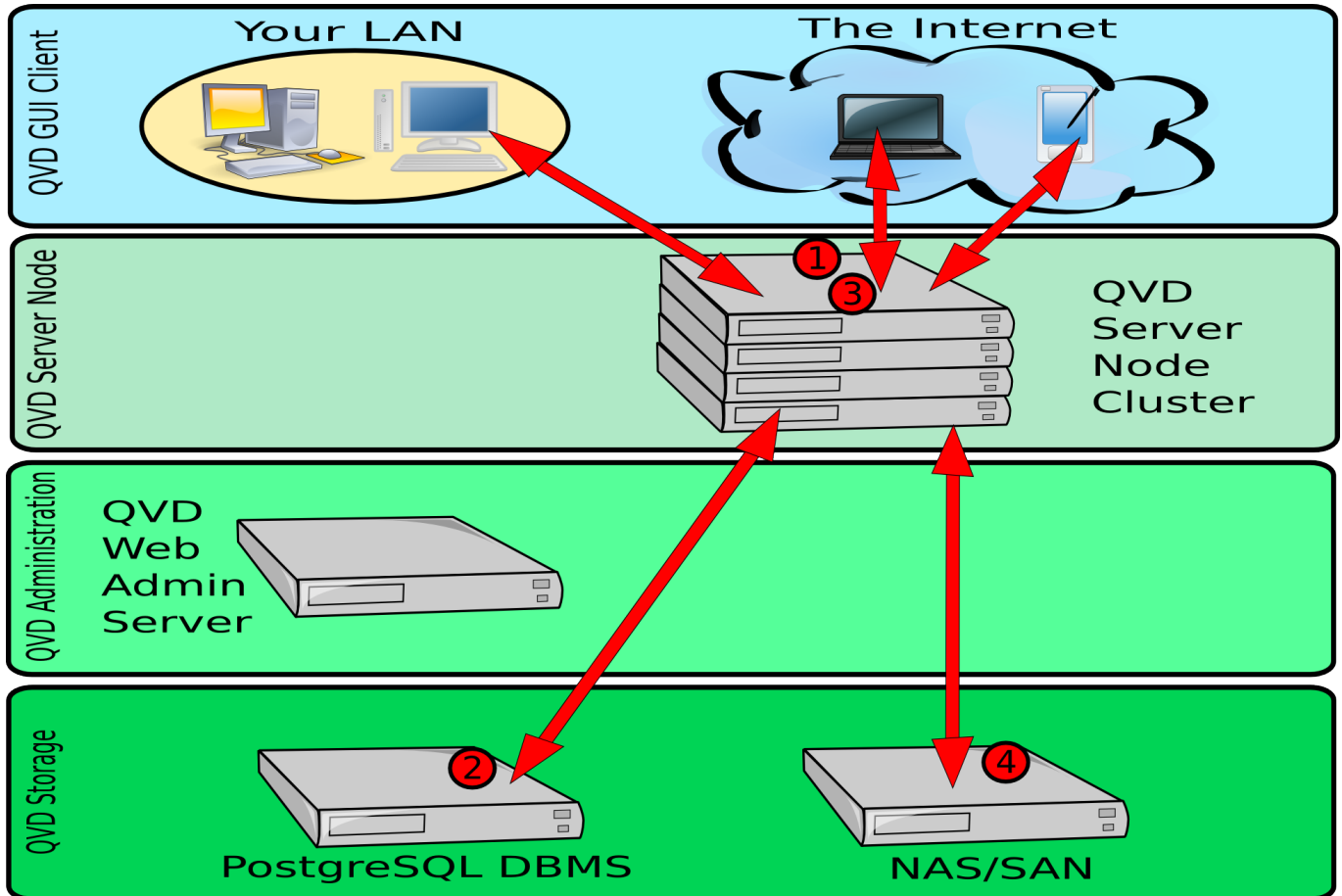


Figure 1.1: Client and Server Interaction in the QVD Architecture

In the above diagram, we show the interactions between the GUI client application, the Server Nodes, Shared Storage and the PostgreSQL database. Note that for this relationship, the QVD Web Administration Tool is not used.

1. The client application can connect over a LAN or via the Internet. The connection initially makes use of the HTTPS protocol to handle the initial authentication and to establish a session.
2. The Server Node connects to the PostgreSQL database to check configuration settings and to authenticate the user. If authentication has been delegated to some other integrated service such as LDAP or OpenSSO, the Server Node will obtain the appropriate information from the database and perform the steps required to authenticate the user. The Server also uses the database to obtain information about which virtual machine it should serve to the user along with other related information. Finally, the Server Node will regularly update status information about sessions, virtual machines and users within the database for management purposes.
3. Once authenticated, the server and client renegotiate an NX protocol connection secured using SSL. The client is then able to connect to a desktop loaded within the allocated virtual machine running on the Server Node.
4. Prior to any connection from the client, the Server Node will load an image from the Shared Storage into a Virtual Machine. The Shared Storage is usually accessed over an NFS connection. When the Virtual Machine is started for a particular user,



the home directory for that user is also mounted over NFS. By keeping the user's home directory within Shared Storage, the user may be able to access different Server Nodes and still access the same profile and data that the user requires. This also means that in the case of Server Node failure, it is easy to quickly reprovision a user on a different Server Node.

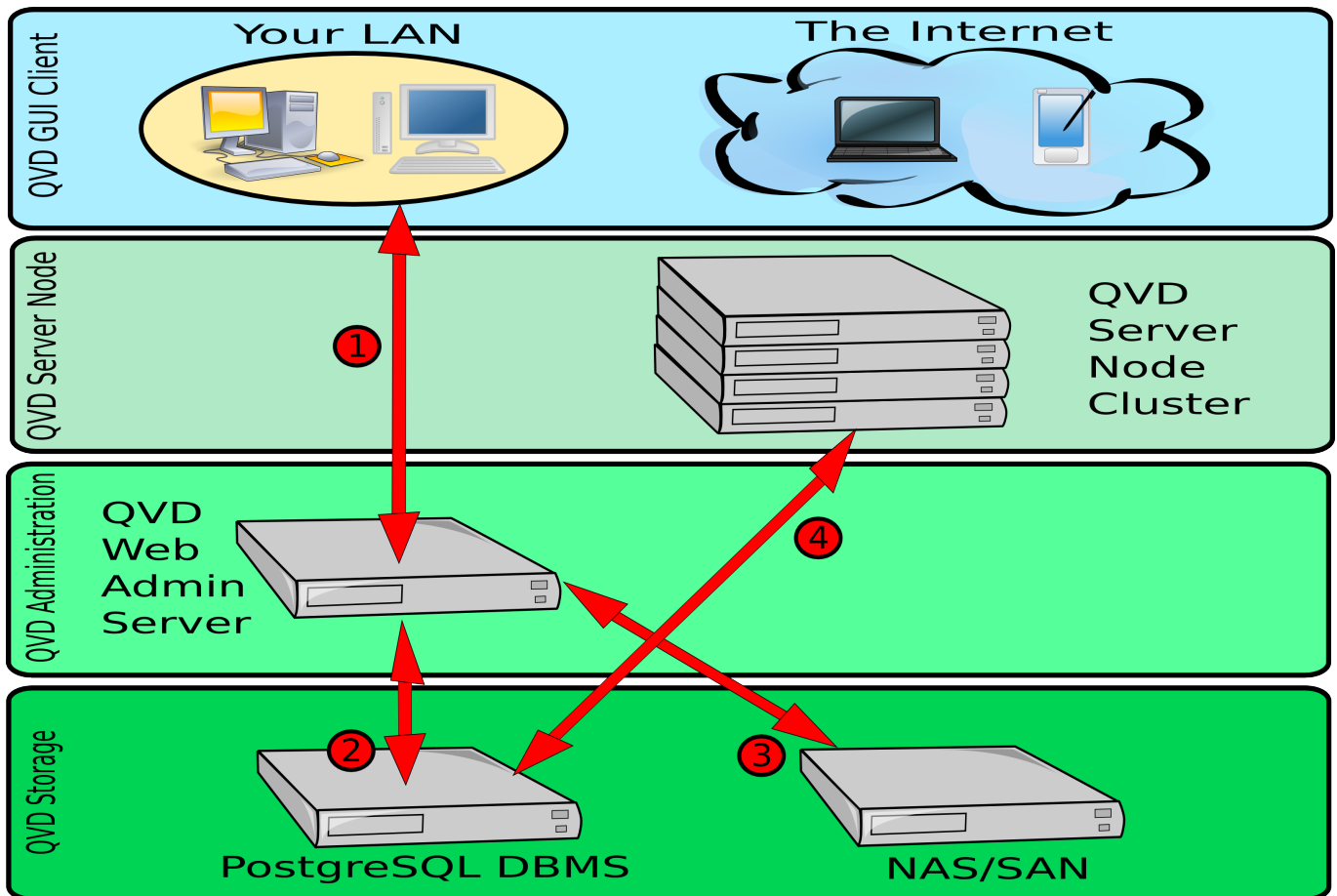


Figure 1.2: QVD-WAT and Server Node Interactions in the QVD Architecture

In the diagram above, we explore the different interactions that are involved in the functioning of the QVD Web Administration Tool (QVD-WAT). The QVD-WAT interacts with all of the components within the solution except the QVD GUI Client. The interactions mentioned here are simplified, since there are a large number of operations that can be performed using QVD-WAT.

1. An Administrator can connect to the QVD-WAT using a normal Web Browser. The connection takes place over HTTP, but can be secured to use HTTPS. The Administrator will need to authenticate in order to make use of the tools offered by QVD-WAT. Authentication uses HTTP-BASIC but credentials are stored within the PostgreSQL database.
2. The QVD-WAT makes use of the PostgreSQL database to store configuration information entered by the Administrator via the web interface. The QVD-WAT also extracts information such as the status of Virtual Machines, Users and Sessions from the database, to present to the Administrator via the web interface.
3. When an OSI (Operating System Image) is available within the Shared Storage, the QVD-WAT can enable it so that it is accessible to any QVD Server Node and can be loaded into a Virtual Machine. In order to do this, the QVD-WAT needs access to the Shared Storage via an NFS connection.
4. The House Keeping Daemon (HKD) on each QVD Server Node regularly polls the PostgreSQL database to pick up configuration and status changes made by the QVD-WAT. For instance, when QVD-WAT starts or stops a Virtual Machine, this change is effected within the database, and when the HKD next polls the database and determines that the status of a Virtual Machine has been changed, the change will be enacted on the QVD Server Node. As a result, the QVD-WAT never interacts directly with any particular Server Node, but always uses the PostgreSQL database as an intermediary.

## Chapter 2

# Planning your QVD Solution

In this guide, we are assuming that you want to set up your first QVD demonstration environment. For this reason, we will assume that the server-side components within the solution will be hosted on the same physical server. In order to keep things as simple as possible, we will also assume that you will test the solution using the QVD Client installed on a separate workstation. Although it is possible to have all of the components including the client running on the same machine, it will require some additional networking steps that we would prefer to avoid in this document.

Since all of the components will be running on the same system, we will not be too concerned about shared storage. However, it is important to understand that QVD makes use of some common storage between different components in the solution, and that to maximize the potential of your solution it is likely that some of these storage directories should be located on an NFS share on a NAS device or a SAN.

With all of this in mind, we will continue to build this solution on a single host to keep things as simple as possible. In reality, it is more than likely that a production environment would keep each of the different components on different systems and the storage would be managed across the different systems that are interacting within the solution. If you feel comfortable configuring your NFS shares, and building and installing each component on a different machine, please feel free to do so.

QVD currently has packages for the server-side components available for the SUSE Linux and Ubuntu Linux distributions. This guide will assume that you are installing the packages on Ubuntu 10.04 (Lucid Lynx). The instructions will mostly be similar for SUSE Linux, although the download repository and installation commands will be different. If you need instructions for SUSE Linux, please contact us and we will endeavour to provide them.

## 2.1 General Requirements

The QVD components require the Ubuntu 10.04 (Lucid Lynx) GNU/Linux operating system. Add the following line to `/etc/apt/sources.list` on any systems where you intend to install QVD components

```
apt-add-repository 'deb http://theqvd.com/debian lucid main'
```

Execute `apt-get update` after modifying `sources.list`.

The QVD repository provides the following packages:

- **qvd-demo-single-instance-nosupport**: meta-package to install a demonstration configuration on a single host
- **qvd-client**: client software
- **qvd-l7r**: network server
- **qvd-hkd**: backend daemon
- **qvd-admin**: command line tools for managing users, virtual machines and operating system images
- **qvd-vma**: agent that runs in virtual machines

- **qvd-wat**: web administration tool
- **qvd-db**: central database for the platform

Each of these packages will have a number of dependencies that can be satisfied by other packages provided from the usual Ubuntu repositories. A summary of other Open Source components required by QVD follows:

- The PostgreSQL RDBMS.
- KVM: Hypervisor.
- NX: protocol that handles remote desktop connections.
- Catalyst: an MVC web-application framework for Perl
- Ebttables: a IP-based firewalling utility for ethernet bridges

**Important**

At the moment QVD works only on PostgreSQL 8, and libcatalyst-perl version 5.80024-1. These are the default versions currently provided with Ubuntu 10.04 (Lucid Lynx).

---

## 2.2 System Hardware Requirements

The QVD server side components should usually be run on independent systems to ensure that they have adequate resources to run, and hardware requirements will vary depending on the number of users that you wish to service, the number of different Operating System images that you intend to make use of and various other factors.

For the sake of this guide, which assumes that you are evaluating QVD and will only install a single image and set up a one or two users at most, we recommend the following system hardware requirements as a guideline:

**System Processor:** 64-bit processor preferably with more than one core and supporting virtualization extensions (Intel or AMD). You can support around 8 users per core.

**System Memory:** At least 2GB RAM. This should be sufficient for up to 4 users.

**Disk Space:** At least 4GB disk space should be available to contain your operating system image etc. More than likely, you should try to double this to work comfortably with the tools involved in importing an image.

**Network Interface:** You will need at least one network interface available. A 10/100 Ethernet NIC should be perfectly sufficient. We have succeeded in serving desktops over Wireless connections as well.

You may use any client system to run the QVD Client software, as long as it supports Linux or a modern Microsoft Windows operating system.

## Chapter 3

# Installing the QVD Demo Metapackage

The QVD provides a QVD Demo Metapackage. Installing this package on a single host will automate many of the steps described in this document. This package is designed to set up and configure a demonstration QVD environment that runs off a single host, and will help you to quickly build a QVD solution that can be used to demonstrate QVD's capabilities without spending time on the initial configuration steps.

You can install this package by running the following command:

```
# sudo apt-get install qvd-demo-single-instance-nosupport
```

---

### Important



If you choose to install a demonstration environment using this metapackage, you can skip through most of this document but you will still need to load an OSI image, set up your first user and create a virtual machine. There are a few things to note if you take this path. Firstly, the metapackage is unsupported, in that it is only designed for demonstration purposes and assumes that your environment is ready to support QVD. For this reason, we highly recommend that you still read through this document so that you understand all of the steps that will be performed by the metapackage, in case something goes wrong. Secondly, if the metapackage installs successfully and the environment is set up correctly, you should be able to skip to the section titled: **Installing and Configuring QVD-WAT** and pick up the instructions from the subsection titled [Running](#).

---

## Chapter 4

# Installing and Configuring QVD DB

Since one of the most fundamental components in the QVD solution is the QVD database, we suggest that you install this first. The database will be used to tie all of the other components together and to enable them to interact.

### 4.1 Installation

The preferred way to install the central database is with the package `qvd-db`. It installs the PostgreSQL database system if needed.

**Note**

At the moment QVD works only on PostgreSQL 8.

To install run as root:

```
root@myserver:~# apt-get install qvd-db
```

After installing `qvd-db` you have to perform various manual steps. They are

1. creating a user account,
2. creating a database,
3. changing the database configuration, and
4. deploying the QVD database schema.

The first two steps need to be performed using the database administrator account, `postgres`. Use the following `sudo` command to change from your normal user to the `postgres` account.

```
$ sudo su - postgres
```

### 4.2 Creating a user account

If you wish to use an existing user account you can skip this step.

Once you have access to the database, you can create user accounts with the `createuser` command. It will prompt for a password for the new user and ask some details on the user account. You can answer `n` to all.

For example, to create a user called `QVDUser` you would use the following command.

```
postgres@myserver:~$ createuser -P QVDUser
Enter password for new role: passw0rd
Enter it again: passw0rd
Shall the new role be a superuser? (y/n) n
Shall the new role be allowed to create databases? (y/n) n
Shall the new role be allowed to create more new roles? (y/n) n
```

The new user can now be assigned as the owner of a database. First we need to create the QVD database.

### 4.3 Creating the QVD database

Use the `createdb` command to create a database for QVD. Use the `-O` switch to set the database's owner to the account you wish to use. In this case we will set the owner to the new user that we created in the previous step.

```
postgres@myserver:~$ createdb -O QVDUser QVDDatabase
```

### 4.4 Changing PostgreSQL configuration.

QVD uses transactions extensively, and requires a higher level of transaction isolation than is configured by default. You also have to enable network access for the user you have just created. To do this you must edit the PostgreSQL configuration files `postgresql.conf` and `pg_hba.conf`. On Ubuntu they are located in `/etc/postgresql/8.4/main`.

The transaction isolation level is controlled with the `default_transaction_isolation` setting. To enable network access to PostgreSQL in general, change the `listen_addresses` setting from `localhost` to `*`.

```
root@myserver:~# cd /etc/postgresql/8.4/main
root@myserver:/etc/postgresql/8.4/main# vi postgresql.conf
listen_addresses = '*'
default_transaction_isolation = 'serializable'
```

To enable network access for the user `qvd`, add the following line to `pg_hba.conf` (its format follows: `host database user CIDR-address auth-method [auth-options]`).

```
root@myserver:/etc/postgresql/8.4/main# vi pg_hba.conf
host>>>QVDDatabase>>>QVDUser>>>.....192.168.0.0/24>>>.....md5
```



#### Note

Make sure to replace the default network `192.168.0.0/24` with the network that your QVD platform uses.

Restart PostgreSQL for the changes to take effect.

```
root@myserver:~# /etc/init.d/postgresql-8.4 restart
```

### 4.5 Installing the QVD tables

It is now time to populate the database with the tables that will be used to hold data for QVD. Before we can use any of the QVD tools, we will need to configure the database, user name and password in the QVD configuration files.

Once done, execute `qvd-deploy-db.pl`. It creates the table structure that QVD needs.

```
# qvd-deploy-db.pl
```

## 4.6 Testing access

The following command lists the tables used by QVD.

```
anyuser@otherserver:~$ psql -U QVDUser -W -h myserver
Password for user qvd:
psql (8.4.1, server 8.3.10)

qvd=> \d
```

## Chapter 5

# Installing and Configuring QVD Server

### 5.1 Installation

It is now time to install the QVD Server. To do this, ensure that you have root privileges.

```
root@myserver:~# apt-get install qvd-node
```

This will install all of the QVD server utilities, as well as all of the dependencies required to run a QVD Node.

It is also worthwhile installing the QVD CLI Administration utility at this point to help with some of the steps that you will perform during this phase of the setup process. The QVD CLI Administration utility is included as a dependency for the **qvd-db** package, so more than likely you will already have it installed. Nonetheless, it is possible that you have decided to install the different components on different hosts, so for the sake of completeness, you can install this utility using the following command:

```
root@myserver:~# apt-get install qvd-admin
```

This useful tool permits scripting all the operations that can be performed using the web administration tool in the **qvd-wat** package. You can install it on any host that you want to use to manage your QVD Server. For instance, you may want to integrate QVD with an external monitoring tool such as Nagios, so installing the QVD CLI Administration utility on this host would make this possible.

The QVD Administration utility requires access to the QVD Node configuration file in order to function properly. We will set this up in the next step, but it is worth keeping in mind that if you want to install this utility on any other host, you should copy the QVD Node configuration file to the host where you intend to run the tool.

### 5.2 Base Configuration

Once you have finished installing the packages, you will need to create a node configuration file. The easiest way to do this is to copy the template configuration to your `/etc` directory:

```
root@myserver:~# cp -R /usr/share/qvd/config /etc/qvd
```

Now you will need to edit the file `/etc/qvd/node.conf` to include the details needed to access the database, and to prepare QVD server to set up networking for each virtual machine. If you have been following these instructions closely, your configuration file should look like this:

```
#
# QVD Node Configuration
#
# Name of this node in QVD. Usually the machine's hostname.
nodename = mycomputer
```



```
# Database connection information.
# database.host: where the QVD database is found
# database.name: the name of the QVD database
# database.user: the user account needed to connect
# database.password: the password needed to connect
database.host = mycomputer
database.name = QVDDatabase
database.user = QVDUser
database.password = passw0rd

# QVD-WAT authentication settings.
# The username and password required for logging in to qvd-wat
wat.admin.login = admin
wat.admin.password = admin

# Which system user account is used to run QVD daemons
l7r.as_user = root
hkd.as_user = root

# Log level. One of ALL, DEBUG, INFO, WARN, ERROR, FATAL, OFF
log.level = ERROR

#network info
vm.network.bridge = qvdnet0
vm.network.dhcp-range = 10.3.15.2,10.3.15.254
vm.network.netmask=255.255.255.0
vm.network.gateway=10.3.15.1
```

There are a couple of entries here that may need some explanation.

First, the `nodename` entry and the `database.host` entry should match your machine's hostname, so the example above does need some editing.

In the configuration above, we have added some network info to the default provided configuration file. We will be creating the network bridge shortly, but for now you should just enter the information provided. It should be fairly apparent that the QVD Server Node will set up its own DHCP service to allocate IP addresses to the virtual machines within the range 10.3.15.2 to 10.3.15.254 and that these virtual hosts will be accessible via a network bridge.

## 5.3 Configuring network

Getting your network configuration right is one of the most important steps to getting QVD running properly. In order to be successful you need to prepare your system to run the QVD Server. If you follow the steps in this section carefully, you should not run into any trouble.

### 5.3.1 Set dnsmasq to be controlled by QVD

QVD uses dnsmasq as a DHCP and DNS server for the virtual machines that run in a node. In order to function correctly, dnsmasq needs to be run by the `qvd-node` process. By default, the Ubuntu package starts the process running as a daemon in the background, so you need to stop it from starting automatically. This is done with the following commands.

```
# /etc/init.d/dnsmasq stop
# sed -i s/ENABLED=1/ENABLED=0/ /etc/default/dnsmasq
```

### 5.3.2 Configure IP forwarding

Next up, we need to make sure that your system is capable of handling the IP forwarding in order to route clients to the correct location. You can do this quickly by running the following command.

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

Unfortunately, when you reboot your host system, this change will be lost. To make it permanent, you can edit `/etc/sysctl.conf` and uncomment the line:

```
net.ipv4.ip_forward=1
```

You can force `sysctl` to reload its settings after you have edited this file by running:

```
# sysctl -p
```

### 5.3.3 Configure a Network Bridge

There are a number of ways to go about configuring your network bridge and the appropriate routing to make sure that a QVD client is routed to the correct virtual machine.

The easiest method is to set up a static network interface and to configure a set of **iptables** routing rules to perform the NAT required to translate IP addresses between your real and virtual network interfaces. To do this, edit the file `/etc/network/interfaces` and add the following lines.

```
auto qvdnet0
iface qvdnet0 inet static
    pre-up brctl addbr qvdnet0
    pre-up iptables -t nat -A POSTROUTING -o qvdnet0 -j SNAT to-source 192.168.0.2
    pre-up iptables -t nat -A PREROUTING -d 192.168.0.2 -p tcp --dport 8443 -j DNAT --to- ←
        destination 10.3.15.1
    address 10.3.15.1
    netmask 255.255.255.0
```

It is important to note that in the above example you will need to change the IP address **192.168.0.2** to the IP address of the network interface that you intend your clients to connect to.

While there are other cleaner approaches to setting up your network, these sometimes run into problems with particular network interfaces such as WIFI. The approach listed above should work for most systems.

Once you have written the network configuration to file, you should bring up the network bridge interface.

```
# ifup qvdnet0
```

## 5.4 Configuring SSL

The QVD server needs an x509 certificate and private key for securing network connections. For a production installation you should use a certificate issued by a recognized certificate authority, such as Verisign or Thawte. For testing purposes you can use a self-signed certificate. In this demonstration, we will step through creating a self-signed certificate, and use this within our configuration. If you already have a certificate signed by a third party, you can skip this step and use your signed certificate instead.

### 5.4.1 Creating a self-signed certificate

The `openssl` tool is required for creating a self-signed certificate. If you have not already installed it you can do so using the Ubuntu repositories:

```
# apt-get install openssl
```

We recommend that for working with your certificates, you create a subdirectory in `/etc/qvd`.

```
# mkdir /etc/qvd/certs
# cd /etc/qvd/certs
```

In order to create your certificate, you must first generate a private key.

```
# openssl genrsa 1024 > server-private-key.pem
```

Given the private key, a self-signed certificate is created with the following command.

```
# openssl req -new -x509 -nodes -sha1 -days 60 -key server-private-key.pem > server-
certificate.pem
```

OpenSSL will prompt you to enter the various fields that it requires for the certificate. You should enter relevant information into these fields. The most important field is the **Common Name** field which should match the fully qualified domain name of the host that will be running your QVD node.

```
You are about to be asked to enter information that will be
incorporated into your certificate request.
What you are about to enter is what is called a Distinguished Name or
a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
++++-----+
Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:Madrid
Locality Name (eg, city) []:Madrid
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Qindel Group
Organizational Unit Name (eg, section) []:QVD Team
Common Name (eg, YOUR name) []:qvd.qindel.com
Email Address []:qvd@qindel.com
```

You will now have a self signed certificate and its corresponding private key.

### 5.4.2 Configure QVD to use the SSL Certificates

In the previous step we created a directory `/etc/qvd/certs` to store our self-signed certificate. If you are using a certificate signed by a recognized CA, you may want to put your certificates into the same place in order for the following instructions to make sense.

In this step, we will configure QVD to make use of the server certificate and private key. To do this, we will use the `qvd-admin` tool.

```
# qvd-admin.pl config ssl key=/etc/qvd/certs/server-private-key.pem cert=/etc/qvd/certs/
server-certificate.pem
```

If the certificate isn't signed by a trusted authority, it has to be added to the system's trusted certificates directory so that the SSL layer can validate it. To work out what that directory is, run the following command:

```
# openssl version -d
```

The trusted certificates directory is always a subdirectory named **certs** within the directory returned by the above command.

For example, the command may return the following response:

```
OPENSSLDIR: "/usr/lib/ssl"
```

This would indicate that the your trusted certificates are stored in `/usr/lib/ssl/certs`. In most cases this is actually a symlink to somewhere else, but this path should be sufficient to work with.

In order for SSL to recognize the certificate, it needs to be named correctly. The following commands will help you to ensure that the certificate is named correctly.

```
# trusted_ssl_path=/usr/lib/ssl/certs
# cert_path=/etc/qvd/certs/server-certificate.pem
# cert_name=`openssl x509 -noout -hash -in $cert_path`.0
# cp $cert_path $trusted_ssl_path/QVD-L7R-cert.pem
# ln -s $trusted_ssl_path/QVD-L7R-cert.pem $trusted_ssl_path/$cert_name
```

You will need to correct the **trusted\_ssl\_path** and the **cert\_path** in the commands listed above to match your environment. The commands listed above will first ensure that we get the correct name for your certificate, and then will copy the certificate to the path where your trusted certificates should be stored, renaming it to *QVD-L7R-cert.pem* so that it has a name that you will be able to make sense of later. Finally, we create a symlink from the certificate to the name that OpenSSL expects in order to use the certificate file.

## Chapter 6

# Installing and Configuring QVD-WAT

The QVD Web Administration Tool (QVD-WAT) is a simple interface that makes it easier to administer and manage your QVD Server Nodes and to monitor active client sessions within your infrastructure. It also gives you the ability to manage QVD Server nodes from remote locations.

Although not strictly required to run QVD, it will certainly help you to get started with the product, so we will install it and configure our server node using this facility.

### 6.1 Installation

Install the package with apt-get:

```
# apt-get install qvd-wat
```

QVD-WAT makes use of the database connection details within the QVD Node configuration file at `/etc/qvd/node.conf`. Since we are installing all of the components on a single host, the existing file should not need to be edited. However, if you have opted to install each component on a different host, you could copy the configuration file from your QVD Server Node host to the host that you are using to run QVD-WAT.



#### Caution

At the moment QVD works only with libcatalyst-perl version 5.80024-1. These are the default versions currently provided with Ubuntu 10.04 (Lucid Lynx). If you are using a newer distribution, you may need to downgrade this package.

### 6.2 Running

The Web Administration Tool is started with the following command.

```
# /etc/init.d/qvd-wat start
```

Now you can test the connection in your browser, visiting <http://localhost:3000>

To login, you can use the default username and password:

- **username:** admin
- **password:** admin

After that, you should change the default password using:

```
# qvd-admin.pl config set wat.admin.password=newpassword
```

The QVD-WAT user is independent of QVD user management. This user is unique to QVD-WAT within the solution, so you can choose any password you prefer.

You will need to restart QVD-WAT to apply the change.

```
# /etc/init.d/qvd-wat restart
```

## 6.3 Add Your QVD Server Node

Now that you are able to connect to QVD-WAT, you will need to register the QVD Node Server that you have configured as a node within the QVD infrastructure. To do this, you can click on the **Nodes** link at the top of the page, or go to <http://localhost:3000/hosts/>.

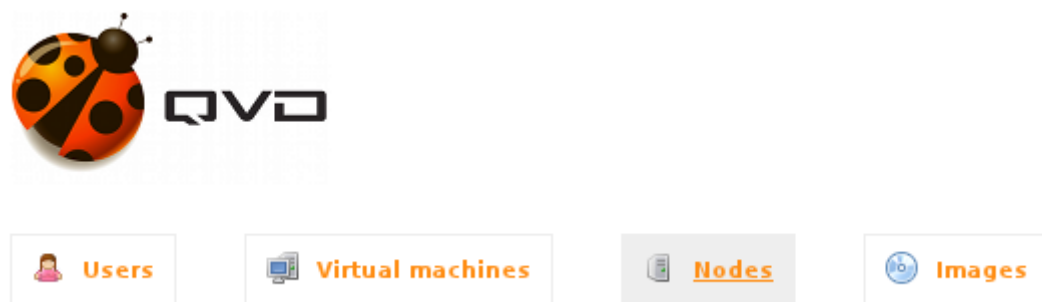


Figure 6.1: The Node link in the QVD-WAT Navigation bar

Click on the **New** button and you will be presented with a screen that looks like this:

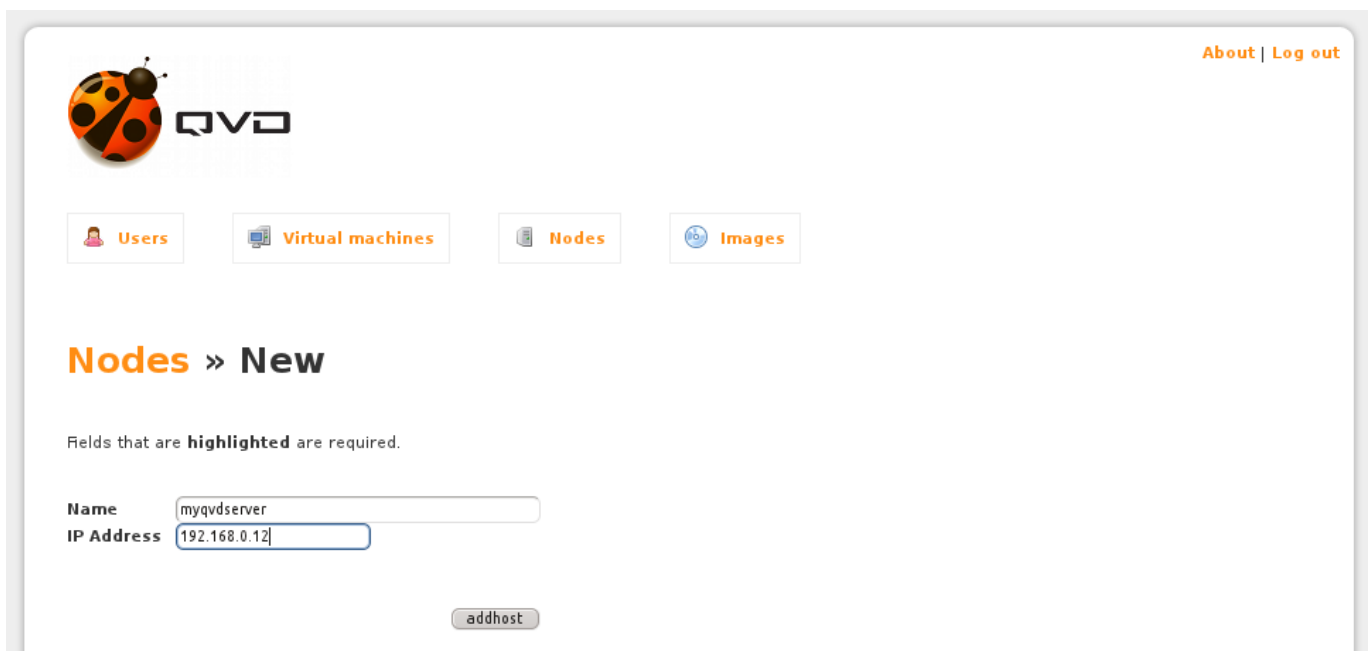


Figure 6.2: Adding a New QVD Server Node in the QVD-WAT

Enter the **Name** of the host node that you are adding, usually the hostname is a good option.

Also enter the **IP address** of the host that is running your QVD Server. This should be the externally facing IP address that you expect client applications to connect to.

Finally, click on the *addhost* button to save the information that you have just entered.

## 6.4 Installing Your First Image

In order for QVD to serve a desktop to a user, it first needs to load an OSI (Operating System Image). The OSI allows you to provide different users with different operating systems, or simply to provide different applications or environments to users with different requirements. As already mentioned, the image is loaded into its own virtual machine for each user that connects to the server. This means that you can have one image that is used to serve multiple users. On the other hand, you can install as many images as you require, so that in a *worst case* scenario you have a different image for each user.

### 6.4.1 Download a demonstration OSI

Since we just want a quick method to get started, we are not going to go to the trouble of creating our own image. Instead, we will simply download an image from the QVD website. You can download the image from the following link:

LINK TO BE PROVIDED: <http://theqvd.com/downloads>

Once you have finished downloading the image file, you will need to move it to a location that can be used by the QVD-WAT. Toward the beginning of this document, we mentioned that QVD makes use of a number of common storage directories. In particular, the directory located at `/var/lib/qvd/storage/staging` is used by the QVD-WAT as a temporary directory to store image files that will eventually be used by QVD Server Nodes to load into a virtual machine. Therefore, you should move the image file that you have downloaded into this directory:

```
# mv qvd-demo.img /var/lib/qvd/storage/staging
```

### 6.4.2 Load the OSI into QVD

Now you need to load the image that you have downloaded into QVD and set the amount of memory that you want the virtual machine to make available to the image.

In QVD-WAT, click on the **Images** link in the navigation bar, or go to <http://localhost:3000/osi/>.



Figure 6.3: The Images link in the QVD-WAT Navigation bar

Now click on the **New** button to add an image. You will be presented with the following screen.

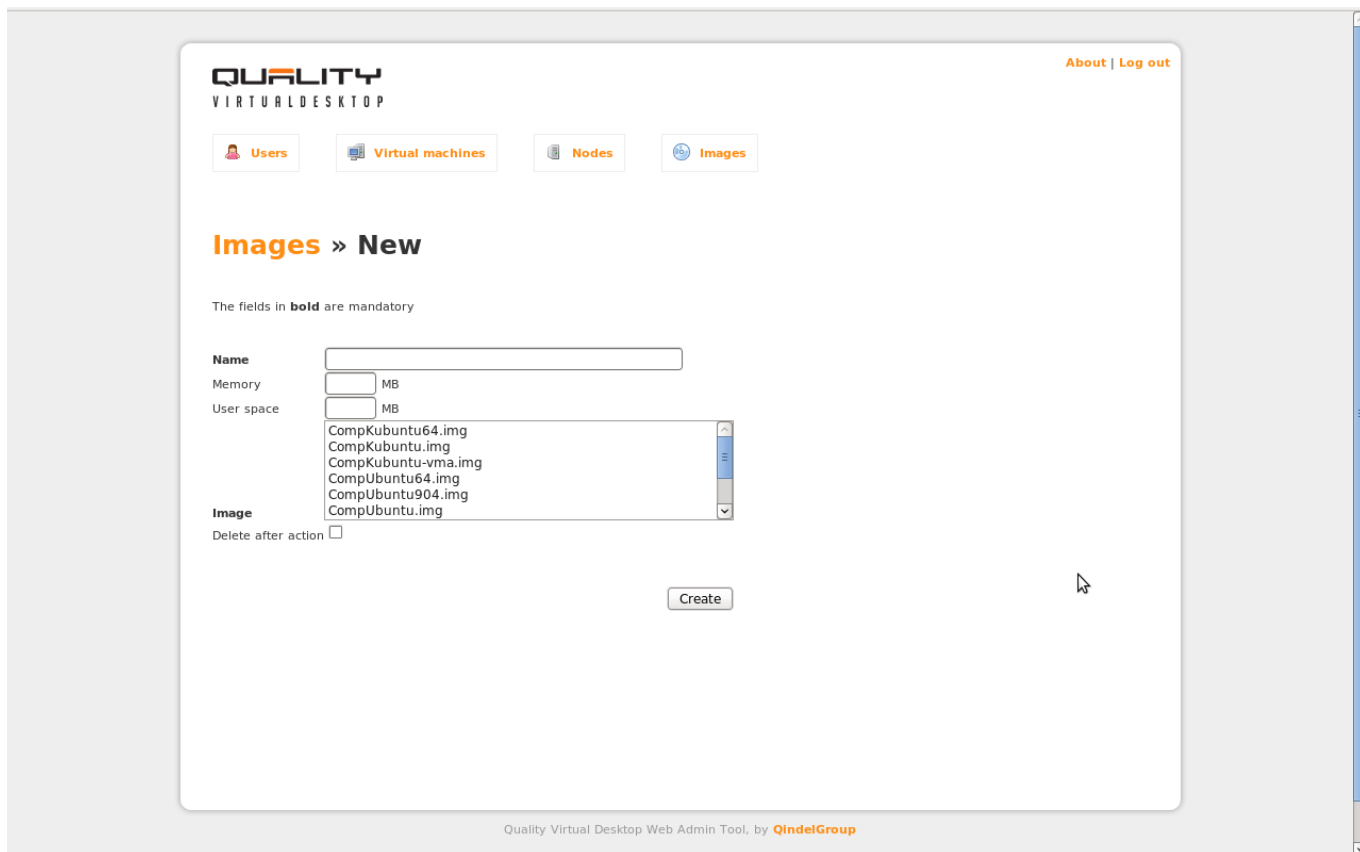


Figure 6.4: Adding an Image to QVD using the QVD-WAT

In the **Name** field, provide a name for the image. Usually you would name an image providing some indication of its purpose. So something like *AccountingUsers* might make sense if the image is going to be used to provision applications used only by Accounting staff members.

The **Memory** field is used to allocate system memory to the Operating System. It has a default setting of 256 MB. This field is not compulsory unless you want to change the default value to suit the image better. For evaluation purposes, the default value should be more than sufficient, so you can leave this field blank. However, on production systems, you may want to increase this to at least 512 MB for the Gnome or KDE desktop environment to run comfortably.

The **User space** field is used to allocate disk space to a user for the purpose of storing a home directory. By default, this option is usually not set. This means that a user's home directory will simply be stored in the same filesystem as the disk that is running the virtual machine (or on the NFS mount that you might have created for this purpose). Setting a value here will create a virtual disk of the size specified. This helps to enforce quotas and to prevent user home directories from being accessible to each other across Virtual Machines. Once again, for evaluation purposes, the default value should be acceptable and you can leave the field blank.

If you have moved your image file into `/var/lib/qvd/storage/staging` you will see it listed in the **Images** selection box. Click on it to select it as the image that you want to use.

Finally, there is a checkbox that allows you to **Delete after action**. This option is available because the original image file is copied to `/var/lib/qvd/storage/images` once you have loaded it into QVD. You may want to delete the image file from the staging directory to save disk space, but you may equally want to reuse it with alternative memory and user space settings for another group of users. It is optional to delete the temporary image file, but during evaluation we recommend that you keep it unless disk space is at a premium.

Finally, click on the **Create** button to load the image.



## 6.5 Adding Your First User

If you click on the **Users** link in the navigation bar, or go to <http://localhost:3000/users/>, you are able to add new users to QVD and manage any existing users. At this point, you will have no users listed. It is time to add a new user.



Figure 6.5: The Users link in the QVD-WAT Navigation bar

Click on the **New** button to add a user. You will be presented with the following screen:

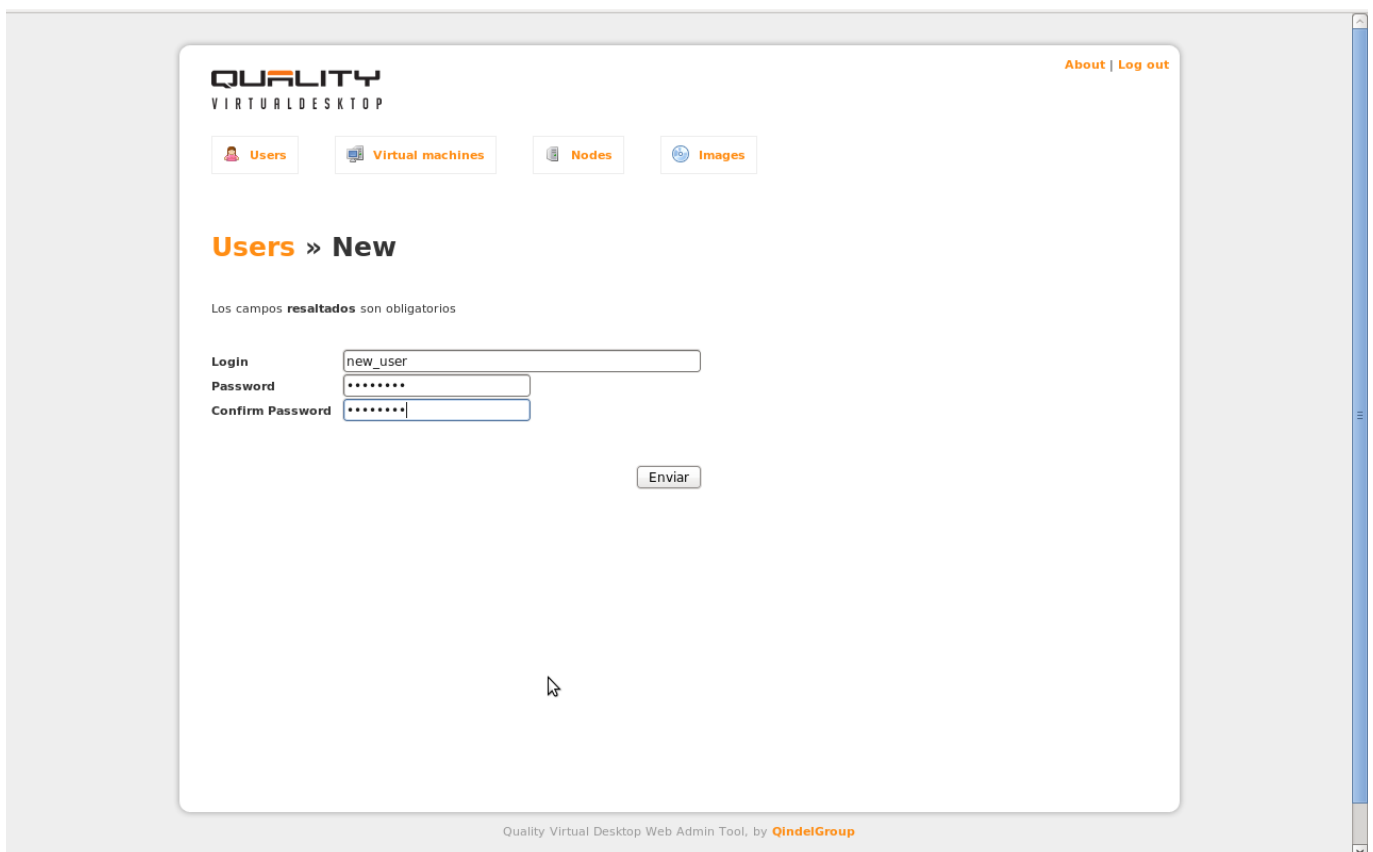


Figure 6.6: Adding a User in the QVD-WAT

Enter a **Login** name for the user. This should be a standard Linux username, usually in lowercase and with no spaces or reserved characters. For the purpose of this guide, we suggest you enter a username like *qvddemo*.

Enter a **Password** for the user. This should be a normal Linux password, although there are no limitations on length. In the spirit of reasonably good security, we recommend that the password is at least 6 characters long and contains uppercase, lowercase and numerical characters. Something like *54ghFe9* would be a sensible choice.

You will need to re-enter the password in the **Confirm Password** field.

Click on the **Submit** button to create your user.

## 6.6 Attaching a Virtual Machine To A User

Once you have added a user, you will return to the Users screen in the QVD-WAT and you will see your user listed here. The user will have been allocated an ID. If it is your first user, this will more than likely be *1*. You will notice that this ID is also a link, and a magnifying glass icon is displayed alongside it.

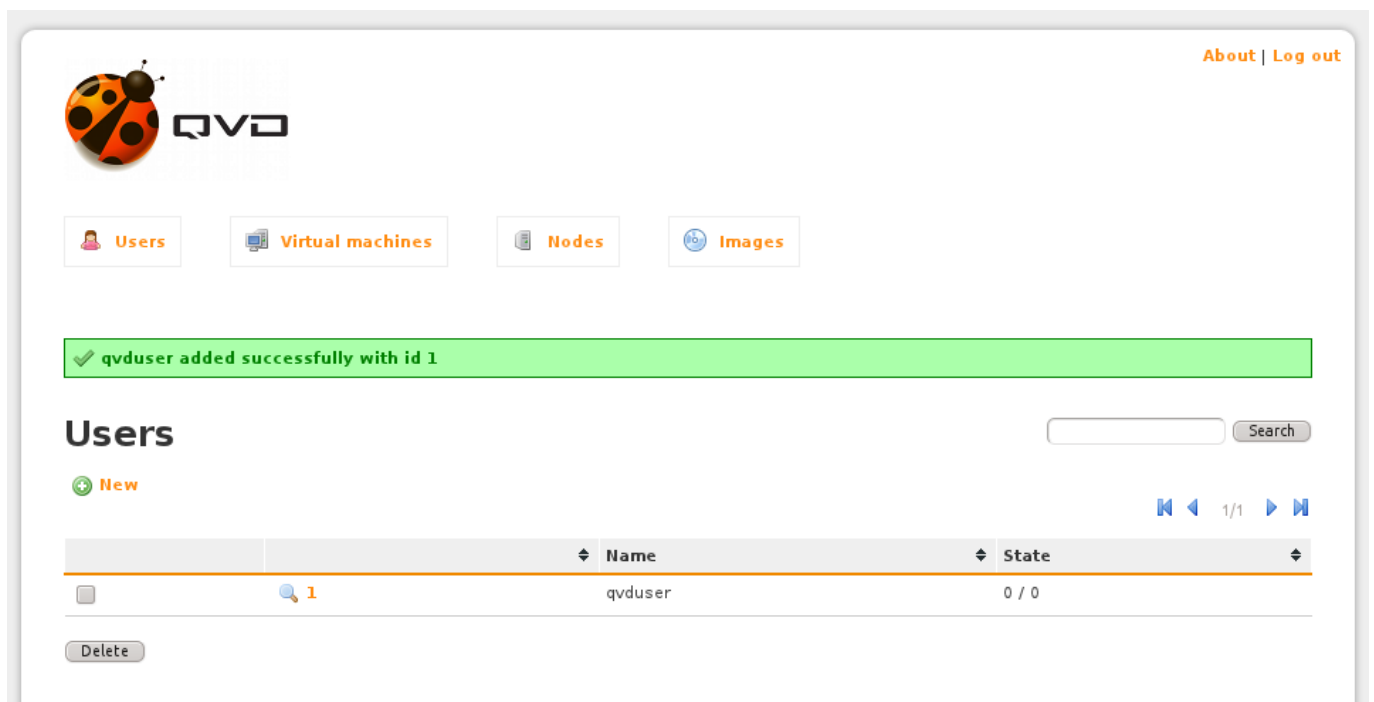


Figure 6.7: After Adding a User in the QVD-WAT

If you click on the ID link or magnifying glass icon, you will be taken to a new screen where you are able to assign a Virtual Machine to a user, or change the User's password.

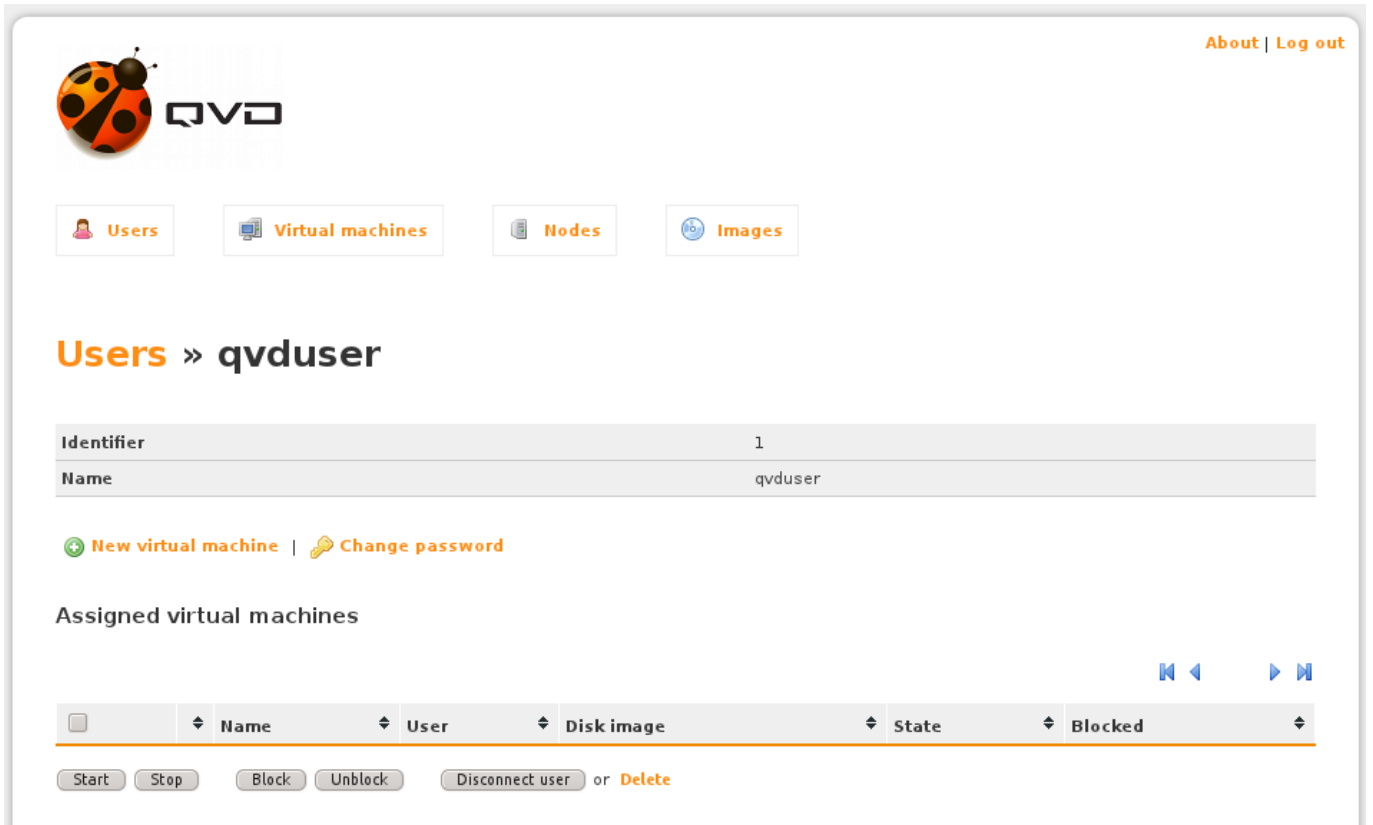


Figure 6.8: Editing a User in the QVD-WAT to Attach a Virtual Machine

At this point, there are no Virtual Machines set up within QVD. We can add a new Virtual Machine and assign it directly to the current user from this page. Click on the **New Virtual Machine** link.



Figure 6.9: The New Virtual Machine link

You will arrive at a screen that looks like this:

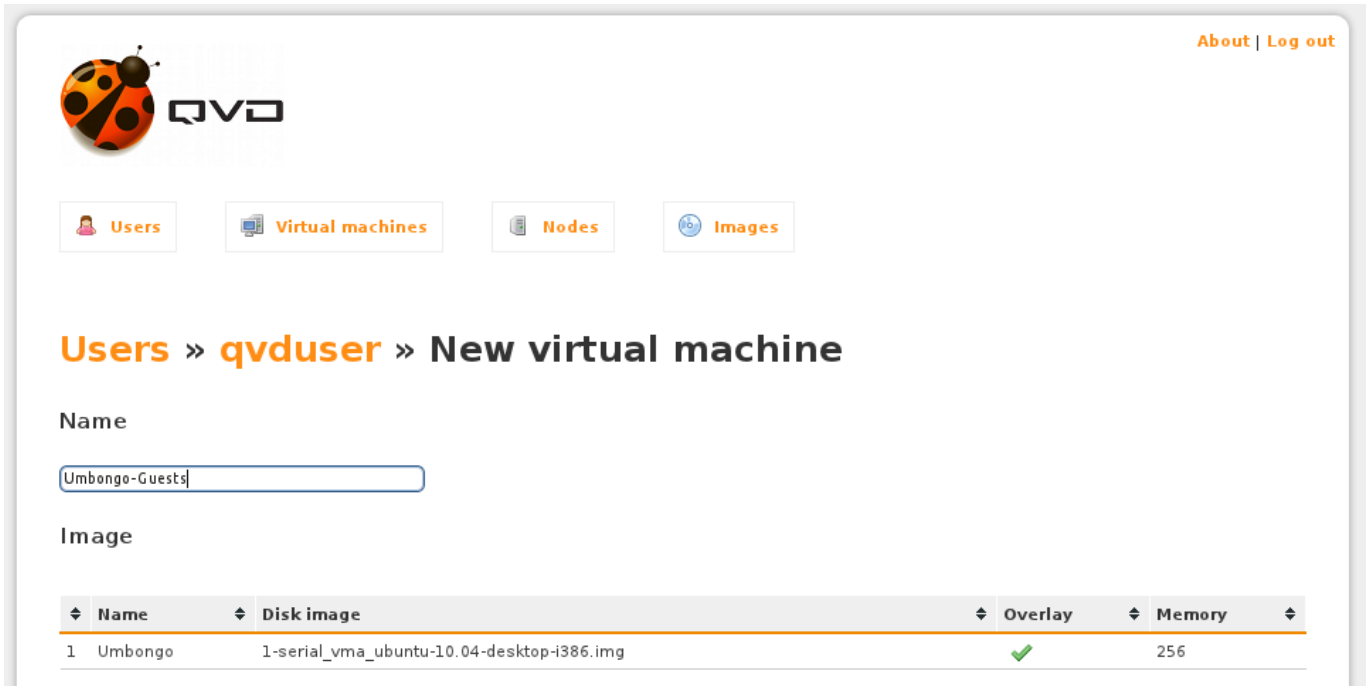


Figure 6.10: Adding a Virtual Machine to a User in the QVD-WAT

Here, you only need to provide a name for the Virtual Machine instance, and select the image that should be loaded into the Virtual Machine.

In the **Name** field, enter a useful name for the Virtual Machine. Usually something like a combination of the Username and the Image name would make sense here. So something like *QVDUser-Accounting*.

Now simply double click on the Image that you wish to load into this Virtual Machine.

If everything behaves correctly, this should bring you to a page that looks like this:



Figure 6.11: The Virtual Machine added to a User in the QVD-WAT

This page is also accessible by clicking on the **Virtual machines** link in the navigation bar, or by going to <http://localhost:3000/vm/>.

From this page, it is possible to see the status and manage any Virtual Machine within QVD. At this point, we want to see QVD in action, so we will not spend any more time discussing the options available here.

Simply check the checkbox next to the listed Virtual Machine and then click on Start. After a minute or two, you should see the Virtual Machine *State* change through various phases of the boot process. If everything goes well, you should eventually see that the *State* changes to *running*.



<input type="checkbox"/>		Name	User	Disk image	State	Node	Blocked
<input type="checkbox"/>	 4	Umbongo_Guests	 qvduser	Umbongo	 running	shamash	

Figure 6.12: The Virtual Machine has started and is running

From this point, you should be able to connect to QVD using a client and connecting as the user that you have just created.

## Chapter 7

# Installing and Configuring QVD Client

The QVD client is available for both Linux and Microsoft Windows platforms. Certainly, we recommend that for more seamless integration you should consider running the client on a Linux platform. However, it is quite possible that you have users that make use of both environments and would like to run a virtualized Linux desktop from within Windows.

Whatever your choice of platform to run the client application, it is best that you run it on a different system to the one that you are using to run server side components. This will give you a much better picture of how the whole environment works.

### 7.1 Downloading and Installing the Windows Client

If you are using Microsoft Windows as your base platform to run the QVD Client application, you will need to download the QVD Client installer manually. You can download the installer from:

<http://qvd.qindell.com/qvd3/windows/qvd-client-3.0.0-1-setup.exe>

Once you have finished downloading the installer, run it as a normal executable file and follow the wizard through the installation process.



Figure 7.1: The Windows QVD Client Installer Wizard

Once you have finished the installation, you can either run the client from the shortcut on your Windows desktop (if you selected to add the shortcut) or from the QVD menu in your Applications menu. This will open the client so that you are ready to connect.



Figure 7.2: The Windows QVD Client

## 7.2 Downloading and Installing the Ubuntu Client

Installing the QVD Client on an Ubuntu Linux platform is simple. You should add the QVD repository to your apt repository sources. Run the following commands as root:

```
# echo 'deb http://qvd.qindell.com/debian lucid main' >> /etc/apt/sources.list
# apt-get update
```

You will now be able to install the client with the following command.

```
sudo apt-get install qvd-client
```

Depending on your Desktop Environment, you should be able to access the client within your *Applications* menu, usually under the *Internet* submenu. Alternatively, you can run the client GUI from the console using the command `qvd-gui-client.pl`.

## 7.3 Connecting to your Virtual Desktop

Once you have the GUI client running, you can enter the **Username** for the user that you created in QVD, the **Password** that you configured for the user, the **Server** hostname or IP address for the QVD Server Node that you created, and you can choose the level of compression for the connection by selecting a **Connection type**.



The image shows a web-based form for connecting to a Quality Virtual Desktop (QVD). At the top left is a large ladybug logo. To its right, the text "QUALITY" is written in a large, bold, black font, with "VIRTUALDESKTOP" in a smaller, spaced-out font below it. Below the logo and text are four input fields: "User" with the text "qvduser", "Password" with a masked field of ten dots, "Server" with the IP address "192.168.0.12", and "Connection type" with a dropdown menu showing "Local". A large "Connect" button is positioned below these fields. The entire form is enclosed in a light gray border.

Figure 7.3: Enter the details for your QVD connection

By default, the **Connection type** is set to *Local*. This setting is appropriate for connections over a local area network. There are also options for *ADSL*, which would be appropriate for any broadband connection, and for *Modem* which can be used in cases where bandwidth is severely limited or impaired.

Changing the **Connection type** will increase the compression used to deliver the virtual desktop across your network connection. It also increases the amount of caching that the client performs to limit the amount of screen refreshing that needs to take place.

In general, using heavy compression and caching will still afford your users the ability to work comfortably within their virtual desktops. However the quality of graphical rendering will be a little inferior.

Once you have completed entering your connection details, simply click on the button labelled **Connect** and your virtual desktop should load.

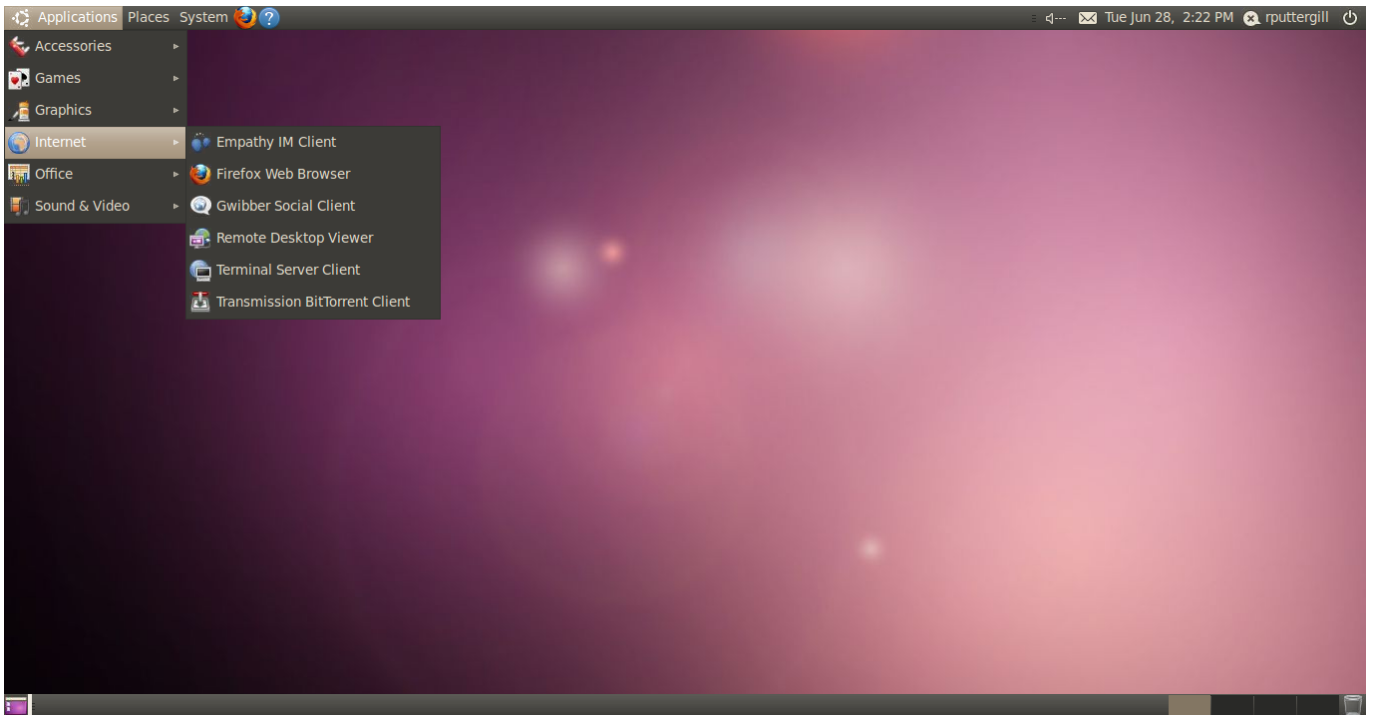


Figure 7.4: A Gnome desktop loaded under QVD

## Chapter 8

# Additional Notes about Server Side Administration

At this point, you should have a functional installation of all of the QVD Server Side components. If you connect to the server with a client, you will be able to access your virtual desktop. Nonetheless, you may still have some other questions. This section is here to help provide you with some answers and some pointers toward documentation that may help you to resolve any additional requirements.

### 8.1 Creating Your Own Image

In this guide, we took advantage of a pre-built image and simply loaded it into QVD. At this point, you are more than likely asking yourself whether you can create your own images. The answer is a resounding *Yes!*. We provided an image so that you could get started very quickly to evaluate our product.

The steps to create your own image are relatively easy to follow and are documented fully in the *QVD Administration Manual* in the chapter titled *OSI Creation*.

An OSI needs to run the QVD Virtual Machine Agent (QVD-VMA). Currently, QVD has packages for Ubuntu and SUSE Linux. However, the QVD-VMA is simply a perl script that can run on any Operating System that supports Perl and that can run an NX Server. This means that you can create an OSI for any Linux distribution or for Solaris.

### 8.2 Making Changes To An Image

If you're happy with the current image, or you have already created your own, you may want to know how you can edit this image to include different applications, or to change behaviours within the image. If you have already tried to do anything like this connected to your virtual desktop using the client application you will have realized that changes within the base operating system image are not persistent. That is, when you restart the image, any changes are lost.

This is actually one of QVD's strengths. It helps to protect all of your users from damaging changes and ensures that if any desktop is compromised the damage only lasts for as long as the desktop is running.

However, it is possible to change an image and to install new applications. In order to do this, you effectively run the image in a different mode. This will allow you to make changes to the underlying image file, so that when it is reloaded all users who access the same image will see the changes.

The ability to switch modes so that you can edit your OSI is available within the QVD-WAT. The process for editing an image is fully documented in the *QVD Administration Manual* in the chapter titled *Editing an OSI*.

## 8.3 Authenticating Users Against LDAP

You may already have your own authentication infrastructure in place, particularly if your organization makes heavy use of Microsoft Windows and Active Directory, or if you use some other LDAP service to manage your users. In this guide, users were created within the QVD-WAT and added into the PostgreSQL database. However, it is relatively trivial to point QVD at an LDAP resource to handle the authentication of users.

This is controlled by configuration keys in the QVD database that can be changed using the CLI Administration Tool.

For example, to set up authentication against an LDAP server that runs on the machine `aguila` on the port 3389 with search base `dc=example,dc=com` you would execute the following command:

```
# qvd-admin.pl config set 17r.auth.mode=ldap 17r.auth.ldap.host=aguila:3389 17r.auth.ldap. ←  
base=dc=example,dc=com
```

When a client attempts to authenticate, the QVD Server Node will connect to the LDAP Host and perform a search for the user with the `uid` attribute matching the provided username. The LDAP search is performed with a default scope of `base`. Once a match is found, a typical BIND request is performed for the user and if successful, the user is authenticated.

It is possible to change the default values for the scope and filter used in the LDAP search, to match your environment. For instance, you might do the following using the CLI Administration Tool:

```
# qvd-admin.pl config set 17r.auth.ldap.filter=(cn=%u) 17r.auth.ldap.scope=sub
```

Currently, you still need to add all of your users within the WAT, since you will need to assign the users each to a virtual machine. However, this facility will simply ensure that users are authenticated against your LDAP repository.

It is possible to automate the provisioning of users and virtual machines from LDAP to QVD and there are tools available to do this, but they are not released with the Open Source Edition of the software. Further details are available within the *QVD Administration Manual*.

## 8.4 Managing A Virtual Machine As An Administrator

If a Virtual Machine doesn't start properly or you need to access it directly as an Administrator, QVD provides a virtual console, SSH access and can even offer VNC access to a virtual machine.

For situations where a virtual machine doesn't start, it will more than likely be shut down and will enter a *blocked* state so that clients are unable to connect to it. An Administrator is able to *unblock* a virtual machine and restart it. Using telnet, the Administrator can then view the boot process as if it a serial console. If the machine manages to get past the boot phase and networking is loaded correctly, the Administrator can use SSH to access a terminal within the virtual machine.

If a virtual machine keeps failing to boot and continually enters a *blocked* state, an Administrator may wish to prevent the House Keeping Daemon from shutting down the virtual machine. This can be achieved by adding the following line to the QVD Server Node configuration file at `/etc/qvd/node.conf`.

```
internal.vm.debug.enable = 1
```

The QVD Server Node will need to be restarted for this change to take effect:

```
# /etc/init.d/qvd-node restart
```

Now, when a virtual machine fails to start, it will enter a *debug* state that will allow an Administrator to access it long after it has failed.

These topics are fairly advanced and are covered in much more detail within our *QVD Administration Manual*.

## Chapter 9

# Conclusion

In this guide, we have stepped through a basic installation and configuration of all of the components within a QVD solution. Hopefully, by following the guide you have managed to get your own virtual desktop solution running and have been able to connect to it using a QVD client.

QVD can be used for a wide range of purposes and scales incredibly well, making it the first choice for a desktop virtualization platform within the enterprise. Its remote management facilities, its ease of integration with other technologies and its ability to service remote users in a safe and secure manner will help you to improve the management of your Linux and Solaris users and to reduce the costs associated with desktop virtualization.

We hope that this demonstration has helped you to get started with QVD and that you will continue to explore its capabilities in the future.

If you have any queries or require additional support, please visit our website at <http://theqvd.com/> or contact us at [info@theqvd.com](mailto:info@theqvd.com).