1 Introduction

This document describes what is now a mostly-complete set up. All work so far is documented, in rough form, in CSCF RT numbers 57854, 60251, 60252, 61396, 61548, 61873, 61841, 63445, 61841, 63445, . The Blade Centre shall be referred to by its domain name, muscat01; external personnel will need to use the machine’s fully qualified domain name (FQDN) muscat01.cs.uwaterloo.ca, although muscat.uwaterloo.ca is a CNAME (alias) for that FQDN. It should be noted that while there are actually two Blade Centre H chassis associated with the installation, the second one should be considered part of the first.

2 Physical Setup

The Blade Centres are physically installed at the University of Waterloo in DC3556. They and their associated disk arrays draw power from a pair of UPSes, with approximately 10-15 minutes of runtime. (Note that currently the machines are not set up to automatically shut down in the event of a power failure. They will simply run until the battery life in the UPSes is exhausted, and then lose their power as a result. This is planned to be fixed in some manner.) It draws power directly from the mains (2xXXX FIXME) – it is not currently on a UPS. Networking is currently provided by two 100mbit uplinks to a CSCF-managed switch; one of these links connects to a dumb gigabit switch and is used for managing the system controllers, the other is plugged directly into the Blade Centre. (See below.) Each link is on a different VLAN. There are three cables for the system controllers — one is for muscat itself, the other two are for the system controllers on the disk arrays. NB: we should upgrade the uplink for the external interface to gigabit, given that it’s possible a lot of traffic will travel through this interface. We have an agreement with CSCF Infrastructure on this, but will need to find a free switch.

The Blade Centres have two internal switches each, each with six external copper RJ45 ports. These can be thought of as plugging directly into the two network devices on the blades themselves — Linux will see them as eth0 and eth1. The first switch has one port in use, which is plugged into a switch port on VLAN7, so any machine activating its first network device will see that network. A short patch cable is used to connect the second switches on each BC, so the eth1 devices can talk to each other on the cluster’s private network.

There is a DS4200 Express SAN attached to the BC by fibre. It contains 13 disks, six of which are allocated to the two arrays and one of which has been assigned as a hot spare, leaving six disks free. (XXX FIXME XXX)

3 Logical Setup

It is intended that some of the “worker” blades will be considered stable, and others experimental. The stable ones will not be changed often, and will be considered production servers in the sense that reboots will be controlled and planned in advance, co-ordinating with the group as a whole. The head node will live
within the stable group. The experimental ones will be released into the direct control of researchers who require them, and those researchers will not be expected to necessarily co-ordinate with the group so long as they have control of those blades.

4 Accounts and Authentication

For the stable group, authentication is done via LDAP running on the head node. We will experiment with creating another, separate, set of LDAP entries for the unstable group. Access to the stable group will be a necessary condition for access to the unstable group, as users will be required to log in to the head node before they will get access (from a network perspective) to the unstable group. (*Caveat:* it is possible to work around this, but the workaround is definitely not the preferred state of being.)

5 Operating System

We will run OpenSuSE by default on the stable nodes, as it is relatively well-understood here and is fairly friendly to Xen virtualisation. The unstable nodes may run any operating system supported by the hardware. While it is desirable that the head for the stable nodes run the same OS as its clients, it is not absolutely necessary. The current release of OpenSuSE is 10.3.

6 Current Status

Blade 1 has been set up with OpenSuSE 10.3 and can be reached from the world by the name *muscat01.cs.uwaterloo.ca*. It currently has NFS, LDAP, DNS, DHCP, and NTP services running for the private interface. It mounts two filesystems from the DS4200 SAN — /home and /scratch-net. Those filesystems are exported via NFS to the stable cluster. There is also a local /scratch directory.

Blades 08-28 have been configured as NFS/LDAP clients of Blade 1, and were installed using System Imager. Each mount /home and /scratch-net from the NFS exports, and have a local /scratch directory as well.

Blades 02-06 have been allocated to Tao Zheng to use in experiments, using various public and private hostnames and IPs. Blade 2 can use *muscat02.cs.uwaterloo.ca*.

10 more publicly addressable IPs and hostnames have been allocated for testing. They are on the same network as *muscat{01,02}* (7), and are called *cerastXX.cs.uwaterloo.ca*, where XX is 01 through 10. Tao is using these for addressing VMs.

Two RAID5 volumes have been created on the first DS4200, for use by *muscat01*. One is approximately 1.3TB, the other is approximately 1TB. They are seen by the blade and mounted as filesystems on local disks. A third volume of approximately 950GB will be allocated for storage of virtual machine libraries.

With respect to the internal ‘private’ network, a DNS zone has been created on *muscat01*. One IP has been allocated per blade *muscat02-28* of the form 192.168.143.xx, where xx is the node number. Further IPs can be allocated as required. (*See [I](#) for current assignments of hostnames and IPs.*) If that is insufficient we will need to re-think our network design. Ideally we can stick with a 192.168.x.y/24 though, as 172.19.x.y and 10.x.y.z are both routed on campus or in CS and could potentially conflict with our own choices.

7 User Notes

This section is not intended to be an exhaustive HOWTO on using Linux, Xen, VMWare, or any other thing mentioned here. Rather, it expresses the basic philosophy of usage.

7.1 Access to the Cluster

To access the cluster nodes, first log in to the head node of the cluster, which is *muscat01.cs.uwaterloo.ca*. For example, using ssh:
Once you are logged in to the head node, you can log in from there to any of the other cluster nodes. Any real work (like creating virtual machines) should be done on the regular cluster nodes, not on the head node.

There are currently two kinds of cluster nodes: stable nodes, which are for general use, and unstable nodes, which are dedicated to particular users. Unless you have arranged for dedicated access to unstable nodes, you should be logging in to one of the stable nodes.

### 7.1.1 Stable Nodes

Currently, the available stable nodes are `muscat08.private` through `muscat28.private`. You home directory should be accessible from any of these nodes, as well as from the head node. In addition, all nodes authenticate using an LDAP service, so your account password is the same on every node.

You may find it convenient to create an SSH key to simplify the login process. Since home directories are shared amongst all of the cluster nodes, copying one’s `id_rsa.pub` to `.ssh/authorized_keys2` will automatically grant access to all nodes. Users wanting to do this should refer to documentation for the `ssh-keygen` command. Note that a key without a passphrase may be used in order to grant more convenient access to the stable nodes, but it is not recommended that a passphrase-less key be used to gain access to the head node from the outside.

### 7.1.2 Unstable Nodes

There are few hard and fast rules regarding the unstable nodes, beyond it is preferable that they expose as few services as possible to the outside world. Ideally, access to the machines should be restricted to the private network. Users are expected to communicate amongst themselves in order to arrange who gets which nodes and for how long.

System administrator assistance, in the form of the CSCF Research Support Group point of contact, is always available and should be consulted for any questions, particularly if network services are required or desired.

### 7.2 Xen Virtualization

Most users are members of the `xenusers` group on all nodes. This gives you permission to use `sudo` to run the `xm` command. The `xm` command is used for Xen virtual machine management operations, like creating virtual machine instances or listing the instances that are currently active. For example, to list the virtual machines running on a node, use:

```
% sudo /usr/sbin/xm list
```

This may prompt you for your password. Note that you must use the full pathname `/usr/sbin/xm`.

More information about Xen, including documentation for `xm`, can be found at [http://xen.org/](http://xen.org/)

### 7.3 Virtual Machine Repository

There is a directory called `/home/vmlib` which is accessible to everyone in the xenusers group, from every cluster node. The idea is that anyone can create a subdirectory under `/home/vmlib` to hold virtual machine image or configuration files that they wish to share.

### 7.4 Networking for Virtual Machines

There are only a limited number of IP addresses available for virtual machines in the cluster. Having multiple virtual machines using the same IP address (or the same MAC address) will cause networking problems.
As a workaround, we have allocated specific MAC/IP address combinations to specific users. When you create virtual machines, use the MAC/IP addresses that have been allocated to you. Appendix C of this document lists the current address allocations. If you need more addresses, let Ken Salem know.

8 Other notes for sysadmins

8.1 Network setup, SAN

All the management hosts are addressable by private network. They are best reached using a web browser; all require a robust Java environment installed. Windows (either XP or Vista) with IE7 seems to work well, but some installations of Ubuntu and CentOS have also been used. I have had very little luck trying to use a Mac OS 10.4 or 10.5 system.

- **muscat-mgmt.cs**: Management module for the first Blade Centre. From here one can use the remote KVM attached to the blades. It is also possible to jump from this MM to that of the second BC.
- **muscat2-mgmt.cs**: MM for the second Blade Centre. Use this to access the consoles for blades 15-28.
- **muscat-sc01**: first system controller
- **muscat-sc02**: second system controller
- **muscat-fc01.cs**: SAN channel controller
- **muscat-fc02.cs**: SAN secondary channel controller
- **muscat2-sc01.cs**: second BC’s system controller
- **muscat2-sc02.cs**: second BC’s secondary system controller

The SAN controller should only be used to configure the disks, and there should be no need to connect to the system controllers — if you need to talk to those, do it through the MM.

Passwords for these modules can be found in The Usual Place. Please contact the CSCF RSG point of contact if you’re a user who needs to be able to configure the disks, or to connect to the KVMs. Users with access to the KVMs are currently Umar Minhas and Tao Zheng.

One also needs the Storage Manager software (available from IBM’s website) in order to configure the disks in the DS4200. SM runs well under CentOS, OpenSuSE, or Windows, and is installed on muscat01. Windows XP users should get the Windows 2003 version of the Storage Manager software, and there is also a Vista version available. Current version is 10.1. Older (9.60) installations will no longer work.

8.2 DNS

Admins will have to configure the DNS information by hand; using the GUI tool currently seems to break the setup for reverse DNS. BIND configurations are stored in `/var/lib/named/master`. The Appendix lists which IPs are currently allocated on the private network, and for what purpose.

8.3 CLI package management

OpenSuSE 10.3 has a new CLI tool called **zypper** which may seem familiar to anybody who’s used **rug** from previous releases of SuSE, or **apt** from Debian-clones. Main commands are things like **zypper repos**, **zypper search packagename** and **zypper install packagename**.
8.4 Adding Accounts

The LDAP setup has notes included in RT#60251. It is possible to use the *yast2* tool in order to add new users, but there’s also a manual way that may work better. I’ve created tools and stashed them in `/root/people` on `muscat01`. The first thing one needs to do is figure out what userid, uid, and password to use for the new user. It is strongly recommended that we stick with using CS uids whenever possible, and always truncate to 8 characters for these userids. These can be retrieved from any core CS machine with the `idregistry` command, like so:

```
mpatters@muscat01:~/people > idregistry request mpatters
mpatters:1633
```

So, the uid for the userid `mpatters` is 1633.

The first tool to use is called `adder.pl`, and is called like this: `adder.pl userid uid password`. The password string should be encrypted, although if you’re not sure what to put here, `' '` should do; you can later change it with the `passwd` command. `adder.pl` simply creates an LDIF file that can be added to the LDAP database using a different script named `addtoldap.sh` – provided you know the LDAP admin password. Here you can see Real Output™ from adding a user. We already know the user’s uid is 6092.

```
muscat01:~/.people # ./adder.pl t3zheng 6092 ''
muscat01:~/.people # ls -ld t3zheng.ldif
-rw-r--r-- 1 root root 277 2008-02-13 16:12 t3zheng.ldif
```

In this situation, where we’ve set the user’s encrypted password to a space, we should immediately change it to something we know. If you need a good method for generating passwords, the *apg* program is commonly available (although not on our installation). In any event, you should immediately remove the LDIF file that was created in this process.

If the user is not a member of the UW community, choose a uid in the sequence starting with 1001. Ideally their userid will not potentially conflict with a UWDir userid, but it’s not a disaster if it does. The usual rules for UWDir are First Initial Middle Initial Lastname, so Henry Aaron Bloggins would be habloggins, which would then be truncated to habloggi. However, the LDAP configuration on `muscat01` does not enforce any particular userid restrictions.

The fastest way to determine which uids are in use on the system is to use `slapcat`, something like this (note that one needs to have `sudo` privileges):

```
mpatters@muscat01:~ > sudo /usr/sbin/slapcat | \
  grep uidNumber | awk ’(print $2)’ | sort -n
1001
1633
5888
6092
14499
```

8.4.1 The GUI Way of adding users

*yast2* actually has two GUls, one is ncurses based and the other X11. It would appear that the ncurses tool is not full-featured, which causes problems given that we would like to be able to set a user’s uid manually, and also it does not seem to accept long passwords. Use the ncurses version at your own risk. A quick note about X11 and root/sudo: use the `-E` option for sudo, then run `xauth merge /home/userid/.Xauthority`. This will allow your new rootshell to use your X11 tunnel cookies.
After starting yast2, you should get a window titled “YaST Control Center”, and then click (from the left) ‘Security and Users’, which will bring you right to that section. (Otherwise you could just scroll down.) Next, select ‘User Management’. It should read in a bunch of settings, then throw you into a screen like this:

![User and Group Administration](image1)

**Figure 1:** User/group administration in yast2

You will want to use the ‘Set Filter’ button to restrict the user list to ‘LDAP Users’, and you will then be prompted for the LDAP server password. The display should update and look something like this:

![User and Group Administration](image2)

**Figure 2:** LDAP User/group administration in yast2
Click ‘Add User’, then fill in the User Data tab. To set the uid and group memberships, choose the Details tab. \(NB:\) the ncurses yast tool does not seem to allow this.) The user we’re adding is named y6hu, has a uid of 1736, and we want this user to be able to start Xen machines:

![Image of Adding a user to LDAP database, details pane](image)

When you’re happy with everything, click ‘Accept’, then ‘Finish’. Since you’ve just set the user’s password, you should be able to try logging in yourself. If all goes well, then you’re set.

### 8.4.2 Creating sudoers

To allow a user to use `sudo` on any stable node, add that user to the `stableadmins` group on the head node. This can be done using the standard OpenSuSE group admin tools in `yast2`. This will allow users to run any command on any of the stable nodes.

\(NB:\) This does not include the ability to sudo on the head node – for that, add the user to the group `wheel` on the head node itself.

Sometimes adding the user to a group doesn’t immediately ‘take’ on client nodes. Rebooting the client node will force this; restarting the nscd service (`/etc/init.d/nscd restart`) may cause it to happen more quickly and less destructively. This is some sort of cache issue that needs to be resolved.

### 8.5 Working with System Imager

RT#61548 is setting up SI on `muscat01`. Things to keep in mind are the machine will need an entry in `/etc/dhcpd.conf` if it doesn’t already, as well as `/var/lib/named/master/*`, and also look in `/var/lib/systemimager/scripts`. Copying the format of other machines is fine. The current image is actually called `Test02`, and the `scripts` directory looks like this:

```
mpatters@muscat01:/var/lib/systemimager/scripts> ls -l
total 68
-rw-r--r-- 1 root root 210 2008-02-13 15:00 hosts
lrwxrwxrwx 1 root root 13 2008-03-12 13:23 muscat12.sh -> Test01.master
```
To prepare the golden client machine, you need to do something like `si prepareclient --server 192.168.143.1 --kernel /boot/vmlinuz`. Once that’s completed, you need to do something like `si getimage --golden-client 192.168.143.114 --image Test02 --exclude '/media/*' --exclude '/scratch*' --exclude '/home/*' --exclude '/tmp/*'`. Previously there have been issues with networking ceasing to function shortly after the node boots up; this has currently been attributed to several factors and has, we believe, been resolved. See the notes in RT 64335, dated 10 October 2008, for more details.

In order to PXE boot a client to re-image it, 3 services need to be running on `muscat01`, you can start them like this:

- `service pxe start`
- `service systemimager-server-rsyncd start`
- `service systemimager-server-monitord start`

Once you are done imaging your new client(s), shut the services off again, so a machine that is accidentally PXE-booted won’t be automatically re-imaged.

### 8.6 Working with the Management Module

If you want to boot a blade off a local ISO image, you need to hit F12 and choose CDROM. If you’re booting a floppy disk image, choose local diskette. You need to make sure that the media tray is set to the machine you want to boot, even if it’s a local image.

The console can sometimes behave very oddly, becoming off-centre or just plain not displaying much at all. Unfortunately, that seems to be partially a consequence of running an unsupported operating system. Sometimes clicking the Paint or Calibrate buttons can help. If not, try changing virtual consoles (Linux host Ctrl-Alt-Fx) or the blade to which the remote console is attached.

### 9 Network layout

In the diagram, the 192.168 addresses are assigned always to the second interface on the blade; in Linux, it will be known as `eth1`. Care needs to be taken in order to avoid having blades other than 01 turn into bridges onto the private network. Ideally, the `eth0` interface will not be brought up on any blade other than the first, although some software management may require this be done. While it is possible to disable the primary (or secondary) interface for individual blades through either the MM or the BIOS of the blade, this is not recommended as it can lead to confusion with respect to device naming.

Not shown here is the connection to the DS4200s. Technically every blade has the ability to see it; however, certain of the storage partitions (the one containing the RAID5s known as `home` and `scratch-net`) should never be accessed by any machine but `muscat01`. Doing otherwise is almost certain to cause catastrophic data loss.

Generally speaking, we should avoid placing blades on the publicly addressable network if at all possible. This reduces the risk of intrusion, as well as the risk of accidentally creating a possibly uncontrolled bridge between the public and private networks.
A Glossary

This is not intended to be an exhaustive or authoritative listing of definitions. Rather, it will present some terms used in this document within the context of how they apply to the CERAS Blade Centre.

**FQDN**: Fully Qualified Domain Name, a hostname followed by the domain name. For instance, muscat13.private is the FQDN for the host muscat13 in a domain named ‘private’. www.cs.uwaterloo.ca is the FQDN for the host named www in a domain named ‘cs.uwaterloo.ca’.

**LDAP**: Lightweight Directory Access Protocol. Commonly used as it is on the stable nodes, with an authoritative server answering requests for authentication from clients. It is generally used to supplement, not to replace entirely, old constructs such as /etc/{passwd,shadow,group}.

**MM**: Management Module, the device on a Blade Centre chassis through which one may control individual blades physically installed in that chassis. Various actions like powering up, down, and taking control of the console are possible.

**subnet**: one may divide big networks into smaller ones, called subnets. The uwaterloo.ca network is what used to be called a Class B network, encompassing all IPs in the range 129.97.0.0 - 129.97.255.255; CIDR classifies this as 129.97.0.0/16. For convenience, the university has divided this into many subnets, mostly /24s: 129.97.7.0/24, for example, has been assigned to Computer Science. IANA has also assigned several networks for use on private nets: 192.168.0.0/16 is one such, and the muscat network uses 192.168.143.0/24 for intra-blade communication.

B Hardware Specifications

**muscat** consists of two IBM Blade Center Model H chassis. They contain 28 blades, model number LS-21. Each blade has 2 AMD dual-core 2212 HE CPUs at 2.0 (1-14) or 2.2GHz (15-28), at least 8GB of RAM, and a single 36GB internal disk. Some blades have greater than 8GB of RAM; these will be documented in due course.
## Node Assignments

Blades are assigned to individual users. Users should only log in to the head node and to those blades that they have been assigned.

<table>
<thead>
<tr>
<th>Blade Name</th>
<th>Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>muscat01</td>
<td>head node</td>
</tr>
<tr>
<td>muscat02</td>
<td>cavram (Hadoop/edge)</td>
</tr>
<tr>
<td>muscat03</td>
<td>cavram (Hadoop/edge)</td>
</tr>
<tr>
<td>muscat04</td>
<td>arajabi</td>
</tr>
<tr>
<td>muscat05</td>
<td>–</td>
</tr>
<tr>
<td>muscat06</td>
<td>t3zheng</td>
</tr>
<tr>
<td>muscat07</td>
<td>–</td>
</tr>
<tr>
<td>muscat08</td>
<td>–</td>
</tr>
<tr>
<td>muscat09</td>
<td>ak5singh</td>
</tr>
<tr>
<td>muscat10</td>
<td>x39liu (MySQL)</td>
</tr>
<tr>
<td>muscat11</td>
<td>x39liu (MySQL)</td>
</tr>
<tr>
<td>muscat12</td>
<td>r46liu (Cassandra/DAX)</td>
</tr>
<tr>
<td>muscat13</td>
<td>arajabi</td>
</tr>
<tr>
<td>muscat14</td>
<td>r46liu (Cassandra/DAX)</td>
</tr>
<tr>
<td>muscat15</td>
<td>rgarcia</td>
</tr>
<tr>
<td>muscat16</td>
<td>r46liu (Cassandra/DAX)</td>
</tr>
<tr>
<td>muscat17</td>
<td>r46liu (Cassandra/DAX)</td>
</tr>
<tr>
<td>muscat18</td>
<td>r46liu (Cassandra/DAX)</td>
</tr>
<tr>
<td>muscat19</td>
<td>r46liu (Cassandra/DAX)</td>
</tr>
<tr>
<td>muscat20</td>
<td>arajabi</td>
</tr>
<tr>
<td>muscat21</td>
<td>ufminhas (VoltDB)</td>
</tr>
<tr>
<td>muscat22</td>
<td>ufminhas (VoltDB)</td>
</tr>
<tr>
<td>muscat23</td>
<td>ufminhas (VoltDB)</td>
</tr>
<tr>
<td>muscat24</td>
<td>ufminhas (RemusDB)</td>
</tr>
<tr>
<td>muscat25</td>
<td>ufminhas (RemusDB)</td>
</tr>
<tr>
<td>muscat26</td>
<td>ufminhas (VoltDB)</td>
</tr>
<tr>
<td>muscat27</td>
<td>ufminhas (VoltDB)</td>
</tr>
<tr>
<td>muscat28</td>
<td>ufminhas (VoltDB)</td>
</tr>
</tbody>
</table>
D Name/IP assignments

IP addresses of the form 10.0.x.x are reserved for the Eucalyptus installation.
The eth1 interfaces of all blades, and all non-Eucalyptus virtual machines, use IP addresses of the form 192.168.143.x.

The following is a list of MAC addresses, 192.168.143.x IP addresses, and hostnames that have been assigned to individual muscat users for their (non-Eucalyptus) needs. The intention is that they, and only they, will create virtual machines using assigned names and addresses. Note that the prefix for all MAC addresses is 00:16:3e. For example, the full MAC address corresponding to IP 192.168.143.150 is 00:16:3e:00:01:50.

<table>
<thead>
<tr>
<th>IP</th>
<th>hostname</th>
<th>MAC</th>
<th>allocated to</th>
</tr>
</thead>
<tbody>
<tr>
<td>.1</td>
<td>muscat01.private</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.86</td>
<td>cerast01.private</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.87</td>
<td>cerast02.private</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.88</td>
<td>cerast03.private</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.89</td>
<td>cerast04.private</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.90</td>
<td>cerast05.private</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.91</td>
<td>cerast06.private</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.92</td>
<td>cerast07.private</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.93</td>
<td>cerast08.private</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.94</td>
<td>cerast09.private</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.95</td>
<td>cerast10.private</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.102</td>
<td>muscat02.private</td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>to</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.128</td>
<td>muscat28.private</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.150</td>
<td>muscatvm01.private</td>
<td>:00:01:50</td>
<td>madalin</td>
</tr>
<tr>
<td>.151</td>
<td>muscatvm02.private</td>
<td>:00:01:51</td>
<td>madalin</td>
</tr>
<tr>
<td>.152</td>
<td>muscatvm03.private</td>
<td>:00:01:52</td>
<td>marin</td>
</tr>
<tr>
<td>.153</td>
<td>muscatvm04.private</td>
<td>:00:01:53</td>
<td>marin</td>
</tr>
<tr>
<td>.154</td>
<td>muscatvm05.private</td>
<td>:00:01:54</td>
<td>slavescu</td>
</tr>
<tr>
<td>.155</td>
<td>muscatvm06.private</td>
<td>:00:01:55</td>
<td>slavescu</td>
</tr>
<tr>
<td>.156</td>
<td>muscatvm07.private</td>
<td>:00:01:56</td>
<td>pasward</td>
</tr>
<tr>
<td>.157</td>
<td>muscatvm08.private</td>
<td>:00:01:57</td>
<td>pasward</td>
</tr>
<tr>
<td>.158</td>
<td>muscatvm09.private</td>
<td>:00:01:58</td>
<td>y6hu</td>
</tr>
<tr>
<td>.159</td>
<td>muscatvm10.private</td>
<td>:00:01:59</td>
<td>y6hu</td>
</tr>
<tr>
<td>.160</td>
<td>muscatvm11-16.private</td>
<td>:00:01:60-65 a2karyakin</td>
<td></td>
</tr>
<tr>
<td>.166</td>
<td>muscatvm17.private</td>
<td>:00:01:66</td>
<td>kmsalem</td>
</tr>
<tr>
<td>.167</td>
<td>muscatvm18.private</td>
<td>:00:01:67</td>
<td>kmsalem</td>
</tr>
<tr>
<td>.168</td>
<td>muscatvm19.private</td>
<td>:00:01:68</td>
<td>ufminhas</td>
</tr>
<tr>
<td>.169</td>
<td>muscatvm20.private</td>
<td>:00:01:69</td>
<td>ufminhas</td>
</tr>
<tr>
<td>.170</td>
<td>muscatvm21-45.private</td>
<td>–</td>
<td>CS848 / ashraf</td>
</tr>
<tr>
<td>.195</td>
<td>muscatvm46-50.private</td>
<td>–</td>
<td>harafiq / CS848</td>
</tr>
<tr>
<td>.200</td>
<td>muscatvm51-56.private</td>
<td>–</td>
<td>tbenke</td>
</tr>
<tr>
<td>.206</td>
<td>muscatvm57-81.private</td>
<td>–</td>
<td>t3zheng</td>
</tr>
<tr>
<td>.231</td>
<td>muscatvm82-87.private</td>
<td>–</td>
<td>pgaharwar</td>
</tr>
<tr>
<td>.237</td>
<td>muscatvm88-91.private</td>
<td>–</td>
<td>tbenke</td>
</tr>
<tr>
<td>.241</td>
<td>muscatvm82-99.private</td>
<td>–</td>
<td>unassigned</td>
</tr>
</tbody>
</table>

E UW Principals

Principals at the University of Waterloo are:

- Johnny Wong (faculty)
- Ashraf Aboulnaga (faculty)
• Ken Salem (faculty)
• Tao Zheng (post-doctoral fellow)

Technical support at UW is provided by the CSCF Research Support Group; currently the primary point of contact is Mike Patterson <mpatters at cs.uwaterloo.ca>, with Lawrence Folland <lfolland at cs.uwaterloo.ca> as backup. Users with questions or problems may also contact Ken Salem <kmsalem at uwaterloo.ca>.

F changeLog

For the documentation, not muscat.
v0.7.17 (9 Aug 2011) Muscat 12,14 to Rui
v0.7.16 (16 Dec 2010) Updates to node/IP assignments for Rui
v0.7.15 (27 May 2010) Updates to node/IP assignments for Alexey
v0.7.14 (18 Feb 2010) Updates to node/IP assignments for Xin
v0.7.13 (21 Sept 2009) Updates to node/IP assignments for Eucalyptus
v0.7.12 (11 June 2009) Updates to node assignments.
v0.7.11 (5 May 2009) Swapped Benke/Garcia nodes.
v0.7.10: (21 April 2009) Updated assignments for Tim Benke and Prashank Gaharwar
v0.7.9: (23 March 2009) Updated assignments for Sean Tozer.
v0.7.8: (4 March 2009) Updated assignments for Tao.
v0.7.7: (19 Jan 2009) Updated assignments for Ronaldo.
v0.7.6: (16 Jan 2009) Removed CS848 node assignments, add assignments for s7lu and pgaharwar
v0.7.5: (18 Dec 2008) Added IP assignments for Tao (t3zheng)
v0.7.4: (18 Nov 2008) Updated node assignments for Tim Benke (tbenke)
v0.7.3: (14, 18 Nov 2008) Updated node assignments for CS848 projects.
v0.7.2: (11 Nov 2008) Updated node assignments.
v0.7.1: (4 Nov 2008) Updated node assignments.
v0.7: (14 Oct 2008) Added physical node assignments map.
v0.6.1: (10 Oct 2008) Added notes about the console. Some copyediting. Update information on imaging new nodes.
v0.6.0: (Summer 2008) The Great Expansion plus UPS notes.
v0.5.1: two more stable nodes. Ken added the list of assigned VM IPs and some Xen user notes.
v0.5.0: added subsubsection on GUI adming of users.
v0.4.5: started in on a glossary, changes to the IP assignments.
v0.4.4: added information on stableadmin group and sudo as well as a few further notes on System Imager.
v0.4.3: added DNS information. Added information on using slapcat.
v0.4.2: minor fixes.
v0.4.1: added stub for user section.
v0.4: added documentation on System Imager and creating userids.
v < 0.4: pre-history of which we shall not speak.