CC1 - Cracow Cloud One

Release 2.0

2013, CC1 Team

April 25, 2014

Content:

CHAPTER

ONE

USER GUIDE

1.1 Introduction

The CC1 provides a complete Cloud Computing system within the model called **Infrastructure as a Service** (IaaS) based on **hardware platform virtualization**.

The aim is to **deliver a Virtual Machine** (VM) (see: Handling Virtual Machines) on demand with required operating system. VM can be considered as a real physical computer. The virtualization makes it possible to run machines with different operating systems on the same underlying hardware transparently.

The resources can be accessed via an intuitive interface based on a Web browser (see: CC1 Web interface) The system is operated in a **self-service mode**, which is one of the key features of Cloud Computing. The users can reserve, use and release resources in an automatic way, i.e. without the need to interact with a system administrator. This way Cloud Computing systems can deliver computer resources in an easy and elastic way. Elasticity means that there is no up-front commitment by users on the size of resources and their reservation time. In the case of large systems, it gives an illusion of infinite computing resources available on demand.

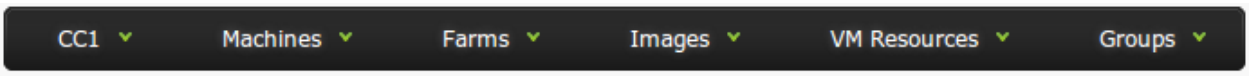
In the case of relatively small systems installed in a given organization (Private Cloud), the quotas on CPU, storage or network elements (public IP numbers) have to be enforced.

1.2 CC1 Web interface

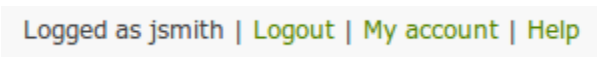
1.2.1 Main WWW interface's elements

The general layout of the CC1 interface is shown in Fig. *CC1 Web interface layout.* There are four main elements:

Main menu bar with the following items: **CC1 | Machines | Farms | Images | VM_Resources | Groups**

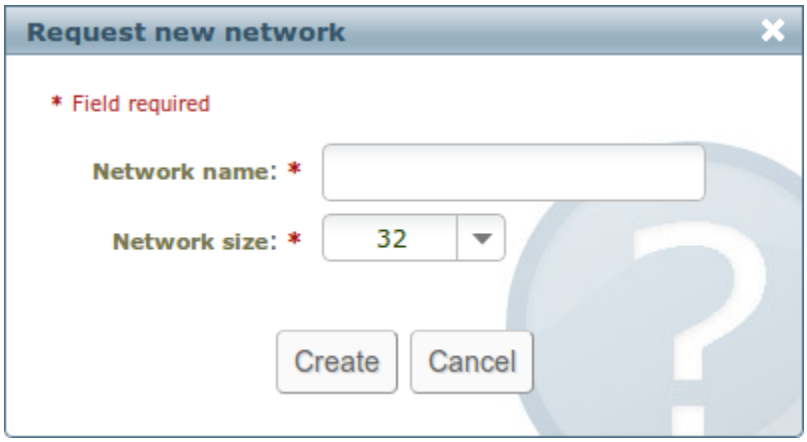


Session menu bar (top-right) with session status and some session related actions: **Logged as** *user_id* **| Logout | My account | Help**



Submenu on the left side of the page with content corresponding to the last clicked item in the main menu bar.

Information window(s) which may contain some active elements like pop-up menu, action buttons etc.



On-line hints are available by clicking on the question mark in the right top corner of the window (not present in every window). They give the basic information related to a given window.

1.2.2 Contact to administrator

The contact information is displayed after selecting **Help** (the last item in the session menu bar).

1.2.3 Interface language and available clusters

Two selection boxes in the upper right part of the interface window give a possibility to select the language (right box - English) or switch between available Cluster Managers (left box - CM).

The interface window is refreshed automatically every few seconds. A secure protocol HTTPS is used for communication with the Web interface.

1.3 Registration and account properties

1.3.1 Registration

Only registered and activated users can enter the system. The registration can be performed from the CC1 Web interface page at the address selected for a particular installation. For standard registration procedure an email exchange is

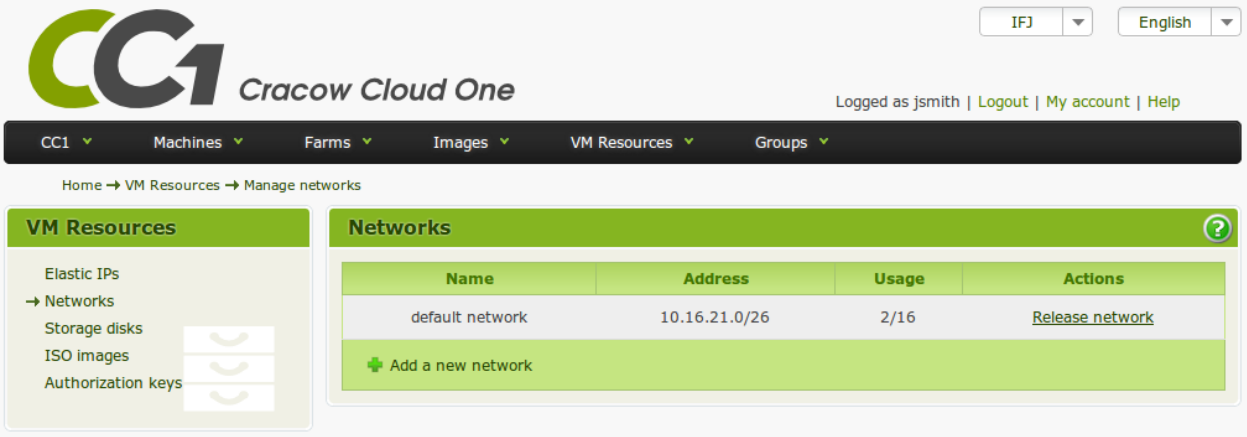


Figure 1.1: CC1 Web interface layout.

necessary to complete all steps:

1. Click on **Register** action or **Registration** menu item, fill all required fields and submit the form by pressing the **Register** button.

   Note: weak passwords are not accepted.

1. Wait for the activation mail and then click on the activation link located in a message. The second mail with the confirmation of user-side activation is sent. At this point access to the system is **not granted** yet.
2. The final step is performed by the CC1 administrator who activates the user account manually. An email informing about account activation is sent to the user.

This default registration procedure is recommended for systems available to the public. In special cases of private installations, the procedure may be different:

- Registration can take place without a confirmation via email.
- There is also an option to automatically activate of account immediately after filling out a form without the need for administrator approval.

1.3.2 Sign in

Access to system resources is granted only for users that are signed in. One must click action **Sign in** from session menu (upper right corner of the web interface). Then username and password (provided during registration) need to be entered.

1.3.3 Account properties

The properties of the account can be displayed by clicking on **My account** item in the session menu bar. There are three items: **Account data**, **Account quotas** and **Change password**. On the **Account data** page the user can view account parameters and modify some of the fields.

EC2 credentials are located at the bottom. They can be used to access the CC1 system via the EC2 interface.

Account quotas shows the current usage/quota for the number of CPU cores, RAM size, disk space (VM private images and virtual disk images), public IPs and points consumed from the beginning of the month. The initial quotas should be set according to an estimated need of an "average" user. The quota can be increased by the CC1 administrator.

Change password the aim of **Change password** is to allow user to modify his password.

1.4 Handling Virtual Machines

Virtual Machine (VM) should be considered as remote physical computer with requested hardware parameters and operating system that is specified by user.

To enable VMs management, one must be signed into CC1 system (see: *Sign in*).

1.4.1 Starting Virtual Machine

To start Virtual Machine, click on **Machines** and select **New Virtual Machine** to create a new VM instance in a simple four step procedure:

1. Select VM image.
2. Select VM template (number of CPU cores and RAM size - fig. *VM creation*).
3. Assign VM resources (disks, IP address, etc; see: VM resources).

   Note: disks and IP number can be attached to a running VM later.

4. Summary - specify name of the instance, its description and confirm with the **Create** button.

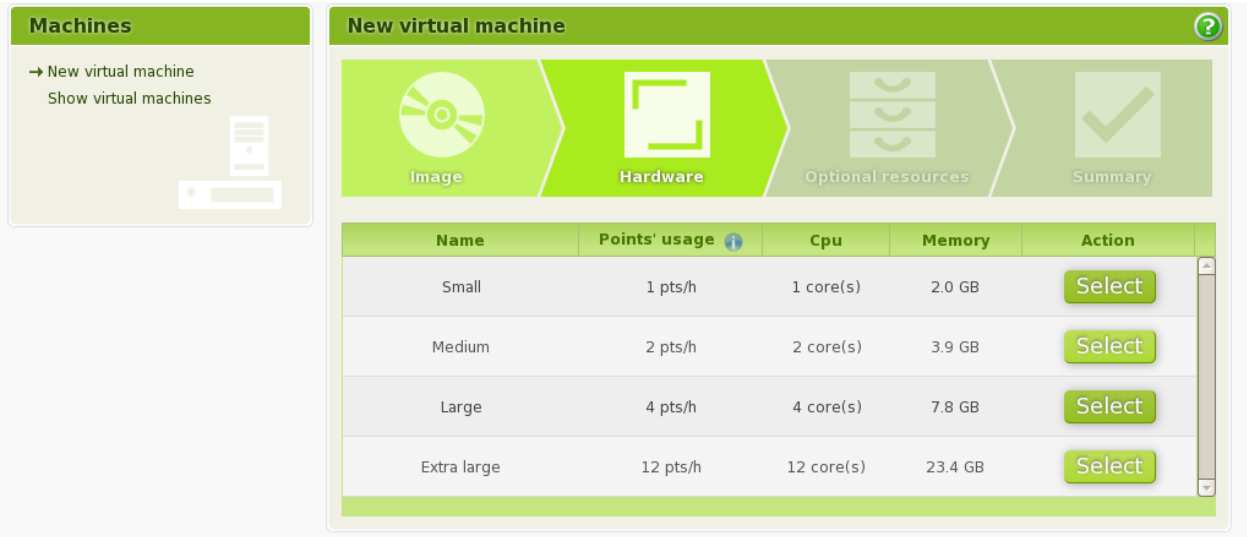


Figure 1.2: VM creation

1.4.2 Virtual Machine info

The created VM should appear in the list of active VMs with the *init* **Status** and, after a while, as *running* (see Fig. *VM list*).

Note: The time of *init* to *running* transition depends on a VM image size and system load, It takes typically a few minutes.

In the line of list of virtual machines (Fig. *VM list*) the following fields are displayed:

checkbox used to mark VM for the execution of a common action for all marked VMs together (**Perform for selected**).

Name name of VM as given by the user.

Info this field contains several icons that inform about various features of running VM, such as public IP assigned, running contextualization, enabled VNC (Sec. Controlling Virtual Machine) etc. A short description of each icon is displayed after placing the pointer on the icon.

Status gives the status of the VM like *init*, *running*, *saving*, *closing*.

Template name name of a template used to create VM. The same VM image can be run with any template. Template's parameters define:

- the number of cores,
- RAM size
- cost per hour (the number of points per hour of usage).

Those parameters are shown in the second step of VM creation process (Fig. *VM creation*).

Image name the name of the image which VM was started from. VM's name and description can be modified any time by right-click **Edit** action.

Load the load of VM. Numerical values and color codes are used to present the CPU load.

Figure 1.3: VM list

The user should be aware that a VM is started with a clone of the selected image from the pool. Any modification affects the clone only. The fate of this clone image depends on the user choice at the time of closing. The user can choose between *destroy* or *save*. The result of these actions are explained in Sec. Controlling Virtual Machine.

1.4.3 Controlling Virtual Machine

The interaction with a VM is possible from the list of running VMs, **Machines->Show virtual machines**. There are two ways. The action can be selected from the menu which appears after right click on the line corresponding to a given VM. The second possibility is left click on that line. This opens a pad with various VM informations combined with active elements (see *VM details*).

A few thematic areas that contain information and possible actions are graphically distinguished.

VM Management

Destroy the machine is closed and the clone of the image used to create VM is deleted.

> **Warning:** all modifications made to the VM are lost.

Save and shutdown closes VM and saves the image in the pool of private images. The user is prompted for a name of the image to save. No action is performed, if the storage quota is exceeded. In this case user should remove unused private images or ask CC1 administrator to increase their storage quota.

Reset corresponds to pressing a reset button of a computer. It is a brute force type of restarting VM and may lead to a corruption of the filesystem.

Edit allows the user to change the name and description fields.

Access to VM

Enable/Disable VNC to allow/forbid VNC connections.

Graphical console (VNC) starts the VNC viewer. It gives access to the VM console. The Java plug-in has to be installed in a browser.

There are clicable fields on the right hand side of the **Access** area:

Public IP - Assign/Revoke to assign/revoke elastic IP numbers. The procedure how to get a public IP is described in Sec. Assigning public IP.

VNC password - Show/hide to show a randomly generated password for VNC viewer access.

Contextualization

Set password changes the password for a specified user on a running VM.

Set SSH key injects SSH keys to login via ssh without giving the password.

Monitoring

It is recommended to close VM via interface buttons **Save and shutdown** or **Destroy**. If the *shutdown* command is executed from the operating system level, the local copy of VM image (the clone of the image the VM was started with) will be stored in the private images pool with the *_autosave* appended to the original VM name. Thus all modifications made to the VM will be saved.

1.4.4 Assigning public IP

The elastic IP mechanism ensures an efficient use of the limited pool of public IPs addresses. It is based on connecting and disconnecting of public IP addresses to running VM instance dynamically - with no need to reconfigure VM's operating system. In effect, the public IP addresses may be assigned to the particular VM right the moment VM needs it.

Any other VM machine can be accessed trough private network from a single machine possessing a public IP. The outside connectivity (the connection initiated by VM) is always assured, regardless on weather the VM has public IP or not.

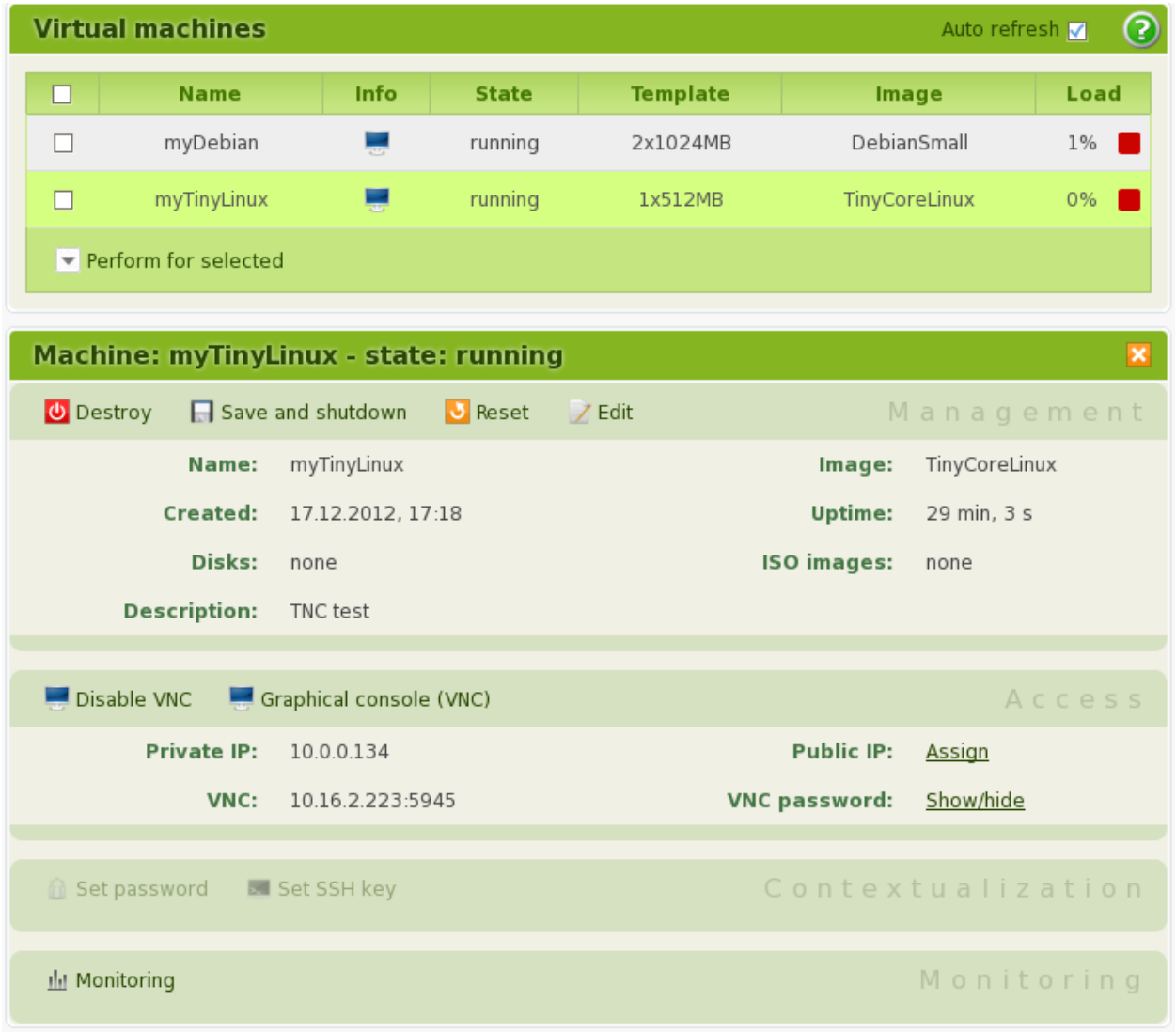


Figure 1.4: VM details

The management of IP addresses section is located in **VM Resources->Elastic IP addresses** menu. Initially the list of IP addresses is empty. One needs to request an IP number to be assigned to his pool by pressing **Request new IP** button. Aquired IP number can be assigned to any VM - whether at the creation stage or later, to a running VM.

1.4.5 Accessing Virtual Machine

There are three main ways to access VM

Graphical console (VNC) The built in VNC viewer can be started as described in Sec. Controlling Virtual Machine. Java plug-in is required to be installed in the browser. An external VNC viewer can be used as well. The connection parameters (IP number, port number and generated password) are available from the VM details window (Fig. *VM details*) in the *Access* pad.

SSH the standard remote login via *ssh* is possible after assigning public IP to a VM. The IP assignment is described in Sec. Assigning public IP. The **SSH** with X11 tunelling switch **-X** can be used to open output windows of graphics application on user's desktop.

Remote Desktop one can connect to a VM that is running Microsoft Windows via the Remote Desktop. The client side can be executed on user's MS Windows desktop or via the *rdesktop* command on Linux.

1.5 VM resources

1.5.1 VM images

There are 3 pools of VM images shown by selecting **Images** item in the main menu bar:

- **My images** - the private images seen by the owner only. A VM started from a public image can be saved into the private pool using **Save and shutdown** action. The user can upload an external image using **+ Upload image** action located at the bottom of the list of private images.
- **Public images** - the images prepared by the CC1 system administrators. They are available for all users.
- **Group images** - a special type of images marked by the owner to be a group image. The group images can be seen by the members of the group only. Note that a group image is not a copy of the corresponding private image. It still resides in the user's storage area and contributes to the user's quota.

Possible actions are available via the pop-up menu located in the last column on the listing of VM images (**Action** column):

- **Create virtual machine** - shortcut to move directly to VM creation
- **Assign to group** - assign VM image to a given group. The Image is removed from the list of private images and appears in the list of group images.
- **Edit** - modify the name and description of the image. **Show advanced options** link can be used to modify image attributes such as type of graphics card, network card or disk interface.
- **Remove** - removes permanently the image. A confirmation to perform this action is required.
- **Change to storage disk** - change of disk type from the VM image to data disk. The disk appears on the list of data disks (**VM resources -> Storage disks**). The VM image is not modified.

It is recommended to remove unused private VM images to keep the disk space within the assigned quota.

Uploadng external VM image

An external VM images can be imported to the system using the **Upload image** action located at the bottom of **Images->My images** list. It might be necessary to set the proper disk interface of the uploaded image via the **Edit** from the **Actions** pop up menu. The relevant *bus* (scsi, virtio, ide, sata) can be set after clicking on **Show advanced options** link.

1.5.2 Storage disks

New virtual disks can be created via **VM Resources -> Storage disks** panel. The creation form that is presented after clicking **Create new disk** is shown in Fig. *Disk creation*. The name, description, size and format type has to be specified.

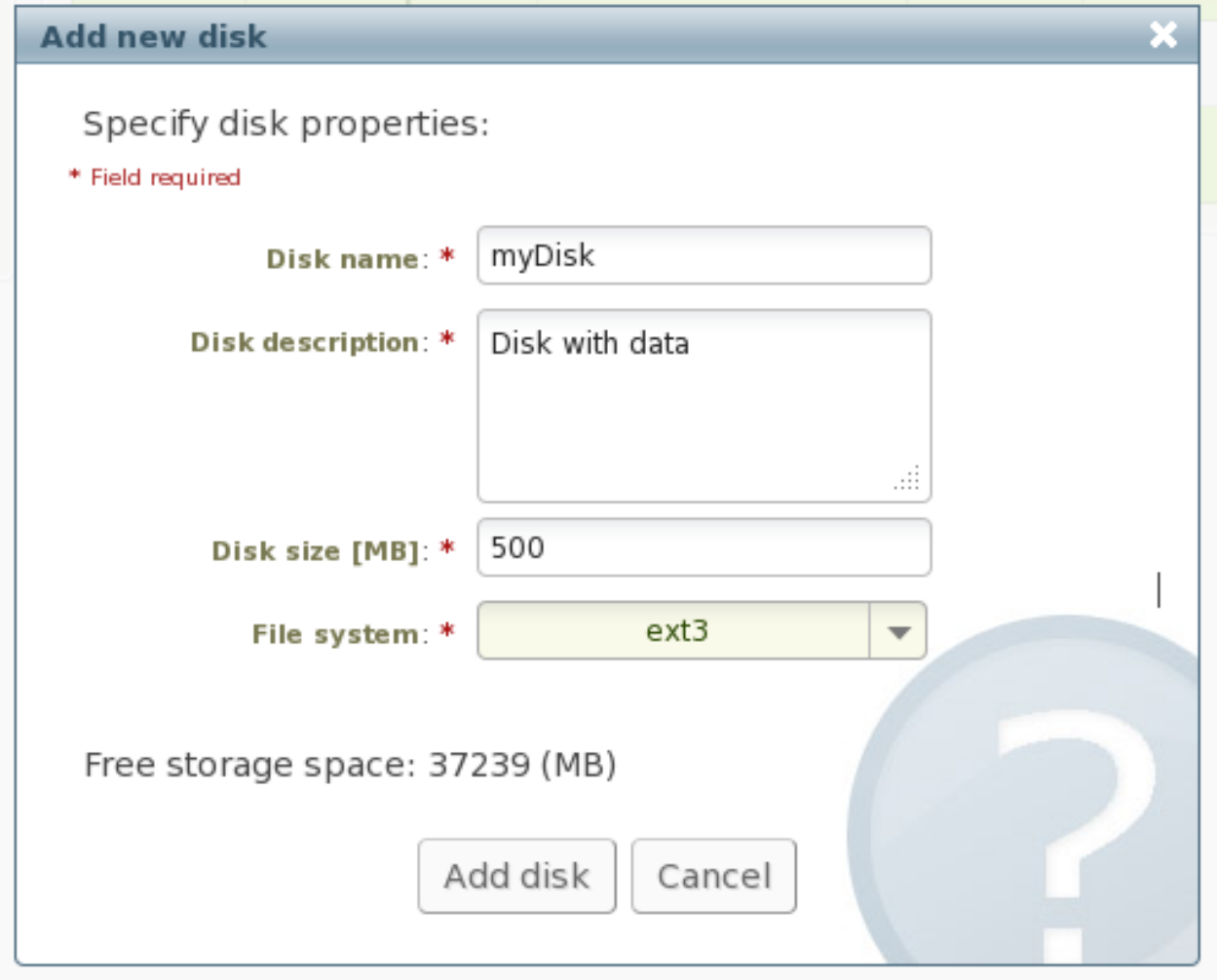


Figure 1.5: Disk creation

The external storage disk can also be downloaded via **Upload disk** action. An important disk parameter is an interface the disk is to be connected to a VM. The default is USB, having the advantage to mount or unmount the device to the VM any time. There are other possibilities accessible via the **Action->Edit** on the disk list panel: scsi, virtio, ide, sata. Their use is less convenient as they have to be attached at the VM creation phase.

In most cases the virtual disk has to be mounted manually (only USB disks are mounted automatically on some operating systems). A manual mounting can be done as follows. On Linux check first the device name assigned by the system using *fdisk -l* command. Usually the command will show two disks: one corresponding to the VM image disk that may contain a few partitions and the second disk having one partition with a typical name /dev/sdb1 (note that the name may be different).

The output of the *fdisk -l* might look as follows:

```
Disk /dev/sda: 10.7 GB, 10737418240 bytes
...
Disk identifier: 0x00076574

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1        1244     9990144   83  Linux
/dev/sda2            1244        1306      492545    5  Extended
/dev/sda5            1244        1306      492544   82  Linux swap / Solaris

Disk /dev/sdb: 5243 MB, 5243928576 bytes
...
Disk identifier: 0x00012503

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1              17       80016     5120000   83  Linux
```

where /dev/sda with three partitions corresponds to the VM image disk (the disk on which the operating system is located) and /dev/sdb with a partition /dev/sdb1 is the disk which has been attached. The latter can be mounted with a command

```
mount /dev/sdb1 /mydisk
```

where /mydisk is the name of an empty directory which has to be created first.

Note that the size of the disk might be a bit different from that declared at the creation phase.

1.6 Virtual Machine's Contextualization (CTX)

CC1 Contextualization provides mechanizms to execute specific remote commands on the Virtual Machine from the Web Interface, e.g.:

- Injecting public RSA key to enable SSH access of the private part's owner
- *Save and shutdown* action on the VM
- Remote OS user password reset on VM

CTX is required to setup computing farm from Web Interface.

1.6.1 CTX installation

To install CTX on your Virtual Machine, connect to that VM first and execute the following steps on it:

- download installation script and run it:

```
wget http://cc1.ifj.edu.pl/vmm/install.sh
bash install.sh
```

- check whether ‘‘cc1-vmm‘‘s service is present

1.7 Groups

The structure of the groups is built on top of the users database. Its primary goal is to enable the users to organize themselves in groups and share VM images by marking a private VM image as a group one. Such an image is still owned by the original user and its behavior is similar to the public images from the other group members point of view - namely they can:

- start VM from the group image,
- modify it,
- save it as a private image.

The process of organizing in groups is "light" in the sense that does not require any action by the CC1 system administrator. Every user can create a new group, becoming initially its administrator. Other users can browse groups and request to be included in a group. The request appears in the list (**Groups->Browse Groups->group**) and can be accepted by the group administrator. The group administrator can assign another user to be the administrator. The group administrator can resign and become an ordinary member of the group provided there is at least one group administrator left inside the group. The user can belong to several groups.

1.8 Quotas

Large, public cloud systems give an illusion of infinite computing resources, that would be available on demand.

In the case of private cloud systems, it has to be taken into account, that the resources are limited. That said, various quotas are introduced to control the usage of resources, in particular - to prevent a single user from consuming all system resources.

The three types of quota were implemented:

Computational quota two independent limits on the number of processor cores and size of the RAM.

Network related quota the number of public IP addresses the user can reserve.

Disk space quota the total size of disk space. This is the sum of two contributions, the size of private VM images and the size of virtual disks.

The computational and network quotas are strict. The disk space quota is a kind of soft quota that can be exceeded to ensure, that all critical operations (such as saving VM image) end successfully.

However, if the disk quota is exceeded, all user operations not related to releasing disk space are blocked. The user should reduce the usage of disk space or request that the disk quota is increased.

1.9 Farms

The functionality to create the preconfigured farm of VMs is provided to perform large scale processing. The farm can be created with any VM image with contextualization installed. The farm consists of a single head node (HN) and a specified number of worker nodes (WNs). All are created with the same VM image. There are two phases of farm creation. In the first phase the configuration is performed by the CC1 system.

The farm creation can be initiated from the **Farm->New farm** item. This is done in a similar way a single VM is created (See: *Starting Virtual Machine*). In addition, the number of WNs and their template have to be specified. The template for HN can be selected separately. The resources for the whole farm are checked to match the quota of the user and available resources on the Cluster Manager. First the HN is started and configured via contextualization. The necessary condition to proceed is to set the communication with HN via contextualization. Otherwise the farm

creation is stalled and the only possibility is to use **Destroy** action to remove the farm from the system. Once the HN is runing, the WNs are created.

When all WNs establish communication with the contextualization server the farm is being configured. The ssh key pair is generated on a HN and then propagated to WNs to allow for paswordless access from the HN. On every farm VM the /etc/hosts file is filled with names and IP numbers of the farm nodes. No central authentication is required. All users present on a base VM image appears as local users on HN and every WNs since they are created from the same VM image.

After successful configuration the farm is marked as *Running* (Fig. *Farm details*). Only *Running* status ensures proper operation of the farm.

Figure 1.6: Farm details

The second phase of configuration is performed using scripts executed on the HN from superuser account. The current implementation is the following. The HN plays the role of the NFS server and the batch system controller. All WNs share the same /home directory exported from HN to all WNs. Optionally the additional disk of HN can be NFS mounted on WNs. Currently the Slurm batch system is preinstalled on farm VM images.

The multipurpose script located in the /opt/cc1/farm directory is used for configuration:

```
/opt/cc1/farm/farm_configure.sh configure
```

is used to start the Slurm batch system and to mount /home directory of HN on all WNs via NFS. Note the *configure* parameter passed to the script. The *sinfo* command of the Slurm batch system can be used for inspection. The typical output of the command should be seen:

```
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
infini*      up   infinite      3   idle farm-901-WN[0-2]
```

The information how to use the batch system can be found in the Slurm User Guide.

Optionally an additional disk can be mounted on HN as described in Sec. *Storage disks*, say, in the directory /soft. This directory can be farm wide mounted (NFS) using the command:

```
/opt/cc1/farm/farm_configure.sh mount /soft
```

The functionality to unmount the directory is also provided:

```
/opt/cc1/farm/farm_configure.sh unmount /soft
```

When the data processing is finished the farm can be destroyed (**Destroy** action) of saved (**Save and shutdown** action). In the case **Destroy** both HN and all WNs will be destroyed. The **Save and shutdown** action results in destroying the WNs while the HN will be saved as in the case of single VM instance. All modifications made on HN will be available for the creation of farm in the future.

1.10 Security requirements

Each user is obliged to pay attention to the computer security. As the owner of the VM with the superuser privileges, each user has to be aware about the security threats. In particular the basic rules described widely in popular articles have to be obeyed.

Regular system updates and setting strong passwords are the two examples. One has to be aware that a compromised machine increases significantly the risk for other users inside the local network.