OpenNebula Tutorial

2013.07.01

Márk Gergely
Laboratory of Parallel and Distributed Systems
gergely.mark@sztaki.mta.hu
Today you will learn about how to ...

- Sign in into the Sunstone Cloud Web Frontend
- Start a Virtual Machine
- Access the Virtual Machine
- Manage the Virtual Machine
- Save a non-persistent Virtual Machine Image
- Upload new images into the cloud
- Create new templates
- Use the EC2 interface
- Use the S3 interface
The Cloud

• The hands-on will be held on the production cloud of the laboratory, so please handle with care
• The architecture of the cloud is really similar to the SZTAKI Cloud, but with less redundancy and capacity
• We will use OpenNebula 3.8 with a few SZTAKI customization
• The Cloud contains:
  • 11 Nodes
  • 184 CPU Cores, 468 GB Memory
  • 33TB of high performance iSCSI storage
• You can interact with the cloud through various interfaces, at first we will use the web frontend.
The address of the Web Frontend is: https://cfe2.lpds.sztaki.hu

Please open it with a Chrome browser (or Safari)

We use our institute’s Authentication and Authorization Infrastructure to authenticate our users. This is an SSO Service.

You have to select an identity provider: choose the “MTA SZTAKI partnerek”
Sign In

• Fill in your username and password, you can find these informations on your badge’s backside
  • Your username is: summerschool#, where # is a number between 1 and 25
  • Your password is an 8 character long case-sensitive string consists of lower and uppercase letters and numbers.
Main screen

After signing in, you will see this screen below.

- You can see your resource usage and your quotas on the middle of the screen
- There is a menu on the left side of the screen
Main screen

• For a cloud user, the most important part of the menu is the “Virtual Resources”.

• Here you can list and manage the 3 type of virtual resources:
  • Virtual Machines
  • Templates
  • Images

List of the corresponding virtual resource
HANDS-ON

Logging in into the cloud
Start a Virtual Machine

In the next section, we will start a virtual machine in the LPDS cloud.

- This is like having a new physical machine with pre-installed OS/applications in just minutes.
- This Virtual Machine (VM) will be used by the subsequent WS-PGRADE / gUSE / CloudBroker tutorials.
- To start a VM, you have to instantiate a template.
- The template describes the Virtual Machine’s preferences
  - Number of CPU Cores
  - Memory
  - Networks addresses
  - Hard disks (images)
Start a Virtual Machine

- In order to start a Virtual Machine, you have to click on the ‘Virtual Machines’ menu.
- Here you can see a list of your existing virtual machines.
- Click the ‘+New’ button.
Start a Virtual Machine

Select the “[BASE] SS2013 - CloudBroker Apps Base” template, choose a name for the new VM or leave it blank (it defaults to the template name), and click the “Create” button.
Start a Virtual Machine

Your new VM should have appeared in the list, and the cloud system started to instantiate it.

- The first state of the new VM is the **PENDING**, it means that your VM is waiting for the Scheduler to deploy it to a node.
- Next state is **PROLOG**, it means that the storage server copies the VM Images and sets up the iSCSI connections.
- The **BOOT** state means that the Middleware instructs the node’s hypervisor to start the VM.
- The **RUNNING** state means that the VM is ready to be used.

<table>
<thead>
<tr>
<th>ID</th>
<th>Owner</th>
<th>Group</th>
<th>Name</th>
<th>Status</th>
<th>Host</th>
<th>IPs</th>
<th>VNC Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>11366</td>
<td><a href="mailto:summerschool24@partners.sztaki.hu">summerschool24@partners.sztaki.hu</a></td>
<td>LPDS-Summerschool</td>
<td>[BASE] SS2013 - CloudBroker Apps Base</td>
<td>PROLOG</td>
<td>pluto.intra</td>
<td>192.168.153.39</td>
<td>VNC</td>
</tr>
</tbody>
</table>
Access a Virtual Machine

Once your VM is running, you have 2 ways to access it:

- VNC within your browser
- Basic network access
Access the Virtual Machine - VNC

You can start the VNC access with the “VNC” button in the VM’s row. The biggest benefit of the VNC access is that it is independent from the VM. You can even observe the boot process of the VM.
Access the Virtual Machine - VNC

- The VNC access is provided by the virtualization hypervisor, therefore it is stable and trustworthy.
- We do not recommend to use the Web based VNC access for rich graphical screens, because it might be slow.
- For real remote desktop usage, you should use TeamViewer, LogMeIn, RDP, or a VNC server installed inside the Virtual Machine.
- You have to use a WebKit based web browser (Chrome or Safari), because this access method requires advanced browser features like Web Canvas, Websocket, etc...
Access the Virtual Machine - Network

The network access is available after a successful boot process, it is highly recommended to enable the SSH or other console access in the VM Image.
Access the Virtual Machine - Network

- We normally use private NAT IP addresses.
- These addresses are reachable from the whole laboratory network, and through our terminal server.
- Other cloud providers may assign public IP addresses.
- It is technically possible to bind public IP addresses to the VMs, however our institute have a strict policy for using public addresses, therefore we need a good reason to use one.
- The username is "root" and the password is "lpds" for the previously started VM.
HANDS-ON

Start and access a virtual machine
Manage the Virtual Machine

You have many options to manage your VM. You can shut it down, stop it, suspend it, reboot or reset it, just like with a real physical machine.

You have to choose a virtual machine before you issue these commands (You do this with the checkbox in the first column).
Manage the Virtual Machine

Whenever you want to shut down your VM, it is important to do it from the Cloud frontend. The Cloud middleware will set the VM state to ERROR if it does not get a notification about the shutdown. This is dangerous if you want to save the disk contents into another image.
Manage the Virtual Machine

Unresponsive or stuck VMs can be deleted with the “Delete” button. It is like you unplug your computer. Pay attention, if your image is non-persistent, then your data will be lost.
Manage the Virtual Machine

Access Control Levels (ACLs) can be adjusted with the Update Properties button. You can share your VM with your group, or with every user in the cloud.
Save a non-persistent VM Image

- Modifications will vanish if you stop a VM with non-persistent disk image. (And we mostly use non-persistent disk images.)
- However you can persist these images with the save-as function.
- The Save-As function copies the contents of the VM into a new disk image.
- With the help of the save-as function, it is possible to develop VM images in an iterative manner.
- These images may contain not just the pre-installed OS, but other pre-configured applications or services.
- It is easy to share the created disk image with your group or with every user in the cloud.
Save a non-persistent VM Image

- Let’s install an AutoDock software pack into the previously started VM
  - SSH into your machine:
    
    ```
    $ ssh root@192.168.15#
    ```
  - Issue the following command:
    
    ```
    $ ./install.sh
    ```
- The install.sh script will install a few packages, wrapper scripts and tools for the tutorial next day
- So we upgraded our base Debian 7.0 image with an AutoDock software pack, so let’s call this image “[APP] CloudBroker AutoDock Worker”.

```
Last login: Thu Jun 29 11:35:32 on ttys005
becco@solarian$ ssh root@192.168.153.73
Warning: Permanently added '192.168.153.73' (RSA) to the list of known hosts.
Linux debian 3.2.0-4-amd64 #1 SMP Debian 3.2.46-1 x86_64

The programs included with the Debian GNU/Linux system are free software; the
exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Jun 24 13:35:42 2013 from jerry.lpds.sztaki.hu
becco@debian$ ./install.sh
Reading package lists...  Done
Building dependency tree
```
Save a non-persistent VM Image

To save a non-persistent disk image:

• Click on the VM’s row
• Choose Disks & Hotplugging
• Choose a name ([APP] CloudBroker AutoDock Worker) and click Save
Save a non-persistent VM Image

After clicking the “Save” button, the storage server allocates the required disk space for the image. You have to shut down the VM in order to the copy actually happen.

The EPILOG state represents that the VM Image is actually being saved into a new image.
HANDS-ON

Manage the VMs and save non-persistent disk images
Upload new images into the cloud

If you cannot find an image which is suitable for your needs, then you can upload your own disk image. There are several methods for uploading images:

- If you want to upload image from your local machine:
  - Through your web browser (We recommend WebKit based browsers (Chrome, Safari), Firefox is fine for 4GB or smaller images)
  - Through the EC2 interface

- From the internet:
  - You can instruct the OpenNebula to download images from the internet (http / https / ftp)
  - Using the OpenNebula Marketplace (We will use this feature.)

Because of the iSCSI storage backend, our system supports the raw image format. If you ever need to convert a disk image, then the easiest way is to use the `qemu-img` linux software pack.
Upload new images into the cloud

In order to upload a new image from the Marketplace, click on the “Marketplace” menu, Select the image called “ttylinux - kvm” in raw format and x86_64 architecture, and click on the “import to local infrastructure button”
Upload new images into the cloud

This popup box is the same like the one when you want to upload a new image from your computer, it just prefills a few parameters.

• Choose a name for the new image (defaults to the name from the marketplace)

• Datastore: iscsi-izabel (the name of the storage computer)

• Type:
  • OS
  • CDROM
  • Datastore
Upload new images into the cloud

- **Persistent:** If a disk image is not persistent, then every modifications you made on the image will be destroyed when you shut down your VM, however you can start multiple instances from the same Image.

- **Device prefix:** leave it blank, unless you know that the operating system supports the VirtIO paravirtualized drivers, then it should be “vd”
Upload new images into the cloud

Image Location:
- Provide a path: choose this if you want to upload an image from a remote machine (http/https/ftp). This field is already filled with the marketplace link.
- Upload: choose this if you want to upload an image from your local machine
- Create an empty datablock: Choose this if you have to create an empty image with a filesystem.
HANDS-ON

Upload images into the cloud
Create new templates

A disk image is just the hard drive of a virtual machine, does not describe the CPU, amount of memory, network addresses and other interfaces. OpenNebula uses Templates to describe a Virtual Machine. To create a new template, click on the “Templates” menu, and click on the “+New” button.
Create new templates

Let’s create a template for the previously downloaded ttylinux image:

- Choose a template name ([BASE] ttylinux)
- How much memory you need (64MB!)
- CPU should be 0.001 (OpenNebula scheduler information)
- VCPU is the desired number of CPU cores
- Other fields are fine with the default value.
Create new templates

- Select the “Image” radio button, after that, select the desired disk image from the dropdown list, and click the Add button.

- Select the desired network for the VM (should be cloud-local). If the guest OS supports the VirtIO paravirtualized network driver, then the model field should be filled with “virtio”
Create new templates

- For seamless Web VNC experience, these informations should be filled into the inputs and Graphics section.

The template is ready to be used, click the "Create" button on the bottom of the pop-up window.

Try to start a VM from the template the way we learned before.
HANSDS-ON

Create new template
Using the EC2 interface

- The EC2 API is the de-facto standard of the cloud interfaces.
- OpenNebula provides EC2 compatible interface.
- It implements a subset of the EC2 commands.

- We prepared EC2 client toolset on the thin-client machine in front of you
- Or you can download and install it on your own machine
  - URL: http://aws.amazon.com/developertools/Amazon-EC2/351
  - It requires Java
Using the EC2 interface

You need 3 parameters to be able to reach the cloud through the EC2 interface:

- **EC2 endpoint:** http://cfe2.lpds.sztaki.hu:4567/
- **EC2 access key:** summerschool#@partners.sztaki.hu
- **EC2 secret key:** your password hashed with sha1 algorithm
  
  **Linux:** $ echo -n "YOURPW" | sha1sum  
  **Mac:** $ echo -n "YOURPW" | shasum  
  **Website:** http://www.sha1.cz

We pre-generated everyone’s hash here: goo.gl/aznhQ
Using the EC2 interface

Let’s try to list the existing virtual machines and images:

• Set the credentials (once in every console session):
  
  $ export EC2_URL=http://cfe2.lpds.sztaki.hu:4567
  $ export AWS_ACCESS_KEY=summerschool#@partners.sztaki.hu
  $ export AWS_SECRET_KEY=sha1hashedversionofyourpassword

• Try to list the images:
  
  $ ec2-describe-images

• Try to list your virtual machines:
  
  $ ec2-describe-instances
Using the EC2 interface

Let’s start a Virtual Machine from the EC2 interface.

This VM will play an important role in the subsequent tutorials.

You cannot instantiate templates from the EC2 interface. Instead of that, you instantiate VM image with an already prepared instance type.

We support 5 instance types:

- m1.micro: 1 CPU, 256MB memory
- m1.small: 1 CPU, 512MB memory
- m1.medium: 1 CPU, 1024MB memory
- m1.large: 2 CPU, 2048MB memory
- m1.xlarge: 4 CPU, 4096MB memory
Using the EC2 interface

- Start a VM with `m1.large` instancetype and image ID `ami-00000737`:
  
  ```
  $ ec2-run-instances -t m1.large ami-00000737
  ```

  Check the instantiation process on the Web Front-End and also through the EC2 interface:
  
  ```
  $ ec2-describe-instances
  ```

- [https://cfe2.lpds.sztaki.hu](https://cfe2.lpds.sztaki.hu)
HANDS-ON

How to use the EC2 interface
Using the S3 interface

• The Simple Storage Service (S3) is an easy to use, very capable cloud based storage interface.

• We built an S3 compatible storage service over our IaaS using the amazing Ceph distributed storage software.

• As it is just an S3 “compatible” interface, it implements just a subset of the S3 features.

• We prepared S3 client toolset on the thin-client machine in front of you

• Or you can download and install it on your own machine:
  • URL: [http://s3tools.org/download](http://s3tools.org/download) or [http://www.dragondisk.com](http://www.dragondisk.com)
  • It requires Python
Using the S3 interface

• Ceph provides a horizontally scalable storage architecture
• Every part of the infrastructure can be easily extended or shrinked to fit the load.
• It is fully redundant, the uploaded data will be replicated on N nodes
Using the S3 interface

You need 3 parameters to be able to reach the S3 interface:

- S3 endpoint: http://s3.lpds.sztaki.hu/
- S3 access key: summerschool#
- S3 secret key: your password hashed with sha1 algorithm
  
  Linux: $ echo -n "YOURPW" | sha1sum  
  Mac: $ echo -n "YOURPW" | shasum  

Website: http://www.sha1.cz
Using the S3 interface

Configure the s3cmd command:

```
$ s3cmd --configure
```

This command will ask for your access key and shared key, other parameters are fine with the default value.

It will fail at the end of the configuration because it tries to connect to the Amazon S3 interface. To change this behaviour:

```
$ sed -i 's/amazonaws\.com/lpds\..sztaki\..hu/g' ~/.s3cfg
```
Using the S3 interface

We want to store a file on S3. As files are stored in buckets, let’s create one first. **Buckets are globally unique names:**

```bash
$ s3cmd mb S3://summerschool# where # is your ID
```

Create a File and upload it to the S3:

```bash
$ echo “testfilecontent” > testfile.txt
$ s3cmd --acl-public put testfile.txt S3://summerschool#/ 
```

You can access this file through S3 or through the Web:

```bash
$ wget https://s3.lpds.sztaki.hu/summerschool#/testfile.txt
```
HANDS-ON

How to use the S3 interface
Thank you for your attention!