Lab 11
Wrap up final projects

Hands-on Unix system administration DeCal

2012-11-19

Important note This lab is not graded and will not be collected. It is for your reference only. It has some *basic* instructions for using the DeCal Cloud—which is part of the final project (which is in turn a *required* part of the course).

The DeCal Cloud

For more information about virtualization and the cloud, refer to Lecture 9. In a nutshell, for our purposes, *virtualization* refers to multiple virtual machines (VMs) on one physical computer, and the *cloud* (or more specifically, "Infrastructure as a Service") refers to virtualization on a large scale with networking, block storage. . . etc.

The DeCal Cloud (hereafter, "Cloud" with a capital "C") is our name for a homebrew cloud service, as opposed to a commercial cloud service like Amazon Web Services (AWS), which will allow you to provision multiple VM *instances* on the same virtual 10.0.x.0/24 network subnet. To make matters complicated, the cloud service is itself a VM (this is called nested virtualization), but don't let *Inception* worry you.

Accessing the Cloud

First thing's first, we need to access the Cloud system. Cloud accounts were created for you.

- User Name: yourOCFusername
- Password: see contents of `/opt/ocf/decal/cloud/yourOCFusername` from `tsunami.ocf.berkeley.edu` (aka `ssh.ocf.berkeley.edu`)

There are multiple interfaces to the Cloud:

- web interface: `https://coupdetat.ocf.berkeley.edu/`
- command-line interface: `nova`
- AWS-compatible interface (we won't go over this, but it's there)

Provisioning an instance

"Provisioning an instance" is essentially[1] a fancy term for creating and booting a VM. Provisioning an instance has multiple components:

- creating the volume (block storage) — this is equivalent to buying (or more accurately, renting?) and connecting a hard drive inside a physical computer
- setting up the installer image — this is equivalent to putting the OS installer CD inside a physical computer's CD drive
- allowing networking access through security groups and floating (aka elastic) IP addresses — this is roughly analogous to configuring the router connected to a physical computer's Ethernet port

[1] Provisioning also includes configuring the Unix system, but that is beyond the scope of these instructions.

Using the web interface

There are many correct ways to do these tasks (don't feel constrained by the instructions that follow), but we'll assume that you're not (yet!) familiar with the command-line interface and want to use the web interface where possible. A bit tedious, but perhaps easier to start off with.

After logging in to the web interface, under "Volumes", create a volume with a reasonable name and a size (if you're not sure, use 5 GB unless you know you will be using more). Then you'll need to upload the install image (typically an ISO image of the installer CD). We'll assume that you want to use Ubuntu, which has already been uploaded, but if you want to explore further, feel free to try uploading something else, under "Images & Snapshots".

Now lets create a security group. A security group is a set of firewall rules that can be applied to a VM (in our Cloud, we only block incoming traffic, but other clouds may block outgoing traffic as well). Lets create one, under "Access & Security". To allow SSH access, you'll need to allow TCP port 22, but don't worry too much about rules because you can edit rules on the fly later.

Now let's actually launch the instance under "Instances". "Instance Source" is "Image" and "Image" is "ubuntu_quantal_amd64_iso" unless you uploaded something else. "Flavor" is one of our predetermined resource allocation sizes (number of virtual CPUs and amount of RAM) and will count against your quota—if you choose smaller flavors you'll be able to run more VMs. As a last step, you'll want to check the security group you created under the "Access & Security" tab. Now launch!

After the instance is launched, you'll want to attach your volume to it under "Volumes". You can use `/dev/vdc` as the device ID although in most cases (such as with the Ubuntu) it shouldn't matter. Now go back to the instance you launched, and access the VNC console to proceed through the install. Nifty? We think so.

Finally, when you want to actually access your VM through something better than a framebuffer web interface, and assuming you've added the appropriate security group rules, you can associate your VM with a public IP address, under "Access & Security". Once you have a VM that you can access over the Internet, you can access the other VMs through it by in turn connecting from it to the listed private IP address (next to the VM name under "Instances") on the 10.0.x.0/24 subnet.

Using the command-line interface

The command-line interface offers many more features than the web interface. It can also be accessed by a script.

In the web interface, under "Settings" (top-right corner) -> "OpenStack API", you can download an RC file which you can source in a terminal session (e.g., over SSH) *on your VM*[2]. Assuming you're running Ubuntu on the VM, you'll need to install the `python-novaclient` package (e.g., `sudo apt-get install python-novaclient`).

These two commands should be very *help*ful:

- `nova help`
- `nova help COMMAND`

And beyond

Feeling feisty? Try snapshotting your VM and creating a new VM from that, or seeing what the command-line interface has to offer.

[2] Due to limitations with the Cloud API, your Cloud password is being transmitted unencrypted, unlike through the web interface (which is HTTPS/SSL). This is not an issue when the Cloud API is accessed from the VM (your VM is running locally on the Cloud server).

Make yourself at home

Now that your project group has its own system, you want to create your own user account on it. On Ubuntu systems, you should be able to run the command `sudo adduser mynewusername` (assuming the account has sudo access, it typically will if it's the first account created during installation, otherwise you will have access to the root account and can run the same command without sudo) to create your account after following the prompts.

Making yourself an administrator

As an alternative to logging in as root, many Unix-like systems advocate the use of sudo, a setuid program that allows some users to run programs with the security privileges of another user (often root). Access to sudo is controlled by the `/etc/sudoers` file.

In many Unix systems, members of a certain group (for example, group sudo on Ubuntu) are allowed complete access to sudo. Users in the sudo group therefore have complete root access by means of sudo, and can impersonate any other user—this is not to be taken lightly.

Add yourself to the sudo group with the command `usermod -a -G sudo mynewusername`. The `-a` option appends the sudo group to your list of groups, and the `-G` option adds sudo as a secondary group.

Do logs not logarithms

The `rsyslog` daemon stores messages generated by the system in `/var/log`. By default in Ubuntu, a copy of most messages is also included in `/var/log/syslog`. Take a look at `/var/log/syslog`.

Authentication logging (for example, unsuccessful and successful login attempts) is by default in Ubuntu stored in `/var/log/auth.log`.

Snooping with publicly-accessible information

Sometimes system logs make you suspicious of a user, but because of privacy concerns, you don't want to look at private information. So look at publicly-accessible information instead.

One of the facilitators wrote a script called `check` to aggregate some publicly-accessible information on a given OCF user. SSH into `tsunami.ocf.berkeley.edu`, and run `check` on one of your project group partners. See `check --help` for syntax.

Bonus: look in CalNet

CalNet is an LDAP directory with both private and publicly-accessible information. Almost a year ago, OCF users started being associated with their CalNet UID numbers (retroactively to the mid-1990's), a unique identifier in CalNet. This allows online password resets[3] to be authenticated using CalNet.

Where available, `check` will report the associated CalNet UID number. You can search the CalNet directory for publicly-accessible attributes in that entry.

```
ldapsearch -x -h ldap.berkeley.edu -b dc=berkeley,dc=edu calnet_uid
```

If you got this far, thanks for being highly devoted: `http://xkcd.com/705/`

- Enjoy a cookie: `http://is.gd/JRxl5A`

[3] `https://secure.ocf.berkeley.edu/account_tools/change_password`