Cloud Tutorial

Running Workflows on Amazon EC2 using Pegasus

This tutorial takes you step-by-step through the process of setting up a worker node to run workflows on Amazon EC2 using Pegasus. We assume you are already familiar with Pegasus and have a workflow that already runs on a local cluster or grid site. This tutorial will show you how to make that workflow run on an Amazon node.

1. Get an AWS account

You need this before you can do anything

Go to http://aws.amazon.com
You will need a credit card. They bill monthly, and charge by the hour.

2. Install Condor on your submit host

You need to have a host outside the cloud running a Condor central manager. We will refer to this host as your "submit host". We assume that you have a submit host already if you are running Pegasus.

First you need to install Condor. That is a bit involved, so we will skip most of it. Go to: http://cs.wisc.edu/condor to get more information about how to install a basic condor manager.

Once you have a basic manager working we need to modify the configuration a bit.

Edit your condor_config file and append to ALLOW_WRITE:

```
ALLOW_WRITE = <what it was before>, *.compute-1.amazonaws.com
```

Now edit your condor_config.local and add:

```
HIGHPORT = 41000
LOWPORT = 40000
UPDATE_COLLECTOR_WITH_TCP = True
COLLECTOR_SOCKET_CACHE_SIZE = 1000
```

Finally, restart Condor

VERY IMPORTANT: The firewall on the submit host should be configured so that anything from *.compute-1.amazonaws.com can connect to port 9618, and ports 40000-41000. These ports are used by Condor.

3. Log into the Amazon Management Console

This is a web application that lets you manage cloud resources.

We will refer to this webapp as the "console", and we will refer to the links on the left side of the console as "areas".

Go to: http://console.aws.amazon.com

Click "Sign in to the AWS console"

Change the region on the upper-left side of the console to "US East".

IMPORTANT: When you use this make sure you stick to one Region (US East or US West). Most things in Amazon don't work across regions.

VERY IMPORTANT: For this tutorial please use region "US East". Or you won't be able to find our public VM image.

4. From the console create a keypair

These are the credentials you use to log into worker nodes.

Go to the "Key Pairs" area in the console
Click "Create Key Pair"
Call it "ec2-keypair" and click OK.

It should popup a download box. Save the file. We will refer to this file as KEYPAIR.

5. From the console create a security group

This is how you authorize machines outside the cloud to access your nodes.

We will assume your submit host is "host.example.com", and that it has an IP of "192.168.1.1". The security group we create here will give "host.example.com" unrestricted access to your nodes.

Go to the "Security Groups" area in the console
Call your new group "host.example.com", add a description, and create the group

Click on the group and add three entries:

Method	Protocol	From Port	To Port	Source (IP or Group)
All	tcp	1	65535	192.168.1.1/32
All	udp	1	65535	192.168.1.1/32
All	icmp	-1	-1	192.168.1.1/32

Note that those are CIDR addresses, so don't forget the /32. Also, don't forget to replace 192.168.1.1 with your submit host IP.

6. Launch the Pegasus public image

This is how you launch a virtual machine (or virtual cluster).

We are going to use a pre-configured image developed specifically for Pegasus. It contains Pegasus, Condor, and Globus.

Go to the "AMIs" area in the console.
We are going to launch ami-858741ec. Filter by "Public Images" and "CentOS" using the drop-downs, type "ami-858741ec" into the text box and hit 'Refresh'. It may take a few seconds to give you a list.
Select the one called "405596411149/centos-5.6-x86_64-pegasus-cloud-tutorial-2" and click "Launch".
A launch wizard will pop up.
Select the number of instances (1), and instance type (m1.large), then "Continue".
On the "Advanced Instance Options" page add the following to "User Data" and hit "Continue" (note: host.example.com should be replaced with your submit host):

```
CONDOR_HOST = host.example.com
```

Next, on the tags page, enter a value, any value, for the Name tag and hit "Continue"
Next, select the keypair you created earlier, and "Continue"
Next, select the security group you created earlier and "Continue".
On the last page click "Launch"

VERY IMPORTANT: Select the security group and keypair you created earlier or else it won't work. Also, make sure you replace "host.example.com" in the User Data with your submit host.

ALSO IMPORTANT: The "User Data" is how you tell the image what to do. This will be copied directly into the Condor configuration file. You can define any extra configuration values you like, but you must specify at least CONDOR_HOST.

7. Log into the node

This is how you SSH to a node you launched.

Go to the "Instances" area in the console.

You should see the instance you just launched go from "pending" to "running". You may need to hit "Refresh" a couple times.
When it says "running" click on it and get the "Public DNS" (call it PUBLIC_DNS).

From your submit host ssh to the worker:

```
$ ssh -i /path/to/KEYPAIR root@PUBLIC_DNS
```

VERY IMPORTANT: Make sure you ssh from your submit host otherwise this won't work because the security group will not match.

8. Check your submit host

Make sure the workers showed up

On your submit host run:

```
$ condor_status
```

You should see something that looks like this:

```
Name               OpSys      Arch   State     Activity LoadAv Mem   ActvtyTime

slot1@50.16.16.204 LINUX      X86_64 Unclaimed Idle     0.080  3843  0+00:00:04
slot2@50.16.16.204 LINUX      X86_64 Unclaimed Idle     0.000  3843  0+00:00:05
                     Total Owner Claimed Unclaimed Matched Preempting Backfill

        X86_64/LINUX     2     0       0         2       0          0        0

               Total     2     0       0         2       0          0        0
```

If you don't see anything, then wait a few minutes. If you still don't see anything, then you need to debug Condor. Check the CollectorLog to see if the workers tried to connect. If it doesn't work contact: pegasus-support@isi.edu.

9. Run a test job

Make sure the workers are usable
Once the workers show up in condor_status you can test to make sure they
will run jobs.
Create a file called "vanilla.sub" on your submit host with this inside:

```
universe = vanilla
executable = /bin/hostname
transfer_executable = false
output = test_$(cluster).$(process).out
error = test_$(cluster).$(process).err
log = test_$(cluster).$(process).log
requirements = (Arch == Arch) && (OpSys == OpSys) && (Disk != 0) && (Memory != 0)
should_transfer_files = YES
when_to_transfer_output = ON_EXIT
copy_to_spool = false
notification = NEVER
queue 1
```

Submit the test job:

```
$ condor_submit vanilla.sub
```

Check on the job:

```
$ condor_q
```

After a few minutes it should run. Then check the output:

```
$ cat test_*.out
```

You should see a hostname that looks like it came from Amazon.

10. Modify the image and register a copy

This is how you create your own custom image.

At this point you can install whatever you want on the running worker node. You might want to install programs, libraries, and tools used by your workflow. If you don't want to install anything that's OK, you can complete this step without modifying the image.

In the "Instances" area of the console click on the running instance and select "Instance Actions" -> "Create Image (EBS AMI)".
Give it a name and a description and click "Create Image".
In the AMIs area of the console clear all the filters (set to "Owned by Me", "All Platforms") and hit refresh. You should see a new image pop up. After some time the state should change from "pending" to "available". You may need to hit refresh a few times.

IMPORTANT: The image could stay in "pending" status for a long time. However, if it is still pending after an hour something is wrong.

11. Shut down your instance

In the "Instances" area of the console click on the running instance and select "Instance Actions" -> "Terminate".

VERY IMPORTANT: Amazon keeps charging until the status is "terminated".

12. Configure pegasus

Add an ec2 site to your sites.xml (note: this is the old XML format, modify as needed if your application uses the new format):

```
<site handle="ec2" sysinfo="INTEL32::LINUX">
    <!-- This is where pegasus is installed in the VM -->
    <profile namespace="env" key="PEGASUS_HOME">/usr/local/pegasus/default</profile>

    <!-- Just in case you need to stage data via GridFTP -->
    <profile namespace="env" key="GLOBUS_LOCATION">/usr/local/globus/default</profile>
    <profile namespace="env" key="LD_LIBRARY_PATH">/usr/local/globus/default/lib</profile>

    <!-- Some misc. pegasus settings -->
    <profile namespace="pegasus" key="stagein.clusters">1</profile>
    <profile namespace="pegasus" key="stageout.clusters">1</profile>
    <profile namespace="pegasus" key="transfer.proxy">true</profile>

    <!-- These cause Pegasus to generate vanilla universe jobs -->
    <profile namespace="pegasus" key="style">glidein</profile>
    <profile namespace="condor" key="universe">vanilla</profile>
    <profile namespace="condor" key="requirements">(Arch==Arch)&amp;&amp;(Disk!=0)&amp;&amp;(Memory!=0)&amp;
&amp;(OpSys==OpSys)&amp;&amp;(FileSystemDomain!="")</profile>
    <profile namespace="condor" key="rank">SlotID</profile>

    <!-- These are not actually needed, but they are required by the site catalog format -->
    <lrc url="rls://example.com"/>
    <gridftp url="file://" storage="" major="2" minor="4" patch="0"/>
    <jobmanager universe="vanilla" url="example.com/jobmanager-pbs" major="2" minor="4" patch="3"/>
    <jobmanager universe="transfer" url="example.com/jobmanager-fork" major="2" minor="4" patch="3"/>

    <!-- Where the data will be stored on the worker node -->
    <workdirectory>/mnt</workdirectory>
</site>
```

In your pegasus.properties file, make sure you disable thirdparty transfer mode:

```
# Comment-out the next line to run on site "ec2"
#pegasus.transfer.*.thirdparty.sites=*
```

If you installed your application code in the image, then modify your Transformation Catalog to include the new entries. (Tip: Make sure the sysinfo of your "ec2" site matches the new transformations you add to the TC)

13. Plan your workflow

Prepare your workflow to run on EC2.

We assume you know how to do this already. Use "ec2" as the target site.

If you run into any problems during planning, debug them before moving on to the next step.

If you have problems contact: pegasus-support@isi.edu

14. Launch a worker node

You will do basically the same thing you did to launch the first worker.

Instead of using the Pegasus image, use the new image you created earlier.

In the "AMIs" area select your new image and click "Launch".

Select m1.large.

Set "User Data" to:

```
CONDOR_HOST = host.example.com
```

VERY IMPORTANT: Don't just copy-paste the above, you need to replace "host.example.com" with the actual DNS name of your submit host.

Choose your keypair and security group as before and launch the node.

Wait until you see the workers show up in condor_status before proceeding. You may want to run your vanilla.sub test job again to make sure they work.

15. Submit your workflow

At this point you should submit your workflow.

If you have problems contact: pegasus-support@isi.edu

VERY IMPORTANT: You are virtually guaranteed to have problems at this point because of the large number of possible configurations required by your application. Please contact us and we will help.

16. Clean Up

Hopefully your workflow will run to completion. When you are finished make sure you terminate any running instances in the "Instances" area of the console.

17. Next steps

This tutorial only shows you how to set up a single worker node. In order to scale-up your workflows you will need to either a) set up a shared file system such as NFS, or b) configure Pegasus to use Condor file transfers for each job.