

A guide to debugging failed instances on the NeCTAR cloud

You've fired up an instance on the NeCTAR cloud. And it doesn't seem to be working. What steps can you take to work out where the problem lies?

There are three places to look for errors:

- the cloud infrastructure
- the instance itself
- the applications on the instance.

The following process has been found useful when trying to track down errors

1. stay calm!
2. look for clues
3. use google
4. form a theory as to the cause of the error
5. test your theory
6. rinse, lather and repeat from step 1 if the theory turns out to be false.

This guide is not comprehensive, but offers up tips on some of the problems that have been encountered on the NeCTAR cloud.

What is the status of the NeCTAR cloud?

It doesn't happen often. But occasionally the NeCTAR cloud does have issues. So before leaping into problem solving mode, it might be worthwhile to have a look at the status of the NeCTAR cloud.

You can see the status of the NeCTAR cloud as a whole by looking at the

<http://status.rc.nectar.org.au/> site.

If that shows that the cloud as a whole seems to be fine, have a look at the known issues on the support site: <https://support.nectar.org.au/support/solutions/folders/6000206494>

Finally, at the command line, the following call will give you an idea as to the status of the individual availability zones:

```
$ nova availability-zone-list
+-----+-----+
| Name          | Status    |
+-----+-----+
| monash        | available |
| monash-test   | available |
| QRIScloud     | available |
| melbourne-np  | available |
| monash-02     | available |
| tasmania      | available |
```

```
| tasmania-s      | available |
| nova           | available |
| melbourne      | available |
| NCI            | available |
| qld            | available |
| melbourne-gh2  | available |
| sa             | available |
| monash-01      | available |
+-----+-----+
```

The cloud infrastructure

If everything checks out the next place to turn to is the NeCTAR infrastructure.

Here it helps to have some understanding of the NeCTAR architecture. The NeCTAR cloud is based largely on a message passing model. You make a request, the request is put on a queue, and at some later time a worker process takes the message off of the queue and starts to perform the requested activity. That worker might in turn put other messages onto a queue, asking other worker processes to perform activities on your behalf. Eventually, when they have all finished their tasks your original request is, hopefully, fulfilled.

If the NeCTAR infrastructure is under stress for whatever reason, workers will not be pulling messages off of the queues as rapidly as they should. Thus under these circumstances it might take a very long time for your instance to become fully active.

Also, given this message passing architecture, if something goes wrong further down the message pipeline you might not associate the error you receive with your initial request.

What does OpenStack tell you?

So the first question to ask here is: what is the state of your instance as reported by the OpenStack tools?

The quickest way of doing this is to simply look at the instance you have launched on the instances tab of the dashboard (<https://dashboard.rc.nectar.org.au/project/instances/>). If all has gone well and it is running the instance status should read “**Active**”.

Similarly, if using the command line tools:

```
$ nova list
```

should show a status of “**ACTIVE**” against your instance if all has gone well.

Here are a list of some of the instance statuses that you might encounter at this point:

| Status | Meaning |
|----------|--|
| ACTIVE | The instance is up and running |
| BUILDING | The instance is still being built |
| ERROR | There was a problem. Your instance is not running, and will will not run. |
| UNKNOWN | There was a severe problem. Your instance is not running, and will will not run. |

A complete list of possible statuses can be found here:

<http://developer.openstack.org/api-ref-compute-v2.html#listServers>.

If your instance is based on an image that has booted successfully before, then consistently arriving at a launch state of ERROR or UNKNOWN will probably require a support request to NeCTAR (<http://cloud.nectar.org.au/contact/>).

Can you access your instance?

NeCTAR's notes on accessing your instance can be found here:

<https://support.rc.nectar.org.au/docs/accessing-your-instances>

If not, the following are some of the questions to be examined.

Are your Security Group settings correct?

If the instance is reported as being in the **ACTIVE** state, but you can't reach it via the network, the next puzzle piece to examine is the Security Group settings that you have in place. Are they correct?

A security group is the set of inbound IP filter rules that control access to your instance via the network. If you do not select a security group when you launch your instance, a default security group that will deny all incoming traffic will be applied.

NeCTAR's notes on security groups can be found here:

<https://support.rc.nectar.org.au/docs/security-groups>

You have to confirm that the ports exposed by the security group match the ports that you are trying to connect to.

Are you connecting to the right server?

Confirm that the IP address that you are trying to connect to is in fact the IP address of your server. It is very easy to mistype an IP address, and occasionally you will find another server at the mistyped IP address. One with the port that you are trying to connect to open...

Did you inject a key, and was it the right one?

If you are trying to ssh into your instance, and you can't connect, did you launch the instance with a key, and if so and are you using the matching key pair to try and access the instance?

A quick test to see if you used a key pair is to simply look at the key pair column on the instances tab of the dashboard <https://dashboard.rc.nectar.org.au/project/instances/>

An indicator for both of these scenarios is being requested to provide a password when you try to ssh into the instance.

NeCTAR's notes on keypairs can be found here:

<https://support.rc.nectar.org.au/docs/launching-123>

Are you using the correct user name?

Different images have different default usernames to be used when logging in. Are you using the correct one?

If using a standard NeCTAR image, the NeCTAR image catalogue shows you the appropriate username to use for the different instance types. It can be found at:

https://wiki.rc.nectar.org.au/wiki/Image_Catalog.

Can you see external hosts from inside the instance?

If your security groups are fine, and you can ssh into the instance, and from inside the instance you can successfully ping an external host such as google.com, it is a safe bet that you don't have a network problem.

The instance itself**Is it based on a correctly configured image?**

If your instance is based on an image that you built yourself, or is one that is not an official NeCTAR image, then be aware that OpenStack expects certain software to be installed on the image: and that OpenStack interacts with that software during the boot process. If that software is not present your image may not boot, or if it does, may it not be accessible from the outside world. More on the creation of images for the OpenStack cloud can be found here:

http://docs.openstack.org/image-guide/content/ch_introduction.html

If launched successfully, is there a log file?

You can view the log file for an instance by either going to the 'Log' tab for the instance on the dashboard or if using the command line tools by issuing the command

```
$ nova console-log <instance id>
```

If the log is either empty or displays only a single character (such as ‘?’) then Nova was unable to retrieve any output from the instance. Nova is configured to retrieve the log output via a serial console. So at this point there are two possibilities:

- although the instance launched successfully, it had some sort of internal issue when booting that prevented the log from being created. An image with a corrupted file system can cause this issue.
- the instance was not configured to send log output to the serial console. Again, this outcome is most likely if your instance is based on an image that you built yourself, or is one that is not an official NeCTAR image.

If there is a log file, what does it say?

The log file should hopefully give you a better idea as to what has happened, and give you clues as to where the instance broke on booting. For example, if you find the words “Kernel panic” in the log file then the operating system encountered an error from which it could not recover. The Internet is your friend at this point!

Has the instance drive been toggled into read only mode?

Sometimes operations in the data center can lead to the instance’s drive being toggled into read only mode. Ordinarily a message to this effect will be seen in the log file. The fix is simple: reboot the instance, and all should be sweetness and light.

Can you run the image off of the cloud?

PS: I’ve done this, but I’m not sure if this is good advice...

If the image is not booting successfully, you can, if the image is in an appropriate format, download it to your desktop and convert then boot it using VirtualBox (<https://www.virtualbox.org/>). It’s a long shot, but sometimes launching an image like this can help give you clues as to what might be going wrong.

To download the image to your local machine:

```
$ glance image-download 7832f771-1a32-4bb8-82f2-ce407b46c5f0 --file broken.img
```

where `7832f771-1a32-4bb8-82f2-ce407b46c5f0` is the ID of the image that you want to download, and `broken.img` is the name you are giving it on your local machine. Make sure that you know the image type.

Notes on converting between image types can be found here:

http://docs.openstack.org/image-guide/content/ch_converting.html.

VirtualBox itself can convert a raw image to one that it can run:

```
VBoxManage convertfromraw broken.img broken.vdi --format vdi
```

Then fire up VirtualBox. Create a new machine, choosing the operating system type of your downloaded image. Select 'use an existing hard drive' and select your newly converted image. Then boot it and see if you can determine where the failure happens.

Did you use Heat?

If you used Heat, and can ssh into the instance, you can find the script that is placed in the `user_data` property of the template at: `/var/lib/cloud/data/cfn-userdata`

Does it look as you expect it to? Were all the variables substituted correctly?

The `cfn-user` data script can take some time to run.

You can check if its still active by issuing the following command:

```
$ top -b -n 1 | grep cfn-userdata
```

If the script is still running, a line will be returned. If it's not running, nothing will be returned. If it is still running, then the instance has not yet been completely set up, or possibly, the script has some sort of loop in it that has not terminated.

The applications on the instance

Did the instance previously work, but is now longer responding?

Can you reboot the instance? A reboot will often fix a hung instance, and indicates that there was some sort of failure on the instance itself. Once rebooted you can examine the log files to try to determine the cause of the failure.

Is there enough disk space?

One of the most common forms of failure in this case is running out of space on the primary partition.

To see how much free space you have when ssh'd into the instance:

```
$ df -h
```

To hunt down where the space is being used when ssh'd into the instance use the 'du' command. eg:

```
$ du -h --max-depth=1
8.0K    ./ssh
20K     .
```

Rebooting the instance fails, or is not an option

If you have an instance that was running successfully, but is now no longer responding, and rebooting doesn't help, Nova Rescue might be just the ticket.

When you use Nova Rescue, the following sequence of events occurs:

- The instance is shutdown
- A new instance is booted from the original instances image
- The new instance has the original instance's file system mounted as a secondary file system.
- The instance is moved to a status of RESCUE

You can then ssh into the rescue instance at the original instances address, and inspect the file system to try and work out what the issue was (running out of disk space is quite common). You can make changes to the file system of the original instance as well.

Once you understand the issue you can then return the instance back to a status of active. At this point the original instance is rebooted and will resume running again. If you don't think that it is safe to restore the instance then you can simply terminate it.

NB: Do not try to unmount any block storage instances that you might have attached to the instance whilst in rescue mode.

To put an instance into rescue mode, you have to use the Nova command line client:

```
$ nova rescue <instance id>
+-----+-----+
| Property | Value      |
+-----+-----+
| adminPass | iXA98EFoAAa7 |
+-----+-----+
```

The reported administrator password is not needed if you're original instance had a keypair injected into it.

Do not ssh into the instance right away. Wait for Nova to show that it has a status of Rescue.

```
$ nova list
```

| ID | Name | Status | Task State | Power |
|--------------------------------------|---------|--------|------------|---------|
| 598ec2ba-9d87-484d-ae3a-4661b4be3d91 | tester2 | RESCUE | - | Running |

nci=130.56.252.26

If you try to access the instance too quickly then the instance won't go into rescue mode. A sure sign of a successful rescue mode reboot is being shown the message:

```

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@      WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the RSA key sent by the remote host is
54:58:68:c1:5a:90:9d:85:26:20:bd:6c:c7:01:4f:57.
Please contact your system administrator.
RSA host key for 130.56.252.26 has changed and you have requested strict
checking.
Host key verification failed.
```

If you do get this message, simply delete the offending line from your `~/.ssh/known_hosts` file and try again. If you don't get this message then the rescue failed and you are ssh'ing into the same instance. Exit, undo the rescue and then try again.

Once you have ssh'd into the new instance:

You should find the old instances file system at `/mnt`.

When you are finished inspecting the file system and the log files, and made any required changes, exit and un-rescue the instance:

```
$ nova unrescue <instance id>
```


Again, Nova list will show you when the instance is Active again:

```
$ nova list
+-----+-----+-----+-----+-----+
| ID                                     | Name | Status | Task State | Power
State | Networks |
+-----+-----+-----+-----+-----+
| 598ec2ba-9d87-484d-ae3a-4661b4be3d91 | tester2 | ACTIVE | -          | Running
      | nci=130.56.252.26 |
+-----+-----+-----+-----+-----+
```

Is the service actually running?

When trying to connect to a service on an instance it can be more than frustrating if it's not running. Try to confirm that the service has in fact been started. The following link gives instructions on how to examine processes that have been running (<http://www.cyberciti.biz/faq/show-all-running-processes-in-linux/>)

Is the service running on the port you think it is?

Sometimes the service might be running, but not listening on the port that you think it is. An examination of the services configuration files might help show what port it's running on.

Is there a local firewall preventing you from accessing the service?

Sometimes a misconfigured firewall on the instance might be preventing you from accessing the service that you are trying to reach. Try shutting it down temporarily and see if you can connect. Sometimes your organization might be running a proxy or a firewall that is blocking you: and they haven't notified you...

Can you connect to the service with telnet?

A transparent way of testing network communications with text based services is to use telnet. The following link gives a good breakdown of connecting with telnet: <http://www.anta.net/misc/telnet-troubleshooting/>.

If you can connect via telnet then the error is more likely to be in your client application.

Would connecting a remote debugger to the instance help?

If you are developing the application that is failing, you can connect a remote debugger to your application and track what is happening to it.

Although not written for the NeCTAR cloud, the following pages give a good overview of the process for:

- Tomcat -
<http://blog.halcyon-solutions.com/2010/04/remote-debugging-apache-tomcat-on.html>
- Django -
<http://karumiblog-env.elasticbeanstalk.com/how-to-remotely-debug-django-apps-hosted-in-ec2/>

Is the issue still not resolved?

If all else fails, submit a support request to NeCTAR. The information on how to do this is here:
<http://cloud.nectar.org.au/contact/>.