

# Persist DevStack After Host Reboot

## How to persist OpenStack / DevStack configurations across rebooting of the hosting machine

This page documents the instructions on how to persist OpenStack / DevStack configurations across rebooting of the hosting machine, and tips on troubleshooting OpenStack / DevStack from Rui's experience.

Chad and Rob helped Rui set up a Ubuntu 12.04 VM named "devstack.ncsa.illinois.edu", to host the DevStack server that is used in the VM Elasticity work. Rui installed a running cloud / stack on it using DevStack. Later Rob requested Rui to reboot it to update the OS.

Thanks to the instruction in "Rebooting with DevStack?" at: <https://ask.openstack.org/en/question/5423/rebooting-with-devstack/>, Rui tried it, debugged issues encountered, and verified the following steps to successfully retain the DevStack configurations across rebooting of a hosting machine. Do:

1. Run `./unstack.sh`.  
This cleanly stops the stack services and screen processes.
2. Reboot the machine.
3. **Manually start the RabbitMQ server.**  
Somehow the RabbitMQ server is not automatically started. Do `sudo service rabbitmq-server start`. Then do `sudo service rabbitmq-server status` to verify that it is started successfully.
4. Run `./rejoin-stack.sh`, do NOT use `./stack.sh` to start the stack.

This will use the "stack-screenrc" file, restart the stack, and keep all the configurations that have been made, such as uploaded images, VM instances in suspended (or shelved, but Rui did not test the shelved state) state, and uploaded key pairs. 😊 For example, the VM instances can be seen both in the "nova list" result and the web UI, and "nova resume ..." of suspended VM instances work.

Notes:

1. When Rui tried it, the RabbitMQ server did not restart by itself after reboot, and caused many nova commands, including "resume" and "delete" to seem to be stuck, since there was no reply from the RabbitMQ queue. Rui spent a lot of time debugging it, but hopefully this finding and solution help you save time. 😊
2. While debugging, learned that if a VM instance went into a state that got stuck and could not be deleted, a user can use `nova reset-state <vm_instance1>` to change its state to "error", then can use the "delete" command to delete it:  
`nova delete <vm_instance1>`
3. Also learned that as a last resort, a user can manually change the VM instance data in the mysql database. Rui had to do that to rescue a VM instance that went into the "resuming" task-state, and got stuck there – could not be changed, could not be deleted. The reason was that RabbitMQ server was not started, so there was no response for "nova resume ...". Rui started "mysql", and used:

```
show databases;
```

```
use nova;
```

```
show tables;
```

```
describe instances;
```

```
select count(*) from instances;
```

```
select id, display_name, image_ref, power_state, vm_state, task_state from instances; (showed that the VM "medici-user-data" with id 1 was in the "resuming" task_state)
```

```
update instances set task_state = NULL where id = 1;
```

Then the web UI showed that the VM's task state changed to "None" from "resuming". Then,

```
nova resume medici-user-data
```

returned instead of getting stuck as in previous invocations. After a short while, pinging its IP returned reachable. Then,

```
ssh ubuntu@10.11.12.2
```

worked! :-)