



National Science Foundation
WHERE DISCOVERIES BEGIN

Chameleon Cloud Tutorial Fundamental Security Lab

Fundamental Security Lab

High Level Agenda

Objectives

In this tutorial, you will learn the basic concepts about Chameleon Identity Service, and how to access the Chameleon resource securely. This tutorial will give the audience understandings on what are the "Chameleon" security features and show you how these features work.

Tutorial Action	Time Required
Step 1: Introduce the basic Identity Service concepts Introduce the basic concepts about Chameleon Identity Service, and illustrate how the Chameleon Identity service cooperates with other components by using a simple example. Introduce the 2 ways (Dashboard Access& API Access) Chameleon provides to perform tasks.	10 minus
Step 2: Secure your instance by using ‘Security Group’ We will how to create the firewall rules through the Chameleon security Group and how to implement it to your instance,	5 minutes
Step 3: Access the instance securely by SSH We will introduce the Public-key cryptography concepts and show how to implement it by using your own key or creating keys in Chameleon	15 minutes
Step 4: Floating IPs Lastly, you will learn how to allocate and release Floating IPs and associate and dissociate Floating IPs to instance on Chameleon Cloud.	10 minutes

Prerequisites

The following prerequisites are expected for successful completion of this tutorial:

- An SSH client (Windows users: download PuTTY from here: <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>)
- A basic knowledge of Linux
- A basic knowledge of Firewall and IPs-filter
- A basic knowledge of cryptography

Step 1: Introduce the basic Identity Service concepts

1.1) The basic Identity Service concepts

To understand OpenStack Identity, you must understand the following concepts:

User Digital representation of a person, system, or service who uses OpenStack cloud services. The Identity service validates that the user who claims to be making the call makes incoming requests. Users have a login and may be assigned tokens to access resources. Users can be directly assigned to a particular tenant and behave as if they are contained in that tenant.

Credentials Data that confirms the user's identity. For example: user name and password, user name and API key, or an authentication token provided by the Identity Service.

Authentication The process of confirming the identity of a user. OpenStack Identity confirms an incoming request by validating a set of credentials supplied by the user.

These credentials are initially a user name and password, or a user name and API key. When user credentials are validated, OpenStack Identity issues an authentication token, which the user provides in subsequent requests.

Token An alphanumeric string of text used to access OpenStack APIs and resources. A token may be revoked at any time and is valid for a finite duration.

While OpenStack Identity supports token-based authentication in this release, the intention is to support additional protocols in the future. Its main purpose is to be an integration service, and not aspire to be a full-fledged identity store and management solution.

Tenant A container used to group or isolate resources. Tenants also group or isolate identity objects. Depending on the service operator, a tenant may map to a customer, account, organization, or project.

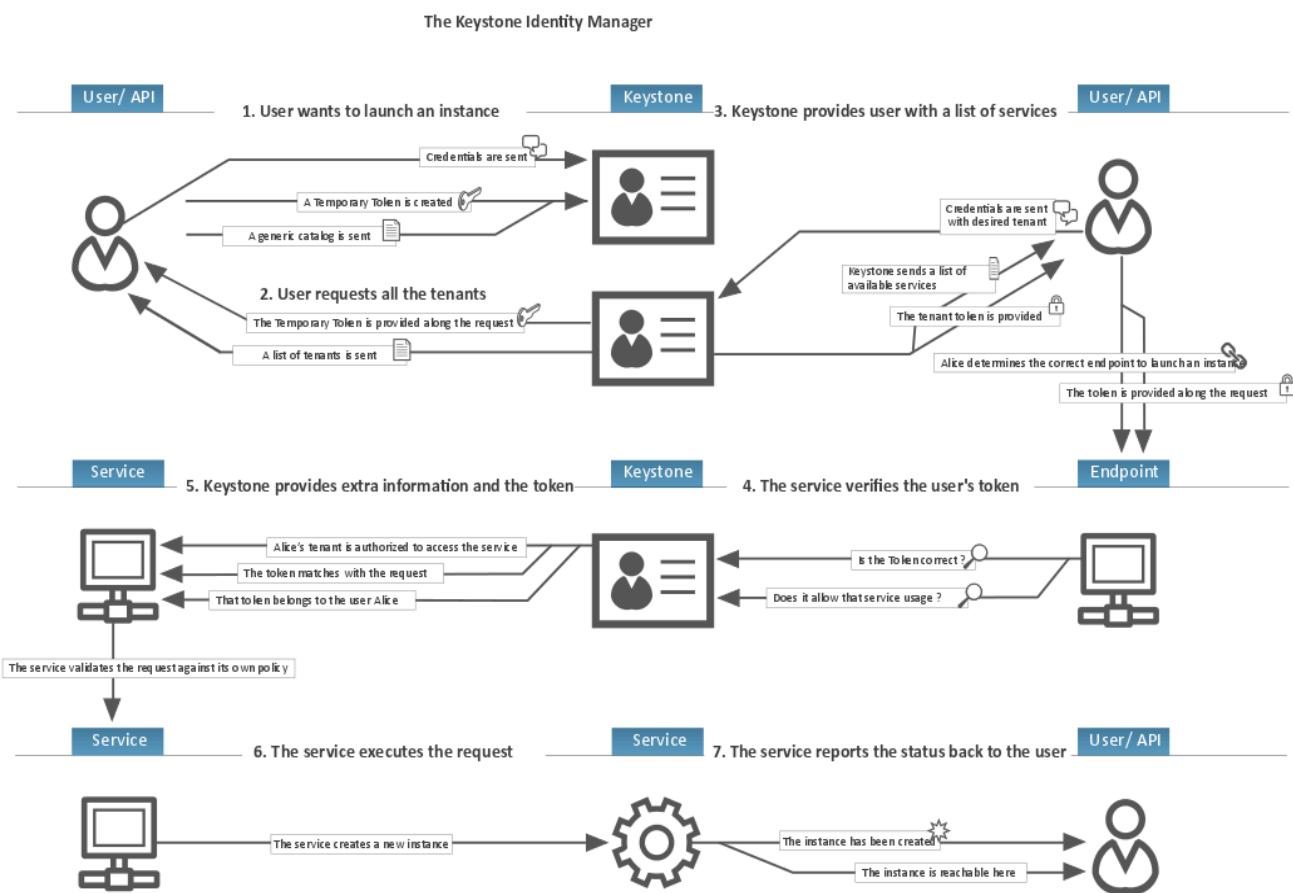
Service An OpenStack service, such as Compute (nova), Object Storage (swift), or Image service (glance). It provides one or more endpoints in which users can access resources and perform operations.

Endpoint A network-accessible address where you access a service, usually a URL address. If you are using an extension for templates, an endpoint template can be created, which represents the templates of all the consumable services that are available across the regions.

Role A personality with a defined set of user rights and privileges to perform a specific set of operations.

In the Identity service, a token that is issued to a user includes the list of roles. Services that are being called by that user determine how they interpret the set of roles a user has and to which operations or resources each role grants access.

The following diagram shows the OpenStack Identity process flow:



1.2) Access the Chameleon through the command-line clients

As a Chameleon cloud end user, you can provision your own resources within the limits set by administrators. Chameleon provides the following 2 ways to perform tasks on the cloud:

OpenStack dashboard. Use this web-based graphical interface, code named horizon, to view, create, and manage resources.

OpenStack command-line clients. Each OpenStack project has a command-line client that you can use to run simple commands to view, create, and manage resources in a cloud and automate tasks by using scripts.

Here we are going to discuss how to perform tasks through command-line clients. To do this, first

1.2.1) Install the command-line clients

Use pip to install the OpenStack clients on a Mac OS X or Linux system. It is easy and ensures that you get the latest version of the client from the Python Package Index. Also, pip lets you update or remove a package.

You must install each client separately.

Run this command to install or update a client package:

```
$ pip install [--upgrade] python-PROJECTclient
```

Where PROJECT is the project name.

For example, to install the nova client, run this command:

```
$ pip install python-novaclient
```

To update the nova client, run this command: \$ pip install --upgrade python-novaclient

To remove the nova client, run this command: \$ pip uninstall python-novaclient

1.2.2) get credential

You must have the appropriate credentials if you want to use the command-line tools to make queries against your OpenStack cloud. By far, the easiest way to obtain authentication credentials to use with command-line clients is to use the OpenStack dashboard. From the top-right navigation row, select **Project**, then **Access & Security**, then **API Access** to find out the “**Download the OpenStack RC file**” button as the following:

Service	Service Endpoint
Baremetal	https://ironic.chameleon.tacc.utexas.edu:6385
Compute	https://ironic.chameleon.tacc.utexas.edu:8774/v2/CH-816772
Network	https://ironic.chameleon.tacc.utexas.edu:9696
Volumev2	https://ironic.chameleon.tacc.utexas.edu:8776/v2/CH-816772
Image	https://ironic.chameleon.tacc.utexas.edu:9292
Metering	https://ironic.chameleon.tacc.utexas.edu:8777
Volume	https://ironic.chameleon.tacc.utexas.edu:8776/v1/CH-816772
Reservation	https://ironic.chameleon.tacc.utexas.edu:1234/v1
Identity	https://ironic.chameleon.tacc.utexas.edu:5000/v2.0
Identityv3	https://ironic.chameleon.tacc.utexas.edu:5000/v3

After clicked the “**Download the OpenStack RC file**” button, a RC file would be downloaded to your computer. The downloaded file looks like this:

```
#!/bin/bash
```

```
# To use an Openstack cloud you need to authenticate against keystone,
which
# returns a **Token** and **Service Catalog**. The catalog contains
the
# endpoint for all services the user/tenant has access to - including
nova,
# glance, keystone, swift.
#
# *NOTE*: Using the 2.0 *auth api* does not mean that compute api is
2.0. We
# will use the 1.1 *compute api*
export OS_AUTH_URL=https://ironic.chameleon.tacc.utexas.edu:5000/v2.0
```

```

# With the addition of Keystone we have standardized on the term
**tenant**
# as the entity that owns the resources.
export OS_TENANT_ID=CH-816772
export OS_TENANT_NAME="CH-816772"

# In addition to the owning entity (tenant), openstack stores the
entity
# performing the action as the **user**.
export OS_USERNAME="paulrad"

# With Keystone you pass the keystone password.
echo "Please enter your OpenStack Password: "
read -sr OS_PASSWORD_INPUT
export OS_PASSWORD=$OS_PASSWORD_INPUT

# If your configuration has multiple regions, we set that information
here.
# OS_REGION_NAME is optional and only valid in certain environments.
export OS_REGION_NAME="regionOne"
# Don't leave a blank variable, unset it if it was empty

if [ -z "$OS_REGION_NAME" ]; then unset OS_REGION_NAME; fi

```

You can now initialize our shell environment to communicate with the Chameleon. From a terminal shell on your own machine, run: source /CH-816772-openrc.sh

This command will prompt you for a password. Type your Chameleon password (it won't be displayed in your terminal) and press Enter. **Note: adapt the path of the RC file depending on where you downloaded it. It should be at the above location on OS X.**

Now you can run client commands, for example you can use “nova list” to view the instances under the project “CH-816772”:

```

sh-3.2# nova list
+-----+-----+
| ID          | Name        | Status |
| Task State | Power State | Networks
+-----+-----+
| 85db8788-8009-4773-b566-792eec393756 | ali-ukf902-n01 | ACTIVE |
|           | Running     | sharednet1=10.12.0.34, 129.114.34.116 |
| 69185323-e3bb-4836-bb41-959cd4b577bc | joseph-mpq055-n01 | ACTIVE |
|           | Running     | sharednet1=10.12.0.26, 129.114.34.117 |
| cc490e03-7f60-462f-825a-dd1f6651e293 | karthi-khj059-n01 | ACTIVE |
|           | Running     | sharednet1=10.12.0.30, 129.114.34.118 |

```

```

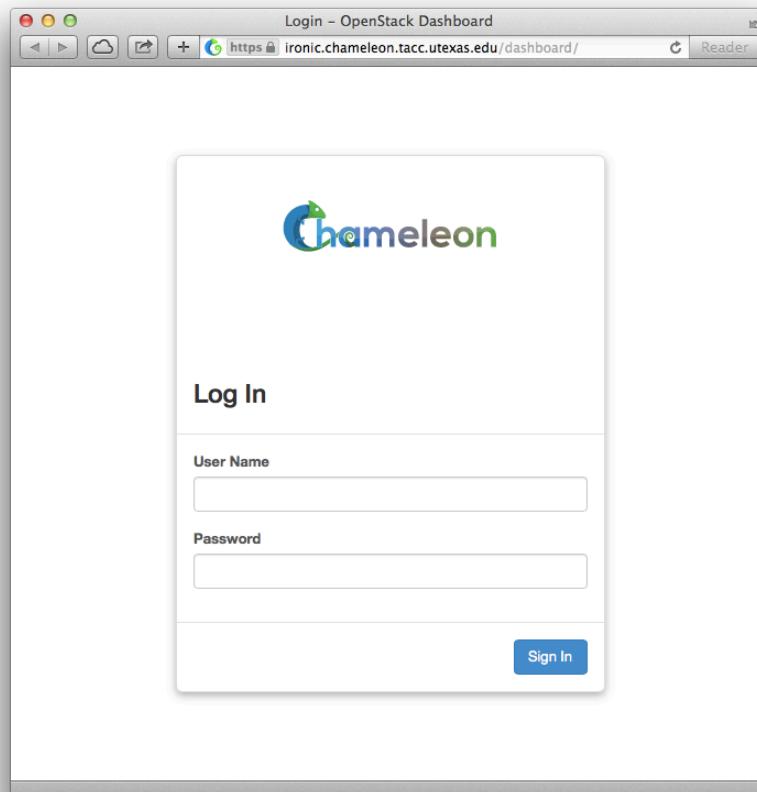
| 2fc2ddf7-5db3-47e6-b2ae-81e5e93b2b8b | ron-xzn964-n01 | ACTIVE |
|           | Running      | sharednet1=10.12.0.252, 129.114.34.122 |
| 4cbb5013-da23-4e8b-afc1-3ce3801065f2 | shawn-hfl398-n01 | ACTIVE |
|           | Running      | sharednet1=10.12.0.50, 129.114.34.119 |
| 7cb2530f-e436-48c2-a9a6-77b6ed8f1c85 | shawn-hfl398-n02 | ACTIVE |
|           | Running      | sharednet1=10.12.0.51, 129.114.34.124 |
| 6a561b54-17de-4df2-838b-021eba69bae3 | shravya-gcg047-n03 | ACTIVE |
|           | Running      | sharednet1=10.12.0.42, 129.114.34.120 |
+-----+
+-----+
sh-3.2#

```

Step 2: Secure your instance by using ‘Security Group’

2.1) Setup IP filter rules by using the ‘Security Group’.

To begin, login to Chameleon Resource Provisioning Dashboard, located at
<https://ironic.chameleon.tacc.utexas.edu/dashboard/auth/login/>



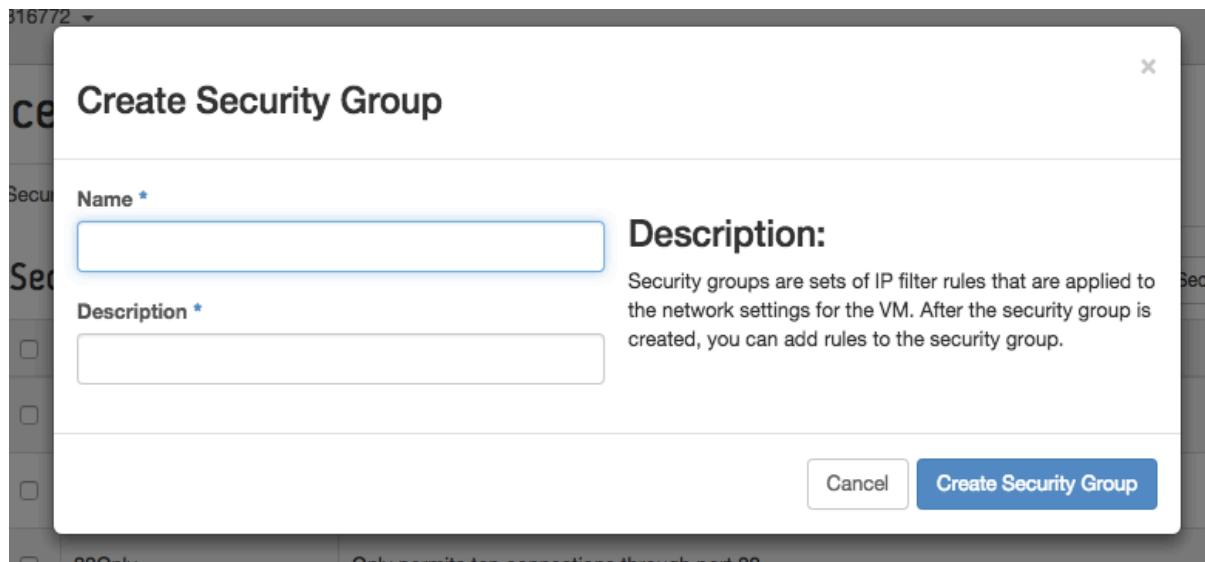
To access the Security Groups, click on “Projects”, “Compute”, then “Access & Security”.

The screenshot shows the AWS Management Console interface. On the left, there's a sidebar with 'Project' selected under 'Compute'. Under 'Access & Security', there are tabs for 'Security Groups', 'Key Pairs', 'Floating IPs', and 'API Access'. The 'Security Groups' tab is active. It displays a table with three rows:

<input type="checkbox"/>	Name	Description	Actions
<input type="checkbox"/>	tempGroup	Testing security group settings.	<button>Manage Rules</button>
<input type="checkbox"/>	default	default	<button>Manage Rules</button>
<input type="checkbox"/>	22Only	Only permits tcp connections through port 22.	<button>Manage Rules</button>

At the bottom of the table, it says 'Displaying 3 items'. There are also '+ Create Security Group' and 'Delete Security Groups' buttons at the top right of the table area.

To setup new IP filter rules, you may **click on the “+Create Security Group” Button**. It should bring up the window displayed below:



1. Pick a name for the Security Group. This name needs to be unique across the security groups. This example uses the name tutorial_Security_Lab.
2. Write the description about the security group.
3. **After completing, click on the “Create Security Group” button.** It should bring up the window displayed below:

Security Groups	Key Pairs	Floating IPs	API Access
Security Groups			
<input type="checkbox"/>	Name	Description	Actions
<input type="checkbox"/>	tempGroup	Testing security group settings.	<button>Manage Rules</button>
<input type="checkbox"/>	tutorial_Security_Lab	IP filter rules for tutorial	<button>Manage Rules</button>
<input type="checkbox"/>	default	default	<button>Manage Rules</button>
<input type="checkbox"/>	22Only	Only permits tcp connections through port 22.	<button>Manage Rules</button>
Displaying 4 items			

Once created the security group. It will be displayed. **Click on the “Manage Rules” button.** It should bring up the window displayed below:

Manage Security Group Rules: tutorial_Security_Lab

Security Group Rules						
<input type="checkbox"/>	Direction	Ether Type	IP Protocol	Port Range	Remote	Actions
<input type="checkbox"/>	Egress	IPv6	Any	-	::/0 (CIDR)	<button>Delete Rule</button>
<input type="checkbox"/>	Egress	IPv4	Any	-	0.0.0.0/0 (CIDR)	<button>Delete Rule</button>
Displaying 2 items						

You can add and delete rules in this page. Rules define which traffic is allowed to instances assigned to the security group. A security group rule consists of three main parts:

Rule: You can specify the desired rule template or use custom rules, the options are Custom TCP Rule, Custom UDP Rule, or Custom ICMP Rule.

Open Port/Port Range: For TCP and UDP rules you may choose to open either a single port or a range of ports. Selecting the "Port Range" option will provide you with space to provide both the starting and ending ports for the range. For ICMP rules you instead specify an ICMP type and code in the spaces provided.

Remote: You must specify the source of the traffic to be allowed via this rule. You may do so either in the form of an IP address block (CIDR) or via a source group (Security Group). Selecting a security group as the source will allow any other instance in that security group access to any other instance via this rule.

When you **click on the “Manage Rules” button.** It should bring up the window displayed below:

Add Rule

Rule *
Custom TCP Rule

Direction
Egress

Open Port *
Port
80

Port ?
80

Remote * ?
CIDR
0.0.0.0/0

Description:
Rules define which traffic is allowed to instances assigned to the security group. A security group rule consists of three main parts:
Rule: You can specify the desired rule template or use custom rules, the options are Custom TCP Rule, Custom UDP Rule, or Custom ICMP Rule.
Open Port/Port Range: For TCP and UDP rules you may choose to open either a single port or a range of ports. Selecting the "Port Range" option will provide you with space to provide both the starting and ending ports for the range. For ICMP rules you instead specify an ICMP type and code in the spaces provided.
Remote: You must specify the source of the traffic to be allowed via this rule. You may do so either in the form of an IP address block (CIDR) or via a source group (Security Group). Selecting a security group as the source will allow any other instance in that security group access to any other instance via this rule.

Add

- Rules** Pick a type for this rule, a template rule or a custom rule.
- Directions** Select a direction for this rule, Ingress or Egress.
- Open Port** Choose to open either a single port or a range of ports.
- Port** Choose the port number or a range of ports.
- Remote** Specify the source of the traffic to be allowed.
- After completing, click on the “Add” button.** It should bring up the window displayed below:

Manage Security Group Rules: tutorial_Security_Lab

Security Group Rules

<input type="checkbox"/>	Direction	Ether Type	IP Protocol	Port Range	Remote	Actions
<input type="checkbox"/>	Egress	IPv6	Any	-	::/0 (CIDR)	Delete Rule
<input type="checkbox"/>	Egress	IPv4	Any	-	0.0.0.0/0 (CIDR)	Delete Rule
<input type="checkbox"/>	Egress	IPv4	TCP	80 (HTTP)	0.0.0.0/0 (CIDR)	Delete Rule

Displaying 3 items

Then you can see that the rule has been added to the Security Group "tutorial_Security_Lab"

2.2) Implement the rules to the instances

Once your security group has been setup; you can implement it to the instance. There are 2 ways you can do it. You can do it while you lance a new instance like the following:

Launch Instance

Details * **Access & Security *** Networking * Post-Creation

Key Pair  ron_chameleon   

Security Groups * 
 tempGroup
 tutorial_Security_Lab
 default
 22Only

Control access to your instance via key pairs, security groups, and other mechanisms.

Cancel **Launch**

Or you can change the security group of instances that have been created. To perform this operation, click on the “Edit Security Groups” Button.

Project  Compute Overview Instances Volumes Images Access & Security Network Reservations Identity

Instances

Instances  ron-xzn964-n01    

<input type="checkbox"/>	Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
<input type="checkbox"/>	ron-xzn964-n01	CC-CentOS7	10.12.0.252 129.114.34.122	baremetal	ron_chameleon	Active	climate:fa51d792-a166-4ce4-9fcf-4fc8801d3143	None	Running	1 week, 2 days	       

Displaying 1 item

After click on the “Edit Security Groups” Button, you can see the following screen:

Edit Instance

Info *

Security Groups

Add and remove security groups to this project from the list of available security groups.

All Security Groups	Filter	Search
tempGroup	+	
22Only	+	
default	+	

Instance Security Groups	Filter	Search
tutorial_Security_Lab		-

[Cancel](#)

[Save](#)

You can add or remove security groups for your instance by click on the “+” or “-” buttons.

Step 3: Access the instance securely by SSH

You can access your instance securely by using SSH. In Chameleon, you can either get Open Stack to create an SSH key pair for you or use your own SSH key pair on your machine. We are going to introduce them respectively.

3.1) Using Public Key Pair created by Chameleon

Go to **Project > Compute > Access & Security**, then select the Key Pairs tab.

The screenshot shows the Chameleon Access & Security dashboard. On the left, there's a sidebar with a 'Project' dropdown set to 'Compute', and sections for Overview, Instances, Volumes, Images, and Access & Security (which is selected). Below that are Network, Reservations, and Identity sections. The main content area is titled 'Access & Security' and has tabs for Security Groups, Key Pairs (which is active), Floating IPs, and API Access. Under 'Key Pairs', there's a table with columns for Key Pair Name, Fingerprint, and Actions. A red box highlights the '+ Create Key Pair' button at the top right of the table area.

Here you **click on the “Create Key Pair” button**, and it should bring up the window displayed below:

The modal window is titled 'Create Key Pair'. It has a 'Key Pair Name *' input field with a placeholder 'keypair1'. To the right, there's a 'Description:' section with a detailed explanation of what key pairs are and how they are used. At the bottom right are 'Cancel' and 'Create Key Pair' buttons.

Pick a name for your Key Pair, and **click on the “Create Key Pair” button**. Then the Chameleon will create a key Pair for you. Key pairs are ssh credentials which are injected into images when they are launched. Creating a new key pair registers the public key and downloads the private key (a .pem file). Please protect the private key in a secure folder (for example: only your account have the full rights on this folder, and the other accounts have no access right to this folder).

After your instance is launched, you should be able to connect to the instance via SSH using the cc account. In a terminal, type `ssh cc@<floating_ip>`, in our example this would be

```
ssh -i /Users/zhuromghua/documents/key/ron_chameleon.pem cc@129.114.34.122
```

Type yes and press Enter. You should arrive to a prompt like this one:

```
[cc@ron-xzn964-n01 ~]$
```

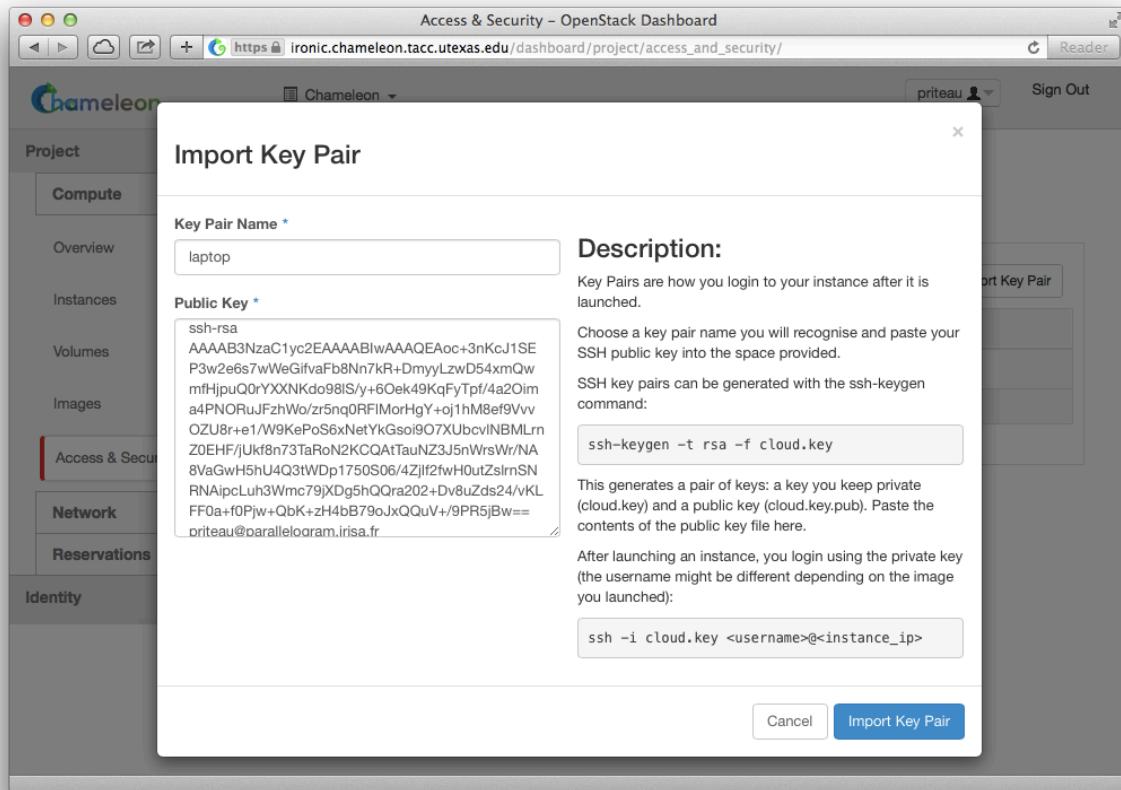
3.2) Using your own Public Key Pair.

You still can use your own SSH key pair on your machine.

Go to Project > Compute > Access & Security, then select the Key Pairs tab. Click on "Import Key Pair".

The screenshot shows the Chameleon OpenStack Dashboard. The URL is https://ironic.chameleon.tacc.utexas.edu/dashboard/project/access_and_security/. The dashboard has a sidebar on the left with categories: Project (Compute selected), Overview, Instances, Volumes, Images, Access & Security (selected), Network, Reservations, and Identity. The main content area is titled 'Access & Security' and has tabs for Security Groups, Key Pairs (selected), Floating IPs, and API Access. Under the 'Key Pairs' tab, there is a table with columns: Key Pair Name, Fingerprint, and Actions. A message says 'No items to display.' Below the table, it says 'Displaying 0 items'. There are two buttons at the top right: '+ Create Key Pair' and 'Import Key Pair' (which is highlighted with a red box).

It should bring up the window displayed below:



Enter a name for the key pair, for example laptop. In the "Public Key" box, copy the content of your SSH public key. Typically it will be at `~/.ssh/id_rsa.pub`. On Mac OS X, you can run in a terminal:

```
cat ~/.ssh/id_rsa.pub | pbcopy
```

It copies the content of the public key to your copy/paste buffer. Then you can simply paste in the "Public Key" box. Then, click on the blue "Import Key Pair" button. This should show you the list of key pairs, with the one you just added.

The screenshot shows the Chameleon Access & Security - OpenStack Dashboard. The URL is https://ironic.chameleon.tacc.utexas.edu/dashboard/project/access_and_security/. The dashboard has a sidebar with 'Project' dropdown (Compute), 'Overview', 'Instances', 'Volumes', 'Images', 'Access & Security' (selected), 'Network', 'Reservations', and 'Identity'. The main content area is titled 'Access & Security' with tabs for 'Security Groups', 'Key Pairs' (selected), 'Floating IPs', and 'API Access'. A green success message box says 'Success: Successfully imported public key: laptop'. The 'Key Pairs' table shows one item: 'laptop' with fingerprint '1c:cb:7c:ce:28:e8:1a:1b:f3:7a:6b:bf:a4:58:b6:22'. Actions include '+ Create Key Pair', '@ Import Key Pair', and 'Delete Key Pair'.

Now you should be able to connect to the instance via SSH using the cc account. In a terminal, type ssh cc@<floating_ip>, in our example this would be

```
ssh cc@129.114.34.80
```

SSH will probably tell you:

The authenticity of host '129.114.34.80 (129.114.34.80)' can't be established.

RSA key fingerprint is 5b:ca:f0:63:6f:22:c6:96:9f:c0:4a:d8:5e:dd:fd:eb.

Are you sure you want to continue connecting (yes/no)?

Type yes and press Enter. You should arrive to a prompt like this one:

```
[cc@my-first-instance ~]$
```

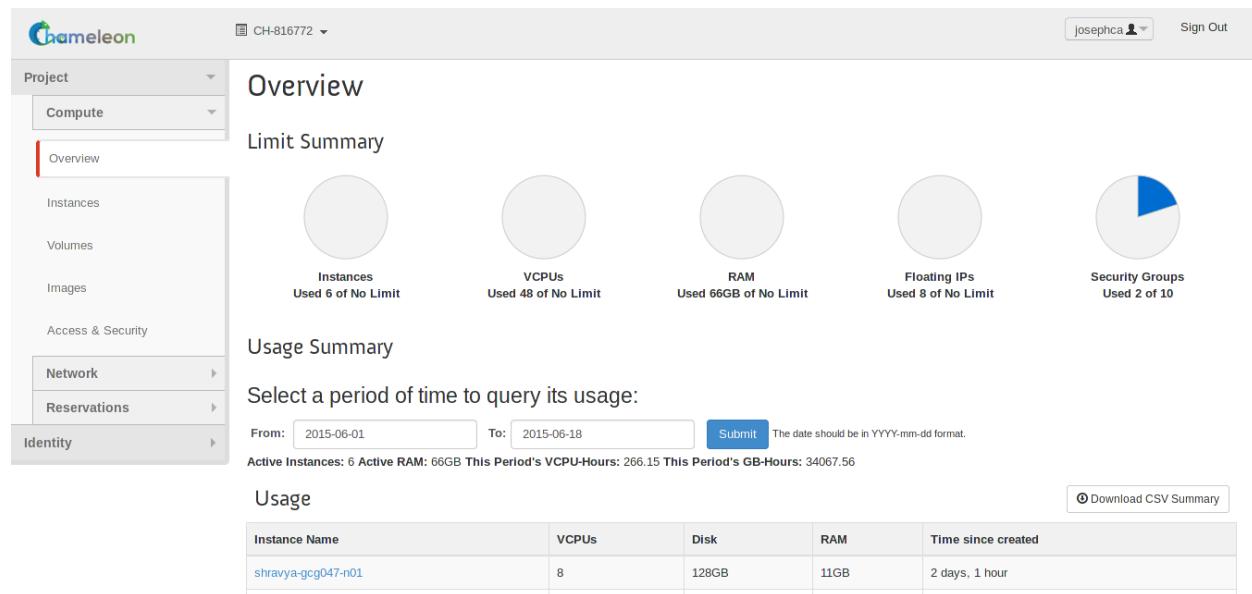
Step 4: Floating IPs

The Floating IPs subsection of the Chameleon Cloud web interface provides an easy-to-use method of managing the IPs of a project on an instance-by-instance basis. The actions one may take include allocating or deallocated IPs for the project as a whole as well as individual IP address assignment. Within the context of the Chameleon Cloud, Floating IPs serve as publicly available IP address that maps to the assigned internal port of a network.

This tutorial is more so an interactive way to get a user accustomed to how to manipulate Floating IPs and what usability they can provide. Going into this tutorial, the expectation is that the user already has an instance created. If the Floating IP is changed for an instance, the user must also change their current configuration and setup for that specific instance upon association.

Step 4.1: Navigating the Floating IPs Subsection

Starting from the beginning, navigate to the project homepage whose IPs you wish to manage. The primary dashboard page, <https://ironic.chameleon.tacc.utexas.edu/dashboard/project/>, should look as follows.



The screenshot shows the Chameleon Cloud Project Overview dashboard. At the top, there's a header with the Chameleon logo, a dropdown for 'CH-816772', a user profile for 'josephca', and a 'Sign Out' button. On the left, a sidebar menu is open under 'Project > Compute > Access & Security'. The main area is titled 'Overview' and includes a 'Limit Summary' section with five circular progress indicators: Instances (Used 6 of No Limit), VCPUs (Used 48 of No Limit), RAM (Used 66GB of No Limit), Floating IPs (Used 8 of No Limit), and Security Groups (Used 2 of 10). Below this is a 'Usage Summary' section with a form to 'Select a period of time to query its usage' (From: 2015-06-01, To: 2015-06-18, Submit). It displays active instances (6), total RAM (66GB), VCPU-Hours (266.15), and GB-Hours (34067.56). A table titled 'Usage' shows details for one instance: shrawya-gcg047-n01, with 8 VCPUs, 128GB Disk, 11GB RAM, and a creation time of 2 days, 1 hour. A 'Download CSV Summary' button is also present.

On the left-hand side, in the collapsible menu, find and click on **Project > Compute > Access & Security**

From the **Access & Security** section, you may now navigate to the **Floating IPs** subsection.

Access & Security

The screenshot shows the Chameleon interface with the 'Access & Security' tab selected. Under the 'Floating IPs' tab, there is a table with one row labeled 'default'. A 'Manage Rules' button is visible in the Actions column.

Name	Description	Actions
default	default	Manage Rules

Now, you will be presented with the main Floating IPs page where we will be managing the IP addresses for our project.

The screenshot shows the Chameleon interface with the 'Access & Security' tab selected. Under the 'Floating IPs' tab, there is a table listing several floating IP addresses, their mapped fixed IP addresses, their floating IP pool, and actions for each entry.

IP Address	Mapped Fixed IP Address	Floating IP Pool	Actions
192.168.34.119	shawn-hfl398-n01 10.12.0.253	ext-net	Disassociate ▾
192.168.34.121	joseph-mpq055-n01 10.12.0.26	ext-net	Disassociate ▾
192.168.34.120	shravya-gcg047-n01 10.12.0.250	ext-net	Disassociate ▾
192.168.34.118	karthi-kjh059-n01 10.12.0.30	ext-net	Disassociate ▾
192.168.34.116	-	ext-net	Associate ▾
192.168.34.122	ron-xzn864-n01 10.12.0.252	ext-net	Disassociate ▾
192.168.34.117	-	ext-net	Associate ▾
192.168.34.115	-	ext-net	Associate ▾

Mainly presented on the page above is a table that is broken into different columns. The first column, **IP Address**, lists the public-facing IP address for the row. Second is the **Mapped Fixed IP Address**, which is used to show the instance, if any, the IP address is currently assigned to. Additionally, it also shows which internal IP address the floating IP is mapped to. Next, the **Floating IP Pool** column displays which of the networks the floating IP is from. Finally, the **Actions** column contains a dropdown menu, button combination which allows control over each entry that we will cover in the following sections.

Step 4.2: Allocate a Floating IP Address

Depending on the current state of your project, you may have several IP addresses already allocated to your project or may not have any. In the previous screen captures, you can see that the example showed several already allocated IP addresses in various states of use. To start with, we will allocate a Floating

IP address. **NOTE: You may not allocate a Floating IP Address if you have already reached the maximum permitted Floating IPs for the project.**

In order to allocate a Floating IP Address, click on the **Allocate IP to Project** button located towards the upper right-hand side of the inner page.

The screenshot shows a web-based interface for managing floating IP addresses. At the top, there are tabs for 'Security Groups', 'Key Pairs', 'Floating IPs' (which is selected), and 'API Access'. Below the tabs is a header row with columns for 'IP Address', 'Mapped Fixed IP Address', 'Floating IP Pool', and 'Actions'. The 'Actions' column contains red 'Disassociate' buttons for the first four entries and grey 'Associate' buttons for the last four. There are also dropdown arrows next to each 'Disassociate' button. The main body of the table lists eight items, each with a small icon and a checkbox. The 'Mapped Fixed IP Address' column contains links such as 'shawn-hfl398-n01 10.12.0.253'. The 'Actions' column for the last four items is currently inactive. A message at the bottom left says 'Displaying 8 items'.

	IP Address	Mapped Fixed IP Address	Floating IP Pool	Actions
<input type="checkbox"/>	192.168.34.119	shawn-hfl398-n01 10.12.0.253	ext-net	Disassociate ▾
<input type="checkbox"/>	192.168.34.121	joseph-mpq055-n01 10.12.0.26	ext-net	Disassociate ▾
<input type="checkbox"/>	192.168.34.120	shravya-gcg047-n01 10.12.0.250	ext-net	Disassociate ▾
<input type="checkbox"/>	192.168.34.118	karthi-khj059-n01 10.12.0.30	ext-net	Disassociate ▾
<input type="checkbox"/>	192.168.34.116	-	ext-net	Associate ▾
<input type="checkbox"/>	192.168.34.122	ron-xzn964-n01 10.12.0.252	ext-net	Disassociate ▾
<input type="checkbox"/>	192.168.34.117	-	ext-net	Associate ▾
<input type="checkbox"/>	192.168.34.115	-	ext-net	Associate ▾

After clicking the button, a dialog will appear over the current page for information and details about the IP that will be allocated. The box contains information about which pool the floating IP will be allocated from. This also shows the current number of allocated floating IPs for your project as well as your limit. Once you have confirmed the settings, click **Allocate IP**.

X

Allocate Floating IP

Pool *

ext-net

Description:

Allocate a floating IP from a given floating IP pool.

Project Quotas

Floating IP (8)

No Limit

Cancel

Allocate IP

Once complete, you will be returned to the Floating IPs page which will now be updated to show the newly allocated IP address at the bottom row of the table.

Security Groups

Key Pairs

Floating IPs

API Access

Floating IPs

Allocate IP To Project

Release Floating IPs

	IP Address	Mapped Fixed IP Address	Floating IP Pool	Actions
<input type="checkbox"/>	192.168.34.124	-	ext-net	<button>Associate</button>
<input type="checkbox"/>	192.168.34.119	shawn-hfl398-n01 10.12.0.253	ext-net	<button>Disassociate</button>
<input type="checkbox"/>	192.168.34.121	joseph-mpq055-n01 10.12.0.26	ext-net	<button>Disassociate</button>
<input type="checkbox"/>	192.168.34.120	shravya-gcg047-n01 10.12.0.250	ext-net	<button>Disassociate</button>
<input type="checkbox"/>	192.168.34.118	karthi-khj059-n01 10.12.0.30	ext-net	<button>Disassociate</button>
<input type="checkbox"/>	192.168.34.116	-	ext-net	<button>Associate</button>
<input type="checkbox"/>	192.168.34.122	ron-xzn964-n01 10.12.0.252	ext-net	<button>Disassociate</button>
<input type="checkbox"/>	192.168.34.117	-	ext-net	<button>Associate</button>
<input type="checkbox"/>	192.168.34.115	-	ext-net	<button>Associate</button>

Displaying 9 items

Upon completing this, you now have an additional floating IP address allocated to your project.

Step 4.3: Associate a Floating IP Address

Now that we have allocated a new Floating IP Address for us to use, we are going to use the “free” IP and associate it with the instance that currently does not have an external IP address assigned to it. From the main Floating IPs page, find the IP address that you wish to assign to your instance. Once found, follow the row all the way to the right and click on the **Associate** button.

<input type="checkbox"/>	129.114.34.121	-	ext-net	Associate 
--------------------------	----------------	---	---------	--

Next you will be shown a new dialog to set up which IP address should be associated with which instance. By default, the IP address whose row you selected from the previous page will be placed into the drop-down menu, but the menu will display all currently allocated and disassociated floating IP addresses belonging to the project. The second drop-down menu is a list of instances currently within the project that an IP address may be assigned to. Find the instance that does not have a floating IP address assigned and select it. Finally, click **Associate** to finalize the association.

Manage Floating IP Associations

IP Address * IP Address * 192.168.34.121  	Select the IP address you wish to associate with the selected instance.
Port to be associated * joseph-mpq055-n01: 10.12.0.26 	
 	

It is also beneficial to note that from this screen, one may also click the **+** button in order to allocate additional IP addresses similar to the method described in Step 4.1.

Step 4.4: Disassociate a Floating IP Address

Step 4.4 will cover how to go about disassociating an already associated IP address from an instance so that it may be moved around, used, or released.

To begin with, find the IP address column of the IP address you wish to disassociate. Once you have selected which IP address to disassociate, follow the column to the right and click the **Disassociate** button.

<input type="checkbox"/>	192.168.34.121	joseph-mpq055-n01 10.12.0.26	ext-net	Disassociate 
--------------------------	----------------	------------------------------	---------	---

This button will spawn a dialog that is used to confirm the IP address you wish to disassociate. Review it to ensure that you have selected the correct floating IP address.

Confirm Disassociate

You have selected "192.168.34.121". Please confirm your selection. This action cannot be undone.

Disassociate **Cancel**

Step 4.5: Release a Floating IP Address

The final step will show how to release a floating IP address from your project back into the floating IP pool. Whether a floating IP address is associated or not does not change whether or not it may be freely released. In such a case that a currently associated IP address is released, it will be automatically disassociated before release. To release an IP address, find the column in the table of the IP address you wish to release and click the drop-down menu. Select **Release Floating IP**.

	IP Address	Mapped Fixed IP Address	Floating IP Pool	Actions
<input type="checkbox"/>	192.168.34.124	-	ext-net	Associate 
<input type="checkbox"/>	192.168.34.119	shawn-hfl398-n01 10.12.0.253	ext-net	Release Floating IP 
<input type="checkbox"/>	192.168.34.121	joseph-mpq055-n01 10.12.0.26	ext-net	Disassociate 
<input type="checkbox"/>	192.168.34.120	shravya-gcg047-n01 10.12.0.250	ext-net	Release Floating IP 

Either of the two will work for disassociating an IP address. Once selected, you will be prompted with a confirmation dialog to ensure that you are releasing the correct floating IP. If you confirm the IP address, click the **Release Floating IP** button.

Confirm Release Floating IP

You have selected "192.168.34.115". Please confirm your selection. This action cannot be undone.

Release Floating IP

Cancel

Upon releasing a floating IP, you will find that it will be deleted from your project's Floating IPs table and is made available to the pool from where it was allocated originally. It is important to make note that once an IP address is released and no longer part of the project, there can be no guarantee that the same IP address can be retrieved through allocation, so take caution that nothing will break when you go through with releasing the IPs.

Should the need arise, the table also supports group actions on multiple IP addresses. In this case, it may be used to release all disassociated floating IPs from the project so that each IP has a corresponding instance. To do so, first use the check boxes to select all the floating IP addresses you wish to release. Once selected, click the **Release Floating IPs** button.

Floating IPs				
	IP Address	Mapped Fixed IP Address	Floating IP Pool	Actions
<input checked="" type="checkbox"/>	192.168.34.124	-	ext-net	<button>Associate ▾</button>
<input type="checkbox"/>	192.168.34.119	shawn-hfl398-n01 10.12.0.253	ext-net	<button>Disassociate ▾</button>
<input type="checkbox"/>	192.168.34.121	joseph-mpq055-n01 10.12.0.26	ext-net	<button>Disassociate ▾</button>
<input type="checkbox"/>	192.168.34.120	shravya-gcg047-n01 10.12.0.250	ext-net	<button>Disassociate ▾</button>
<input type="checkbox"/>	192.168.34.118	karthi-khj059-n01 10.12.0.30	ext-net	<button>Disassociate ▾</button>
<input checked="" type="checkbox"/>	192.168.34.116	-	ext-net	<button>Associate ▾</button>

With the previous actions, this one will also bring up a confirmation dialog to ensure that you are releasing the correct IPs. Read over the list to make sure that you are not releasing any IP that you did not mean to.

X

Confirm Release Floating IPs

You have selected "192.168.34.124", "192.168.34.116". Please confirm your selection. This action cannot be undone.

[Release Floating IPs](#)

[Cancel](#)

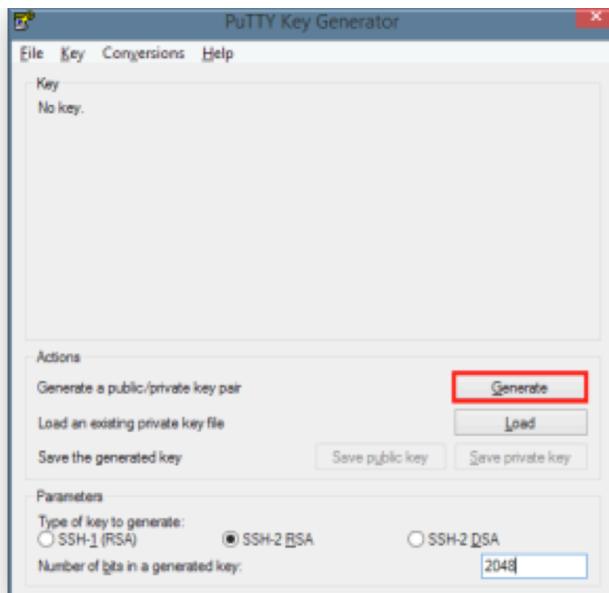
Appendix A: Generating and Using SSH Key Pairs on Windows Platforms

First, we will need to download tools to generate our key pair, as well as an SSH client that can make use of them. These tools are called PuTTYGen and PuTTY, respectively.

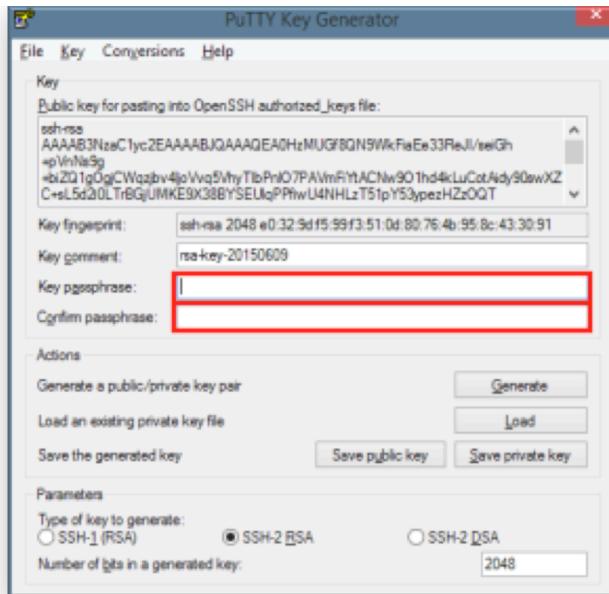
The location to download PuTTYGen is: <http://the.earth.li/~sgtatham/putty/latest/x86/puttygen.exe>

The location to download PuTTY is: <http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe>

Download and run the PuTTYGen executable:

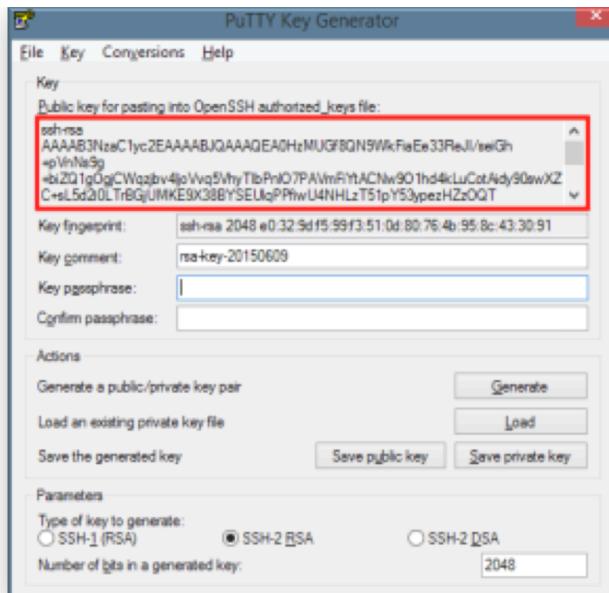


Click on the “Generate” button and move your mouse as indicated in order to generate enough random movement with which to help create your new key pair. When the process has completed, you may enter a password into the “Key passphrase” input box in order to protect your private key.

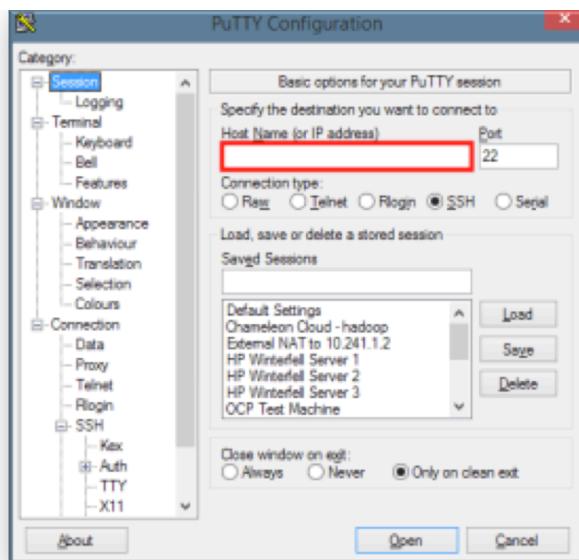


Click on the “Save private key” button and save this file to a location you can easily recall later.

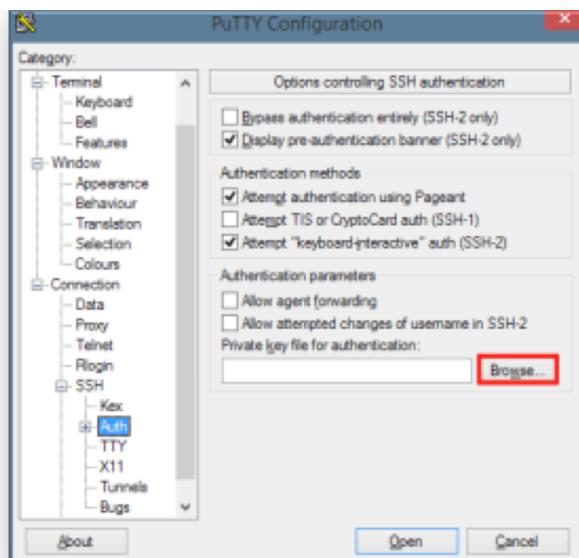
Copy the text in the “Public key for pasting into OpenSSH authorized_keys file:” input field. This is your public key that will be input into the Chameleon cloud web interface.



To connect to a Chameleon server, download and execute the PuTTY tool. Enter the public IP address into the “Host Name (or IP address)” input box.



Next, in order to specify a private key file, click on the Connection -> SSH -> Auth menu item on the left. Click on the “Browse” button and select your private key file.



Finally, click the “Open” button to connect using this key.