

---

# Table of Contents

## Support and Contributing

Introduction	1.1
Table of Contents	1.2
Support	1.3
Contributing	1.4
Contribute with Git	1.4.1
Git in the Command Line	1.4.2
CADES Authoring Guide	1.4.3
Glossary	1.5

## Birthright Cloud User Guide

Getting Started with OpenStack	2.1
Request Your Birthright Cloud Allocation	2.1.1
Manage Your Birthright Cloud Allocation	2.1.2
OpenStack Project Quota	2.1.3
OpenStack Help	2.1.4
Launch a VM Instance	2.2
Log In & Name the VM	2.2.1
Configure the VM	2.2.2
Networks & Security	2.2.3
Key Pair Use	2.2.4
Access VM Instances	2.3
Access Your VM Instance Using SSH	2.3.1
Access Your VM Instance Using PuTTY (Windows)	2.3.2
Access Your VM Instance Using Horizon	2.3.3
Add More Users to Your VM Instance	2.3.4
Manage Your VM Instances	2.4
Delete a VM Instance from Your Project	2.4.1
Delete a Volume from Your Project	2.4.2
Resize a VM Instance	2.4.3
Create a Snapshot	2.4.4
OpenStack Security Groups	2.4.5
Modify the Default Security Group	2.4.5.1
Create a new Security Group	2.4.5.2
Security Group CIDR Examples	2.4.5.3

---

Overview - Birthright Information	2.5
Available VM Images & Configurations	2.5.1
Software & Hardware	2.5.2
Network & Storage	2.5.3
Additional OpenStack Resources	2.6
Request Firewall Exception	2.6.1
Run a Simple Web Server	2.6.2
SSL - Let's Encrypt	2.6.3
Install CPUID	2.6.4
CPUID Hypervisor Codes	2.6.4.1
CPUID Instance Codes	2.6.4.2

## User Contributed Tutorials

Note: Community contributed content	3.1
not officially supported by CADES.	3.1.1
Launch a Docker Container	3.2
Launch Shiny within Docker	3.3
Eclipse in CADES HPC	3.4
Allinea DDT in CADES HPC	3.5

## SHPC Condo User Guide

Prerequisites	4.1
Getting Started	4.2
Request Your SHPC Condo Allocation	4.2.1
Access Your SHPC Condo Allocation	4.2.2
Execute a Job on Your SHPC Allocation	4.2.3
SHPC Condo Configuration	4.3
SHPC Condo LDAP Groups	4.4
Troubleshooting	4.5

## SHPC Condo Tutorials

Execute a Job on Your SHPC Allocation	5.1
Condo Workflows	5.2
Crystal Workflow	5.2.1

## CADES Policies

Birthright Cloud User Policy	6.1
------------------------------	-----



[CADES](#) → User Documentation

## CADES User Documentation

Oak Ridge National Laboratory's (ORNL) Compute and Data Environment for Science (CADES) provides an integrated computing infrastructure to deliver data science solutions and workflows to ORNL personnel. CADES provides dedicated computing resources through our **SHPC (Scalable High Performance Computing) Condo** allocations and customizable Software as a Service (SaaS) through our **Birthright Cloud** solution.

This diverse computing and data ecosystem is bolstered by a CADES support team that employs experts matrixed from different ORNL research directives to provide focused, expert support for a user's scientific computing needs.

### Features of this Guide

- The **navigation panel** on the left provides you with a birds-eye view of the content of these pages.
- GitBook **search** at the top of the left-hand side allows you to list only content that matches your keywords.

### CADES Quick Summary (**Cheat Sheet in pdf format**)

- [Download CADES Quick Summary Cheat Sheet](#)

# CADES User Documentation Table of Contents

## Support and Contributing

- [Table of Contents](#)
- [Support](#)
- [Contributing](#)
  - [Contribute with Git](#)
  - [Git in the Command Line](#)
  - [CADES Authoring Guide](#)
- [Glossary](#)

## Resource Guides

### Birthright Cloud User Guide

- [Getting Started with OpenStack](#)
  - [Request Your Birthright Cloud Allocation](#)
  - [Manage Your Birthright Cloud Allocation](#)
  - [OpenStack Project Quota](#)
  - [OpenStack Help](#)
- [Launch a VM Instance](#)
  - [Log In & Name the VM](#)
  - [Configure the VM](#)
  - [Networks & Security](#)
  - [Key Pair Use](#)
- [Access VM Instances](#)
  - [Access Your VM Instance Using SSH](#)
  - [Access Your VM Instance Using PuTTY \(Windows\)](#)
  - [Access Your VM Instance Using Horizon](#)
  - [Add More Users to Your VM Instance](#)
- [Manage Your VM Instances](#)
  - [Delete a VM Instance from Your Project](#)
  - [Delete a Volume from Your Project](#)
  - [Resize a VM Instance](#)
  - [Create a Snapshot](#)
  - [OpenStack Security Groups](#)
    - [Modify the Default Security Group](#)
    - [Create a new Security Group](#)
    - [Security Group CIDR Examples](#)
- [Overview - Birthright Information](#)
  - [Available VM Images & Configurations](#)
  - [Software & Hardware](#)
  - [Network & Storage](#)
- [Additional OpenStack Resources](#)
  - [Request Firewall Exception](#)

- [Run a Simple Web Server](#)
- [SSL - Let's Encrypt](#)
- [Install CPUID](#)
  - [CPUID Hypervisor Codes](#)
  - [CPUID Instance Codes](#)

## User Contributed Tutorials

- Note: Community contributed content not officially supported by CADES.
- [Launch a Docker Container](#)
- [Launch Shiny within Docker](#)
- [Eclipse in CADES HPC](#)
- [Allinea DDT in CADES HPC](#)

## SHPC Condo User Guide

- Prerequisites
- Getting Started
  - [Request Your SHPC Condo Allocation](#)
  - [Access Your SHPC Condo Allocation](#)
  - [Execute a Job on Your SHPC Allocation](#)
- SHPC Condo Configuration
- SHPC Condo LDAP Groups
- Troubleshooting

## SHPC Condo Tutorials

- [Execute a Job on Your SHPC Allocation](#)
- [Condo Workflows](#)
  - [Crystal Workflow](#)

## CADES Quick Summary (Cheat Sheet in pdf format)

- [Download CADES Quick Summary Cheat Sheet](#)

## CADES Policies

- [Birthright Cloud User Policy](#)

[CADES](#) → [User Documentation](#) → [Support](#)

## Getting Help

Within this documentation the CADES team and user community have assembled getting started guides, a glossary, and user-created tutorials to help you use resources within our environments.

Sometimes, though, docs are not enough. If you have questions not answered here, or would like to open a trouble ticket, please contact the CADES team directly at [cades-help@ornl.gov](mailto:cades-help@ornl.gov).

You may also join us at <http://cades.slack.com> where both CADES operations and community members share tips and can work more interactively together.

## Contributing

If you would like to add your own tutorials to this site, correct errors, or expand and clarify content, your contributions are welcome. Please see our [contributing guide](#).

## Ways to Contribute

Would you like to make things better? There are a few ways you can contribute to improving our documentation and adding user-created tutorials or content.

1. Email your suggestions to the team <mailto:cades-help@ornl.gov>
2. Join our community on Slack! It's friendly. <http://cades.slack.com>
3. Want to change things? Feeling adventurous? Want to git savvy?  
See [instructions for our git workflow](#) to branch our documentation repository and hack away. You got this.
4. We've made note of a few things to keep in mind while creating user content. You can find them in our [authoring guide](#).

## Recommended Workflow for Git and Atom

GitLab is a popular platform to share code, store software solutions, and host documentation.

ORNL provides two GitLab servers <https://code.ornl.gov> and <https://code-int.ornl.gov>, the later being accessible only inside of ORNL.

Access to GitLab repositories is controlled by project owners. You may login and create your own projects and repositories, and share them with others.

While there are many text editors to choose from, Atom is recommended due to its ability to be customized and integrated with GitLab/Git.

Install Atom: <https://atom.io/>

Would you prefer not to use Atom? Here is git documentation for [git in the command line](#).

### CADES User Documentation

Documentation published to CADES users is available within Gitlab at <https://code-int.ornl.gov/cades-ops/user-documentation> and users are encouraged to contribute to improving the material, or providing user created tutorials to share with the community.

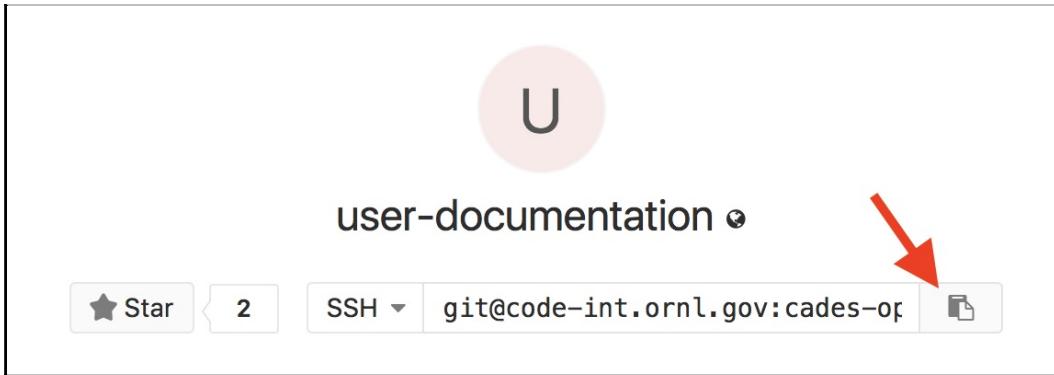
## Configuring Atom and Git

Atom has several packages which enhance the user experience and some of them you'll need for the workflow. The packages can be installed by entering the `Settings` screen and choosing `Install`. There are thousands of packages, so try some out and have fun.

1. Necessary Packages:
  - o `git-plus` (integrates Git)
  - o `todo-show` (provides visuals for TODO and FIXME tags)
2. Optional but Recommended Packages:
  - o `linter` (aids in code validation, will also need linter package for each language)
  - o `minimap` (displays thumbnail version of document along with your location within it)
  - o `git-checkout` (allows checking out remote branches within Atom)
3. Now that Atom is set-up, you'll need to make sure Git knows who you are. You'll only need to complete this step if you've never used Git on your machine before.
4. Open a terminal window.
5. If you need to install Git, see [here](#) for detailed instructions for popular operating systems.
6. To configure Git, input your user name and email as below:
  - o `git config --global user.name "abi"` (3-letter UCAMS username)
  - o `git config --global user.email "nameab@ornl.gov"` (ORNL email)

## Connecting Atom to a Repository

1. Navigate to the GitLab repository in your web browser. For this example, we'll use the `user-documentation` repository. Copy the SSH address to your machine's clipboard.



2. Now, back inside Atom, open the `Command Palette`. On Mac, press `shift + command + p`. In Windows/Linux, press `control + shift + p`.
3. Type `git clone` and press `enter`.
4. Paste the ssh address in the resulting window. You may also modify the location of the local folder.
5. Wait for the repository to clone.
6. Now you can see the files have populated into the folder you specified. These files represent a local copy, to which you will make changes.
7. Before you leave GitLab's webpage, consider adding your ssh key to your profile so you will not be prompted for credentials after every commit. To add your public ssh key to GitLab:
  - o Click on your user image in the top-right of the GitLab window.
  - o Select `Settings`.
  - o On the left, click `ssh keys`.
  - o Paste your **public** ssh key in the box, provide a title, and save by clicking `Add key`.

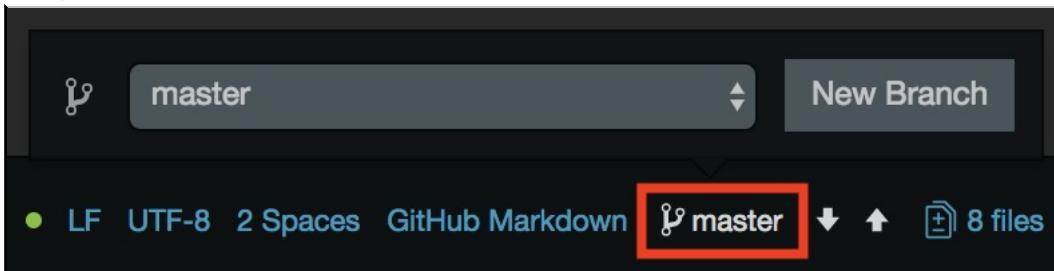
**Note:** You have now checked out the master branch of the remote repository. You may lack permission to push your changes to master, or may wish not to do so.

## Working from Branches

At this point you likely either want to create a new branch and add your contributions there, or checkout a different branch you or someone else has already created. Each of these options are shown next. Unless you have reason otherwise you should choose one of these, rather than attempting to work from the master branch.

### Create a New Branch

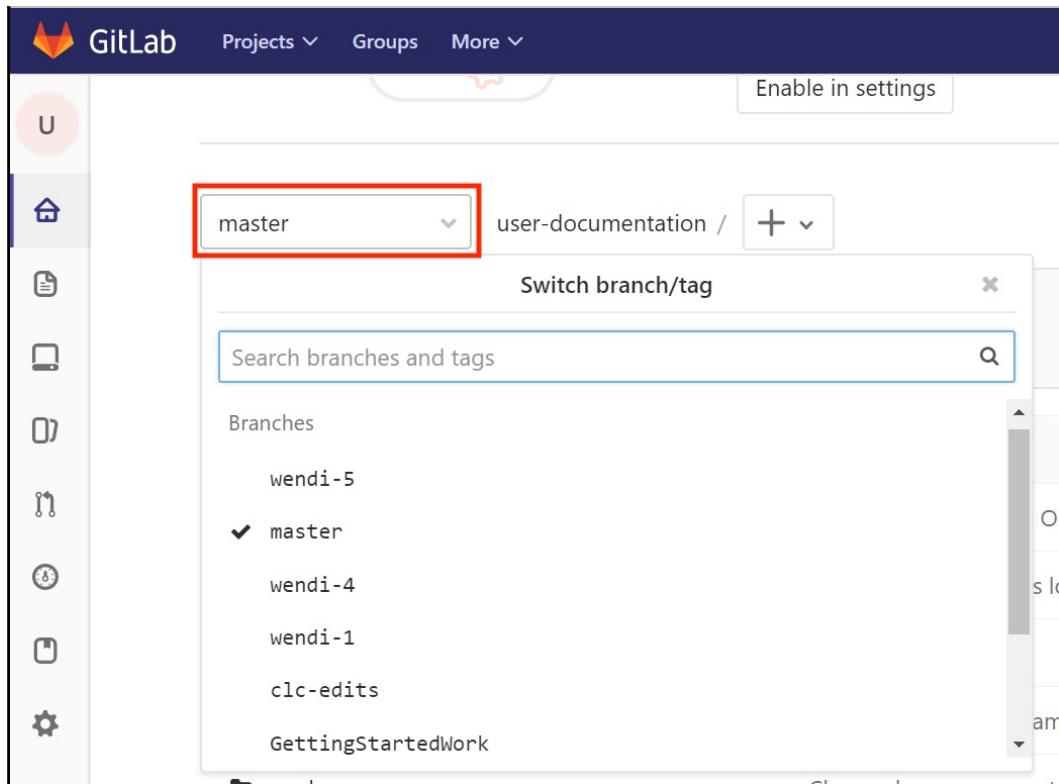
1. Create a git branch by clicking on the `master` button on the bottom-right of the Atom window. Name the branch something descriptive.



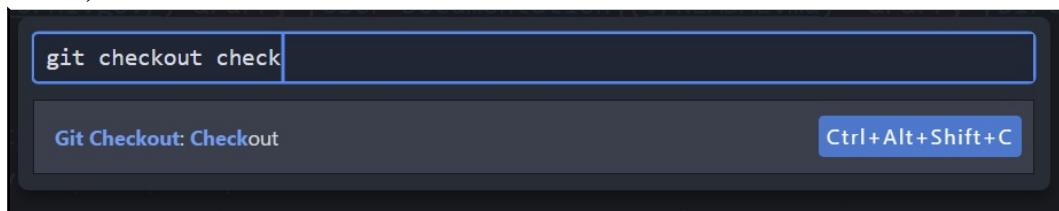
### Checkout an Existing Branch

We will need the name of the remote branch we wish to work on

1. The GitLab project page displays a dropdown with the name of available branches.

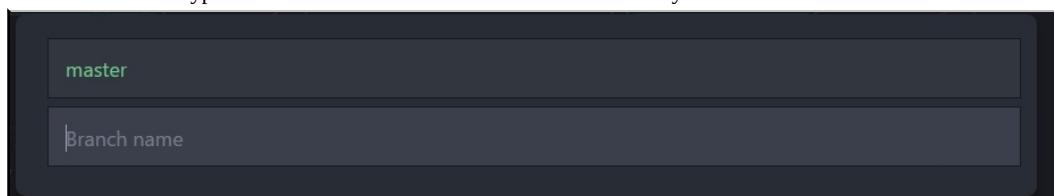


2. Open the Atom command palette and search for `git checkout checkout` (requires git checkout plugin having been installed).

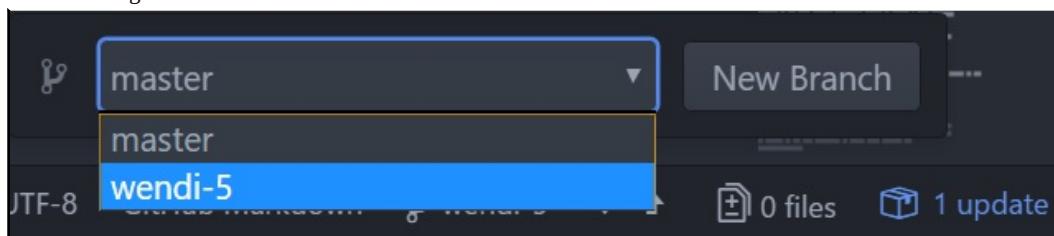


You may also open the checkout dialog directly using the hotkeys `ctrl + alt + shift + c` (or `ctrl + opt + shift + c` on Mac).

- The checkout dialog is a list of branches to checkout or switch to. Using the arrow keys, highlight the `custom` entry and hit enter. Now type in the name of the remote branch name. You may find this in GitLab.



- A notice will display if the checkout was successful. You may then switch between branches using the branch selector in the bottom-right toolbar.



**Note:** If checking out a remote branch within Atom using the 'git checkout' plugin, ensure you read the directions carefully. Do **not** click on the 'custom' branch, use the arrow keys and hit enter and then supply the name of the remote branch you wish to checkout. Lastly, if you make a mistake while typing in the branch name, you will end up creating a new branch with the typo. Be

sure to correctly type the branch name.

## Command Line Branching

Rather than using the checkout dialogs in Atom, you may also list and checkout remote branches using the `git` command line tools.

List remote branches:

```
$ git branch -r
origin/GettingStartedWork
origin/HEAD -> origin/master
origin/master
origin/wendi-5
origin/user-contributions
```

Checkout a remote branch from the command line. `git checkout --track origin/wendi-5` after which you may select the branch within Atom.

## GitLab GUI Editing

You don't have to use Atom for editing. You can hit the `edit` button in GitLab and edit directly, and preview before committing. Note that only repository owners can edit directly this way - otherwise you can create a branch and edit your branch directly.

## Uploading Your Changes

1. Make changes to the files as needed.
2. You can open the Git window by pressing `^ + .` Here you view unstaged and staged changes.
  - `Unstaged` means the files are not ready to be committed.
  - `Staged` means the files are ready to be committed.
3. Stage all of the files that you'd like to commit to the branch. This is accomplished by selecting the `+` symbol next to each file. Alternatively, on the top-right of the Git window, there is a `Stage All` button.
4. Commit your changes either within the Git window, or by entering `git commit` in Atom's command palette. Enter a commit message that will help you and others understand what changes were made. Then click `Commit`.
5. Push changes to GitLab by typing `git push` in the command palette, or using Atom's up/down Git arrows located on the bottom-right of the window.

**Note:** If you get an error after typing `git push` that says "No upstream branch" open your terminal and navigate to the local copy of the repository. Then type `git push --set-upstream origin name_of_branch`. From then on you should be able to use the command palette to type `git push` or use the up/down arrows on the bottom-right of the window.

## Creating a Merge Request

Merging your branch into the master branch, thereby makes your changes appear in the final version of the files.

You can send a merge request using the GitLab GUI.

1. From the left menu panel in Gitlab, select `Merge Request` then the green `New merge request` button.
2. Select your branch in the "Source Branch" side.
  - Target branch will be `master`
  - Click `compare branches`.
3. On the next screen the only thing needed is simply:
  - Assign to: <Project Owner, etc. >

- o Click `Submit merge request`

## Previewing Changes

When you push a branch, your changes will get built in a "Review Environment" at [http://user-documentation-stf011.granite.ccs.ornl.gov/\\_review/](http://user-documentation-stf011.granite.ccs.ornl.gov/_review/) and will allow you to click on the name of your branch.

A link to this review environment will be available on the Pull Request page that is created for your branch. This will allow us to take a quick glance at changes before we merge them into production.

**Note:** Preview sites are a prototype in the workflow. This feature may not always work and is offered as an unsupported convenience.

# Git Workflow from the Command Line

There are many reasons one would prefer to work from the command line. Regardless of your reasons, here is how to contribute to the CADES documentation using only command line tools.

This guide is adapted from [GitLab's documentation](#).

It is assumed that users of this guide understand basic git/version control principles. To read more, visit [this page](#).

## Install Git and Set-up

1. First, check to see if git is installed.

```
git --version
```

- To install and/or update git using your package manager:

- CentOS, RedHat:

```
sudo yum install git  
sudo yum update git
```

- Debian, Ubuntu:

```
sudo apt-get install git  
sudo apt-get update git
```

- MacOS, use [Homebrew](#):

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"  
brew install git  
brew upgrade git
```

- Windows: download [Git for Windows](#) and install it.

2. Set-up git with your access credentials to GitLab with the following commands:

```
git config --global user.name "your_username"  
git config --global user.email "your_email_address@example.com"
```

- You can review the information that you entered during set-up: `git config --global --list`

3. (*Optional*) Prior to cloning the repository, consider adding your ssh key to your GitLab profile so you will not be prompted for credentials after every commit. To add your public ssh key to GitLab:

- Click on your user image in the top-right of the GitLab window.
- Select `Settings`.
- On the left, click `ssh keys`.
- Paste your **public** ssh key in the box, provide a title, and save by clicking `Add key`.

## Using Repositories and Branches

1. Clone an existing repository.

```
git clone git@code-int.ornl.gov:caedes-ops/user-documentation.git
```

- o If you have already cloned the repository but are returning to your local version after a while, you'll want to make sure your local files are up to date with the branch. You can pull updates from `master` or from `branch_name`.

```
git pull origin branch_name
```

2. You're ready to make edits using your favorite text editor. You will need to create a new branch or checkout an existing branch that will later be merged into the master branch. When naming branches, try to chose something descriptive.

- o To create a branch: `git checkout -b branch_name`
- o To list existing branches: `git branch -r`
- o To checkout an existing branch: `git checkout --track origin/branch_name` or `git checkout branch_name`
  - Note: You may only have one branch checked out at a time.

3. When you are satisfied with your changes, commit them to your branch by adding and committing the changes.

```
git add --all  
git commit -m "descriptive text about your changes"
```

4. After committing the edits, you'll want to push the changes to GitLab. If the following produces an error, see below the code snippet for common solutions. The structure of this command is `git push <remote> <branch>`.

```
git push
```

- o Upstream error: `git push --set-upstream origin branch_name` or `git push -u origin branch_name`

## Creating a Merge Request

At this time, GitLab does not natively support submissions for merge requests via the command line.

You can send a merge request using the GitLab GUI.

1. From the left menu panel in Gitlab, select `Merge Request` then the green `New merge request` button.
2. Select your branch in the "Source Branch" side.
  - o Target branch will be `master`
  - o Click `compare branches`.
3. On the next screen the only thing needed is simply:
  - o Assign to: <*Project Owner, etc.*>
  - o Click `Submit merge request`

# Authoring Guide for CADES

Perhaps you've got some how-to documents tucked away in folders that you'd like to share with the CADES community. Or maybe you've discovered a way of doing things that would benefit other users.

You can submit your user guides for publication within the [CADES documentation site](#)! See the [contributing](#) page for instructions.

We've assembled here the fundamental authoring guidelines for CADES user documentation.

## Document and Content Preferences

- Documents should be created using [markdown](#).
- Oak Ridge National Laboratory (ORNL) uses the [\*Chicago Manual of Style\*](#) (CMOS) as a basic style guide.
- Define the first instance of every acronym in each document. Ensure that the long form is not repeated after it is defined.
- Buttons and links that the user should "click" should go in `code`. For example, "Next, click the `Manage Rules` button."
- Put `&#128221;` in front of NOTES. Renders:
- Use `&#9662;` for those "carrot" drop-down menus. Renders:
- For headings: only use title case for the first three heading levels, `#`, `##`, and `###`. The remaining header levels should be sentence case.

## Pictures and Images

Screenshots and images cannot be resized using markdown. Therefore, we embed `.html` that will be rendered when we publish the tutorial to the documentation site.

- Images and screenshots are stored in a folder `./screenshots/`.
- Files should be named descriptively. For example, use names such as `adding-IP-address.png` instead of `image03.png`.
- To remain consistent with other images in tutorials, please use the following `.html` code to resize, add a border, and open in a new browser tab when clicked. Note that you'll need to change the file name twice in the following code.

```
<a target="_new" href="screenshots/ssh_import_pub_key.png"></a>
```

## Other Considerations

- Have you redacted sensitive information from text and images?
- Have you removed information that is protected by copyright?
- Are you using a specific version of your software and have you included in the documentation?

## Related Topics

- Using a [Git Workflow](#) for creating user content.



## Glossary

The definitions below are related to this document and are provided for quick reference. A more complete list of definitions can be found in the official [OpenStack documentation](#).

### Bash

A UNIX shell used for entering command-line executions. Included with most Linux distributions and macOS. Includes SSH capability.

### Horizon

A web GUI front end for OpenStack that is accessed via <https://cloud.cades.ornl.gov>.

### Instance

A virtual machine set up through OpenStack. See "Virtual Machine".

### OpenStack

A cloud operating system that controls large pools of compute, storage, and networking resources throughout a data center.

### Project

The base unit of "ownership" in OpenStack. All resources in OpenStack should be owned by a specific project. In OpenStack Identity, a project must be owned by a specific domain.

### Tenant

A legacy OpenStack term for "Project" that is still used in the Horizon web GUI.

### Virtual Machine

An operating system instance that runs on top of a hypervisor. Multiple virtual machines (VM) can run at the same time on the same physical host.

# Getting Started with your Birthright Resources

Using CADES OpenStack resources is meant to be as straightforward as possible to get users up and running quickly. This guide provides walk-throughs guides and detailed information for you to get the most out of these resources.

## [What is OpenStack? What are Birthright Resources?](#)

This guide includes several sections that act both as a step-by-step guide and a quick reference.

In this and the following sections, you will discover how to:

1. [Request and Manage your Birthright Allocation](#)
2. [Launch a Virtual Machine \(VM\)](#)
3. [Access a VM](#)
4. [Manage a VM](#)
5. Learn about [Additional Resources](#)

Before you begin, be sure you meet the prerequisites below.

## Prerequisites

To properly utilize your CADES Birthright Cloud allocation, you will need a couple of utilities loaded on your local machine. These utilities are free and widely used for this type of application.

- Required: **SSH client**
- Recommended: **Bash terminal**

**Note:** CADES does not provide support for getting these utilities up and running on your personal computer.

## MacOS and Linux

Both macOS and Linux distributions includes a Bash terminal and an SSH client by default. No additional software should be required to access your VM Instance.

## Windows Users

► Click for Details

## Next Steps

Before you can use the Birthright resources, you will need to [request a cloud allocation](#). After your request has been approved, you can [manage your resources](#) and [launch a VM](#).

## Looking for More Information?

- [Overview of Birthright Resources](#)



CADES → User Documentation → Birthright Cloud User Guide → Getting Started → Request Cloud Allocation

## Request Your Birthright Cloud Allocation

Any member of a [science and technology directorate](#) can request a Birthright Cloud allocation. This is currently an automated process that takes 1–2 hours to complete. Use the steps below to enter your request.

1. Navigate to ORNL's [XCAMS portal](#). The preceding link will prefill a request for "CADES Birthright Cloud" resources.
2. Ensure your `UCAMS ID` is selected.
3. Enter a reason for the request (e.g., "I need a Birthright Cloud allocation.").
4. Click `Next` to complete the XCAMS request.

An activation notice will be dispatched to your ORNL email address when your resources are ready for use.

## Important Notes for Requesting Your Birthright Cloud Allocation

- Owing to resource constraints, we are currently throttling access to Birthright Cloud allocations. If during registration you get an error that the group is full, please [contact the CADES team](#).
- The registration process can take 1–2 hours. If after 2 hours you are unable to log in, please contact the [CADES team](#) and include the following information in your email:
  - UCAMS ID
  - Contact information
  - Approximate time of your registration attempt

# Manage Your Birthright Cloud Allocation

Once your request for resources has been approved, you can manage your allocation using OpenStack's web-based Horizon GUI.

1. Navigate to the Horizon web interface at <https://cloud.cades.ornl.gov/>.
2. Log in with your UCAMS credentials.
  - o Domain: `ornl`
  - o User Name: `UCAMS ID`
  - o Password: `UCAMS password`
3. Select `Project` from the top left menu to view available resources.

The screenshot shows the 'Project' dashboard of the RED HAT OPENSTACK PLATFORM. At the top, there are six circular pie charts representing resource usage: Instances (Used 1 of 4), VCPUs (Used 4 of 4), RAM (Used 8,192 of 8,192), Floating IPs (Used 0 of 50), Security Groups (Used 2 of 10), and Volumes (Used 2 of 4). Below these are two sections: 'Limit Summary' and 'Usage Summary'. In 'Limit Summary', there are two more pie charts: Volume Storage (Used 16 of 40) and another unlabeled one. Under 'Usage Summary', a table lists active instances: `kelantest` with 4 VCPUs, 80GB Disk, 8GB RAM, and a creation time of 2 weeks, 5 days. A 'Download CSV Summary' button is also present.

From here you can easily manage your allocation through a variety of tools that enable you to:

- o [View Your OpenStack Project Quota](#)
- o [Manage VM Instances](#)
- o [Manage OpenStack Security Groups](#)
- o [Configure SSH Access to VM Instances](#)

## Your OpenStack Project Quota

Each CADES Birthright Cloud allocation creates a "Project". Each Project in CADES has a preset resource quota that can be leveraged at the user's discretion.

## Request More Resources

If a user requires more resources for their allocation, he or she can submit a proposal to the CADES Resource Utilization Council (RUC) to request a quota increase. This proposal should describe the resources desired (RAM, CPUs, storage, etc.) and the scientific goal and merit of the work being performed using the CADES Birthright Cloud allocation. These requests for increased resources, subject to review, should be directed to the [CADES team](#).

## View Your Project Quota

1. Navigate to the Horizon web interface at <https://cloud.cades.ornl.gov/>.
2. Log in with your UCAMS credentials.
  - Domain: `ornl`
  - User Name: `Your three-letter UCAMS ID`
  - Password: `Your UCAMS password`
3. Navigate to `Project` → `Overview`.

In this Overview you can see your resource allocation, including:

- **Instances** – The number of VMs you can run at once.
- **VCPUS** – The number of CPU cores you can use across all of your VMs.
- **RAM** – The total amount of RAM you can use across all of your VMs.
- **Floating IPs** – COMING SOON – IP addresses that you can attach and move between Instances. These addresses are in addition to the IP(s) already allocated to your Instance.
- **Security Groups** – Blocks of firewall rules that you can attach to an Instance. All groups start with a default Security Group that contains a basic configuration to get started. You can use this "default" group or you can create your own group. Check out the [Security Groups section](#) for more information.
- **Volumes** – These are the number of block-storage volumes that you can create. You can attach any number of volumes to an Instance. These are also the preferred method of storage in the Birthright cloud environment.
- **Volume Storage** – This is the total storage available to the volumes in your Project.

Once an Instance is launched, the resource utilization appears at the bottom of Overview screen, under the Usage section.

## Getting Help

In addition to OpenStack's documentation, the CADES team has assembled a list of [frequently asked questions](#), a [glossary](#), and detailed information to help you get moving on your OpenStack Project.

If you have questions that were not answered in the resources listed here or would like to open a trouble ticket, please contact the CADES team directly at [cades-help@ornl.gov](mailto:cades-help@ornl.gov).

If you have questions about getting your SAFER firewall rules in place, email the SAFER team directly at [opsapprovers@ornl.gov](mailto:opsapprovers@ornl.gov).

## General OpenStack Usage Questions

The OpenStack community has extensive documentation, a general mailing list, and a mechanism for asking general OpenStack usage questions. These resources can be indispensable for new users, and the CADES support team recommends that new users leverage the [OpenStack documentation](#) and other support features for general OpenStack questions.

Information	Description
<a href="#">OpenStack documentation</a>	OpenStack maintains significant documentation for general OpenStack usage. The <a href="#">user manuals</a> cover OpenStack's functionality in depth, although not all of the features listed in their documentation are available from CADES.
<a href="http://ask.openstack.org">ask.openstack.org</a>	General end-user resource with questions and answers.
<a href="#">OpenStack mailing list</a>	General questions mailing list, OpenStack operations mailing list, and a developer mailing list.

## Frequently Asked Questions

- *Who can request a CADES Birthright Cloud allocation? Can I have one?*
  - Any member of a science and technology directorate can request a Birthright Cloud allocation. See [list of ORNL's technical organizations](#).
- *What kind of user support does CADES provide for Birthright Cloud allocations?*
  - The CADES team can provide support for issues in OpenStack and can help a user get started with their allocation—up to and including loading and launching a VM Instance from one of the CADES-provided images. Once a user begins adding software and configuring the operating system on their VM, they are more-or-less responsible for any issues that might present themselves in the VM.
- *Does CADES provide backups or support for backing up my VM Instance(s)?*
  - In short, no. CADES does not provide backups of a user's VM Instances and does not currently support this functionality. There are several methods a user could leverage to back up the operating system running on the VM, but any such method must comply with ORNL's best practices, and this backup strategy is the user's responsibility and is beyond the scope of this user guide.
- *Can I have my instance IPs be sequential?*
  - Owing to the dynamic nature of a cloud environment this is not possible. We recommend using `/etc/hosts`, Netreg (DNS), or another mechanism to accomplish host-to-host communication.
- *How do I get access to Lustre on my instance?*

- Lustre access is currently provided on a case-by-case basis. [Contact the CADES team](#) if you wish to add Lustre to your Project.
- *What ORNL file systems can I mount within my Birthright Cloud Instance?*
  - The target file system must be "open science." No moderate/confidential file systems or their respective data can or should be mounted within your VM Instance.

## Launch a VM: At-A-Glance

Creating and launching a VM Instance is one of the first steps to utilizing your CADES Birthright Cloud allocation. When launching a VM, you will choose a name for your Instance, which network it will utilize (internal or external), which operating system you would like to run (CentOS or Ubuntu), and which VM flavor you need for your application.

**Note:** When launching a new VM Instance, be aware that your VM Instance name *may* also serve as your DNS host name, *if you choose to register your VM with NetReg*. This name can contain up to 20 alphanumeric characters and a hyphen. **No** special characters are permitted, and the host names are case sensitive. For example, good: `my-instance-server`, no good: `my_instance&server`. See [RFC 952](#) and [RFC 1123](#) for more information.

## Prerequisites

You will need to have a CADES Birthright Cloud allocation before you can launch your VM Instance using the web GUI. The link below will show you how to request your Birthright Cloud allocation.

How to: [Request Your CADES Birthright Cloud Allocation](#)

There are three primary steps to complete to have a functioning VM Instance. These steps are outlined below. After each section a link is provided where you can find more information and in-depth descriptions about the configuration options.

1. [Log in to Horizon, name your VM.](#)
2. [Choose a flavor, image, and boot source.](#)
3. [Set up a security group.](#)
4. [Configure a key pair for accessing the VM.](#)

## Launch a VM: Log In & Naming

Once you receive the email notification that your resources are available, you can login to the Horizon web interface and get started.

## Log In to Horizon

- Navigate to the web interface at <https://cloud.cades.ornl.gov/>.
- Log in with your UCAMS credentials.
  - Domain: `ornl`
  - User Name: `Your three-letter UCAMS ID`
  - Password: `Your UCAMS password`

We can now launch a new VM Instance from within the web interface.

## Launch an Instance

1. Navigate to `Project` → `Compute` → `Instances`.
2. Click the `Launch Instance` button, and fill out each section of the resulting dialog.

Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
No items to display.										

### Details Tab – Fill out the required fields for the Details section.

- `Instance Name` – Your VM Instance name also serves as your DNS hostname. This name can contain up to 15 alphanumeric characters and a hyphen. No special characters are permitted, and the host names are case sensitive. For example, good: `my-instance`, no good: `my_instance&server`.
- `Availability Zone` – "nova" is the default zone. **Do not change unless instructed to do so by a CADES administrator.**
- `Count` – The number of instances to start up at once. If using multiple Instances, the Instance names will be numerated (`instance-1`, `instance-2`, `instance-3`, and so on).

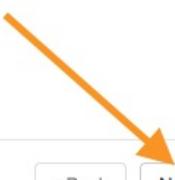
### Launch Instance

Please provide the initial hostname for the instance, the availability zone where it will be deployed, and the instance count. Increase the Count to create multiple instances with the same settings.

**Details**

Source \*  
Flavor \*  
Networks \*  
Network Ports  
Security Groups  
Key Pair  
Configuration  
Metadata

Instance Name \*  Total Instances (4 Max)   
Availability Zone   
Count \*   
0 Current Usage  
1 Added  
3 Remaining



< Back **Next >** Launch Instance

- Once finished, click **Next >** to proceed to the next tab.

## Launch a VM Instance: Flavor and Image

On the next two tabs, you will choose the storage type and size, and the operating system image.

### Source Tab – Fill out the required fields for the Source tab.

- **Select Boot Source** – Set this to `Image`
- **Create New Volume** – Set this to `Yes`. This creates a virtual disk on CADES's central storage (recommended method). If set to `No`, a virtual disk is created on a hypervisor; this disk is not persistent when the VM is deleted (an ephemeral instance).
- **Delete Volume on Instance Delete** – Set to `No` if data should persist between Instance restarts.
- **Volume Size** – Must be equal to or greater than the flavor size and fit within your [allocation quota](#). Flavor size is listed in the [Field tab](#) (you can move between these tabs without losing data).
- **Device Name** – This should almost always be the default, `vda`.
- **Allocated** – Images allocated to the Instance. Choose from the available options by clicking the `+` next to the desired image.

We're choosing the Ubuntu image for this example.

Name	Updated	Size	Type	Visibility
CADES_CentOS-7.3_v2017031_0_0	3/14/17 3:04 PM	1.43 GB	QCOW2	Public
CADES_Ubuntu16.04_v2017011_1_0	1/12/17 9:33 PM	1.83 GB	QCOW2	Public

- Upon selecting an image, it will move from the Available list to the Allocated list.

**Launch Instance**

Details

Instance source is the template used to create an instance. You can use a snapshot of an existing instance, an image or a volume (if enabled). You can also choose to use persistent storage by creating a new volume.

**Source**

Flavor \*

Networks \*

Network Ports

Security Groups

Key Pair

Configuration

Metadata

Select Boot Source

Image

Create New Volume

Yes No

Delete Volume on Instance Delete

Yes No

Device Name

vda

Name	Updated	Size	Type	Visibility	
CADES_Ubuntu16.04_v2017011 1.0	1/12/17 9:33 PM	1.83 GB	QCOW2	Public	-

Available 1 Select one

Click here for filters.

Name ^	Updated	Size	Type	Visibility	
CADES_CentOS-7.3_v2017031 0.0	3/14/17 3:04 PM	1.43 GB	QCOW2	Public	+

Cancel Next > Launch Instance

- Once finished, click **Next** to proceed to the next section.

### Flavor Tab – Choose an image flavor for your VM Instance.

- Available – List of available images. Choose an image, and click + to add it to your allocation.

The screenshot shows the 'Launch Instance' wizard. The current step is 'Flavor \*'. The interface includes sections for 'Details \*', 'Allocated', 'Source \*', and 'Flavor \*'. The 'Flavor \*' section has a dropdown menu showing 'Available 5' items. A search bar and a filter link are also present. Below the dropdown is a table with columns: Name, VCPUS, RAM, Total Disk, and Public. The table lists five flavors: m1.tiny, m1.small, m1.medium, m1.large, and m1.xlarge. Each row has a '+' button to the right. An orange arrow points to the '+' button for the m1.tiny row. At the bottom are buttons for 'Cancel', 'Back', 'Next >', and a large blue 'Launch Instance' button.

Name	VCPUS	RAM	Total Disk	Public
m1.tiny	1	512 MB	8 GB	Yes
m1.small	1	2 GB	20 GB	Yes
m1.medium	2	4 GB	40 GB	Yes
m1.large	4	8 GB	80 GB	Yes
m1.xlarge	8	16 GB	160 GB	Yes

**Note:** The indicates that your quota has inadequate resources for the image.

- Upon selection, the chosen image will move to the `Allocated` list.

Launch Instance X

Details \* Flavors manage the sizing for the compute, memory and storage capacity of the instance ?

Source \* Allocated

Name	VCPUS	RAM	Total Disk	Public	
m1.tiny	1	512 MB	8 GB	Yes	-

Networks \* Networks \* Available 4 Select one

Network Ports

Security Groups

Key Pair

Configuration

Metadata

Name	VCPUS	RAM ^	Total Disk	Public	
m1.small	1	2 GB	20 GB	Yes	+
m1.medium	2	4 GB	40 GB	Yes	+
m1.large	4	8 GB	80 GB	Yes	+
m1.xlarge	8	16 GB	160 GB	Yes	+

Cancel Back Next > Launch Instance



- Once finished, click **Next** to proceed to the next section.

# Launch a VM: Networks and Security

## Networks Tab – Choose a network for your VM Instance.

**Note:** Contact the CADES team if you require more than one IP from each Network. Additionally, if your needs are not met by following this guide, feel free to [email the CADES team](#) to discuss options.

- **Available** – List of available networks. Choose one of two routable network configurations, and click **+** to add it to your allocation.
  - `general_extnetwork1`, 128.219.184.0/21 - Available from outside ORNL. However, outward-facing services (e.g., SSH, web server) will require ORNL firewall exceptions ([instructions](#)).
  - `general_intnetwork1`, 172.22.0.0/20 - Internal to ORNL.
- For this example, we will choose the external network (`general_extnetwork1`).

Network	Subnets Associated	Shared	Admin State	Status
<code>or_provider_general_extnetwork1</code>	<code>or_provider_general_extsubnet1</code>	Yes	Up	Active
<code>or_provider_general_intnetwork1</code>	<code>or_provider_general_intsubnet1</code>	Yes	Up	Active

- **Allocated** – Upon selection, the chosen network will move to the Allocated list.

Network	Subnets Associated	Shared	Admin State	Status
<code>or_provider_general_extnetwork1</code>	<code>or_provider_general_extsubnet1</code>	Yes	Up	Active
<code>or_provider_general_intnetwork1</code>	<code>or_provider_general_intsubnet1</code>	Yes	Up	Active

- Once finished, click **Next** to proceed to the next section.

## Network Ports Tab – No user input required. Skip this step.

## Security Groups Tab - Choose a security group for your VM Instance.

**Note:** Skipping this step will make your VM Instance unreachable! [See additional documentation on Security Groups.](#)

- Available – List of available security groups. Choose the desired Security group, and click + to add it to your allocation. The default Security Group, used for this example, has the basic services you need to get started. Users can also [create their own custom Security Groups.](#)

**Launch Instance**

**Details \*** Select the security groups to launch the instance in.

**Source \*** Allocated

**Flavor** Select one or more security groups from the available groups below.

**Networks**

**Network Ports**

**Security Groups**

Name	Description
Custom	Custom security group
default	Default security group

**Cancel** **Next >** **Launch Instance**

- Allocated – Upon selection, the chosen Security Group(s) will move to the Allocated list.

**Launch Instance**

**Details \*** Select the security groups to launch the instance in.

**Source \*** Allocated 1

**Flavor**

**Networks**

**Network Ports**

**Security Groups**

Name	Description
default	

**Available** 1 Select one or more

Name	Description
Custom	Custom security group

**Cancel** **< Back** **Next >** **Launch Instance**

- Once finished, click `Next` to proceed to the next section.

## Launch a VM: Key Pair Use

### Key Pair Tab – An SSH key pair is required to access your VM Instance.

You can create a new key pair for this Instance ([Option 1](#)) or choose a key pair from the Available list ([Option 2](#)).

**Note:** Before deciding, you should check your local machine for an existing key pair ([instructions](#)). If you skip this step, the instance will not allow you to log in!

See the [Access your VM Instance Using SSH documentation](#) for more information.

- **Option 1: Create a new key pair for this instance.**

- Click the `+ Create Key Pair` button.

The screenshot shows the 'Launch Instance' dialog with the 'Key Pair' tab selected. On the left, there's a sidebar with tabs: Details, Source, Flavor, Networks, Network Ports, Security Groups, Configuration, and Metadata. The 'Source' tab is highlighted. In the main area, there's a note about key pairs and two buttons: '+ Create Key Pair' (highlighted by an orange arrow) and 'Import Key Pair'. Below that is a section for selecting an existing key pair, which is currently empty. At the bottom are 'Cancel', 'Next >', and 'Launch Instance' buttons.

- Enter a name for your new key pair in the resulting dialog.
- Click `Create Key Pair` to associate this new key pair to your Instance.

The screenshot shows the 'Create Key Pair' dialog. It has a note about key pair naming rules. There's a 'Key Pair Name \*' field containing 'blackmesakey' (highlighted by an orange box). At the bottom are 'Cancel' and 'Create Keypair' buttons, with an orange arrow pointing to the 'Create Keypair' button.

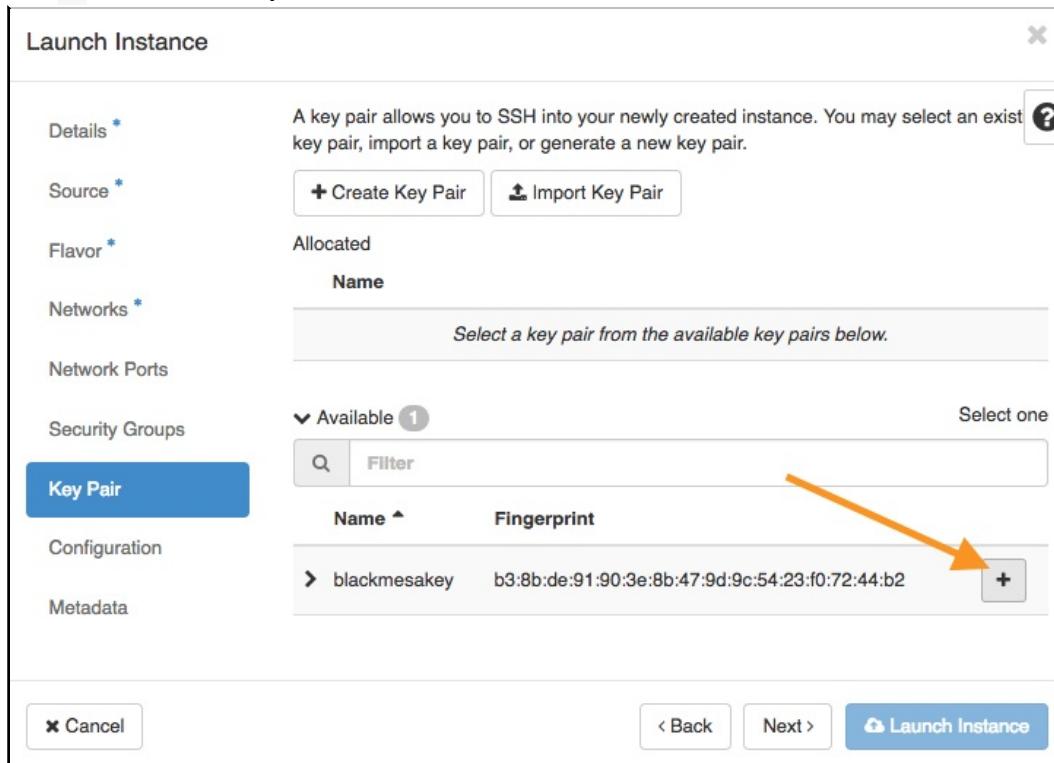
- The private key will be downloaded to your local machine as a `.pem` file.
- On your local machine, place the `.pem` file in the `~/.ssh/` directory ([instructions](#)).
- Once finished, click `Next` to proceed to the next tab.
- **Option 2: Use available key pair for this Instance.**

- Available – List of available key pairs that were previously generated or imported. Choose the desired key pair, and click + to associate it with your VM Instance.

**Launch Instance**

Details *	A key pair allows you to SSH into your newly created instance. You may select an existing key pair, import a key pair, or generate a new key pair.
Source *	<a href="#">+ Create Key Pair</a> <a href="#">Import Key Pair</a>
Flavor *	Allocated
Networks *	Name <i>Select a key pair from the available key pairs below.</i>
Network Ports	
Security Groups	
<b>Key Pair</b>	<b>Available 1</b> Select one <input type="text" value="blackmesakey"/> <a href="#">+</a>
Configuration	
Metadata	

**Cancel** [Back](#) [Next >](#) [Launch Instance](#)

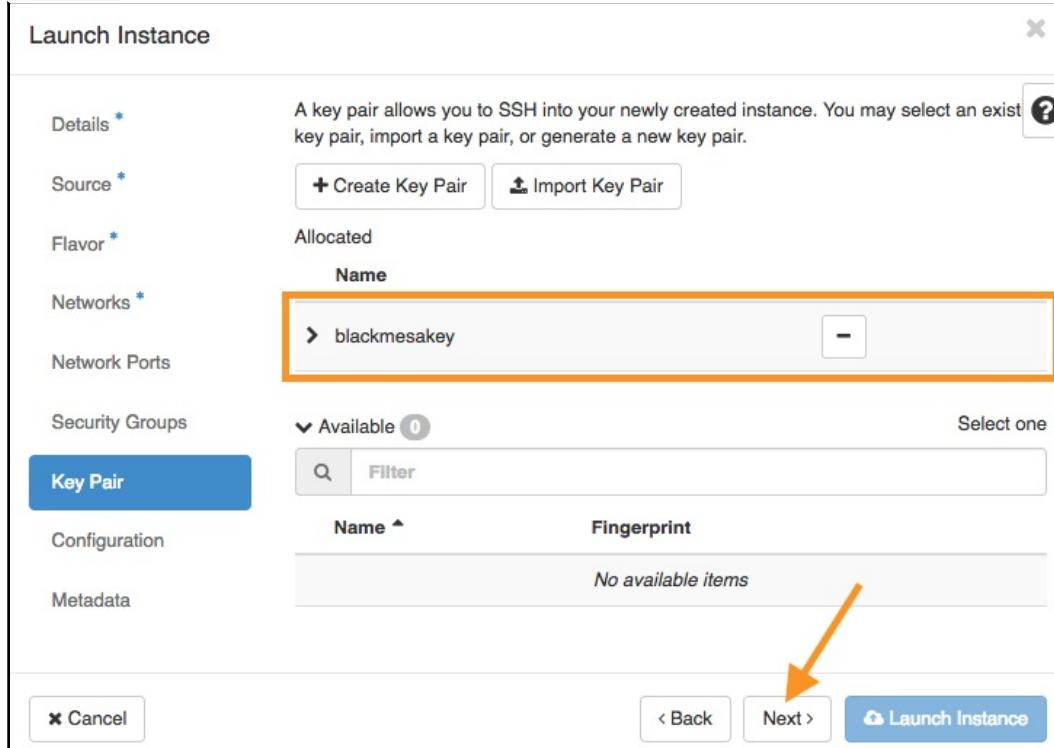


- Allocated – Upon selection, the chosen key pair will move to the Allocated list.

**Launch Instance**

Details *	A key pair allows you to SSH into your newly created instance. You may select an existing key pair, import a key pair, or generate a new key pair.
Source *	<a href="#">+ Create Key Pair</a> <a href="#">Import Key Pair</a>
Flavor *	Allocated
Networks *	Name blackmesakey <a href="#">-</a>
Network Ports	
Security Groups	
<b>Key Pair</b>	<b>Available 0</b> Select one <input type="text"/> <a href="#">Filter</a>
Configuration	
Metadata	

**Cancel** [Back](#) [Next >](#) [Launch Instance](#)



- Once finished, click Next to proceed to the next tab.

**Configuration Tab – This section is not required for deployment and is not currently supported by the CADES team.**

## **Metadata Tab – No user input required. Skip this step.**

Click `Launch Instance` when you have completed all required sections. *Congratulations!* A new instance will be launched. Once fully provisioned, the status will change to "Running," and you can access your VM Instance using SSH ([instructions](#)).

## Access VM Instances Running in OpenStack

There are several ways you can access your VM Instances.

1. The best way to access your VM Instance is [through an SSH connection](#).
  - If you use Windows: [Access Your VM Instance Using PuTTY](#)
2. You can also [access the VM Instance's terminal using Horizon](#).
  - But you must first [create a new user for this purpose using SSH](#).

**Note:** UCAMS credentials and SSH key pairs are unrelated, unconnected authentication methods. A user will only be prompted for a UCAMS username and password when he or she logs in to Horizon. All other authentication relies on SSH key pairs or, in the case of creating your own non-UCAMS users, a generic user-created password.

**Note:** To use the Horizon console, you must first [add a user and password](#) to the Instance's operating system (via SSH) to enable access to the VM without an SSH key.

CADES → User Documentation → Birthright Cloud User Guide → Access Your VM Instance → Access Your VM Instance Using SSH

## Access Your VM Instance Using SSH

Once you have created a VM Instance using Horizon, you can access this VM Instance through a secure shell (SSH) using an SSH key pair. The following tutorial will walk you through connecting to your VM Instance through SSH using Linux or macOS.

If you have several users that require access to a single instance, you may request they extract and provide you with their public key, which you may add to your instance's `/home/cades/.ssh/authorized_keys` file. Doing so grants access, as the user `cades`, to that instance. If desired, you may [create specific local accounts](#), with limited sudo roles, etc. rather than use the `cades` user account.

Windows users will have to install an SSH client on their machine, and the CADES team recommends PuTTY for this purpose. The CADES team has compiled a separate tutorial for Windows/PuTTY users, linked below.

How to: [Access VM Instances Using PuTTY \(Windows\)](#)

If you would like to access your VM Instance over SSH from *outside* of the ORNL network, you must first [create a firewall \(SAFER\) exception](#). All local (within the ORNL network) SSH connections are permitted by default. Request a SAFER exception with source 'VISITORS' and your VM(s) as destination.

## Table of Contents

- [Add SSH Keys to an Instance](#)
  - [Check for Existing Key Pairs](#)
  - [Generate a New Key Pair](#)
  - [Import Existing Key](#)
- [Connect to Your VM Instance Using SSH](#)
  - [Find Your Instance's IP](#)
  - [Connect to Your Instance](#)
- [Extract Public Key](#)
- [Related Tutorials](#)

## Add SSH Keys to an Instance

The first step to adding a key pair is actually to check for existing key pairs on your local machine. If no key pairs exist, then you can [generate a new key](#) in the Horizon web GUI.

If you do have an existing key pair on your machine, then you can [import the public key](#) using the Horizon web GUI. Alternatively, you can create a new (additional) key for your Instance and use a `config` file to manage your SSH credentials.

## Check for Existing Key Pairs

1. Open a Bash terminal.
2. Execute `ls -al ~/.ssh .`
3. Check the results of the directory listing.

If the directory listing is empty or the directory is not found, then you do not have an existing SSH key and you should follow the procedure outlined in [Generate a New Key Pair](#).

If the directory contains one of the files listed below, then you *do* have an existing SSH key, and you can import the public key into your Instance using the Horizon Web GUI by following the procedure outlined in [Import Existing Key](#).

- id\_rsa.pub
- d\_dsa.pub
- id\_ecdsa.pub
- id\_ed25519.pub

## Generate a New Key Pair

### Generate the keys

1. Navigate to the Horizon web interface at <https://cloud.cades.ornl.gov/>.
2. Log in with your UCAMS credentials.
  - Domain: ornl
  - User Name: Your three-letter UCAMS ID
  - Password: Your UCAMS password
3. Navigate to Project → Compute → Access & Security → Key Pairs .
4. In the Key Pairs screen, click the + Create New Pair button.

5. In the resulting dialog, name your Key Pair, and click Create Key Pair when you're finished.

6. The private key will be downloaded to your local machine as a .pem file. The public key will be stored in OpenStack.
7. Place the downloaded private key in ~/.ssh/ , e.g., ~/.ssh/blackmesakey.pem .
8. Secure the private key by setting the permissions to 600 in a Bash terminal.  

```
$ chmod 600 ~/.ssh/blackmesakey.pem .
```
9. Copy (from Horizon [Option A]) or generate (Bash terminal [Option B]) the public key and place in ~/.ssh/ , e.g., ~/.ssh/blackmesakey.pub .
  - **Copy public key from Horizon (Option A)**

- Navigate to Project → Compute → Access & Security → Key Pairs
- In the Key Pairs screen, click on the Key Pair you just created ( blackmesakey ).

The screenshot shows the 'Key Pairs' tab selected in the top navigation bar. A red arrow points to the 'Key Pair Name' column, highlighting the entry 'blackmesakey'. The 'Fingerprint' column shows the value 'b3:8b:de:91:90:3e:8b:47:9d:9c:54:23:f0:72:44:b2'. On the right, there are 'Actions' buttons for 'Delete Key Pair'.

- In the resulting screen, copy the entire public key (outlined in orange below).

The screenshot shows the 'Compute' tab selected in the top navigation bar. The 'Access & Security' tab is selected in the sub-navigation. The 'Public Key' section is highlighted with an orange box, displaying the following text:

```

Name: blackmesakey
Fingerprint: b3:8b:de:91:90:3e:8b:47:9d:9c:54:23:f0:72:44:b2
Created: March 24, 2017, 12:43 p.m.
User ID: 2f7d9c71fb69394e3720eac17afbd2bc66687d5f31b8500ae5288cece7d90cc
Public Key: ssh-rsa
AAAAB3NzaC1yc2EAAAQABAAQDmj... (long string of characters)
Generated-by: Nova
  
```

- Using a text editor, create a new file (e.g., `blackmesakey.pub`) and paste the public key into this file.
- Save/move this new file to `~/.ssh/blackmesakey.pub`.
  - **Generate public key using a Bash terminal (Option B)**
  - Open a Bash terminal.
  - Ensure your private key is in `~/.ssh/`.
  - Use `ssh-keygen` to generate your public key:

```
$ ssh-keygen -y -f ~/.ssh/blackmesakey.pem > ~/.ssh/blackmesakey.pub .
```

10. Ensure that both your public and your private keys are in `~/.ssh/`.

**Note:** Having completed the procedure above, you can now connect to your Instance via SSH using a long-form login ([instructions](#)) or you can take these newly generated keys and place them in your local system's default files ([instructions](#)).

## Place keys on local machine

If you prefer to leave the Key Pair as it is—as documented above—you can skip the following steps and use a long-form SSH login that specifies the public key's location and file name for each log in attempt. [Click here for instructions](#).

However, if you wish to log in without specifying the key location with each authentication attempt, you can put your public and private keys in their default location on your local system.

### Key locations for Linux and macOS:

- Private key should be `~/.ssh/id_rsa`.
- Public key should be `~/.ssh/id_rsa.pub`.

**Note:** Since we are generating new keys, these files (and the `/.ssh/` directory) may not exist on your local machine.

## Place the Private key

1. Create a new file using a text editor.
2. Copy your private key from your `.pem` file and paste it into this new file.

3. Save the new file as `id_rsa` and place it in `~/.ssh/`.
  - o Create the `/.ssh/` directory within `~/` if necessary.
4. Set permissions on this new file using `$ chmod 600 ~/.ssh/id_rsa`.

## Place Public key

1. Create a new file using a text editor.
2. Copy your public key from the Horizon web GUI ([instructions](#)).
3. Save the new file as `id_rsa.pub` and place it in `~/.ssh/`.

## Import Existing Key

If you have an existing key pair that you would like to also use for your OpenStack Project, then you can import the public key using the Horizon web GUI.

1. Navigate to the Horizon web interface at <https://cloud.cades.ornl.gov/>.
2. Log in with your UCAMS credentials.
  - o Domain: `ornl`
  - o User Name: Your three-letter UCAMS ID
  - o Password: Your UCAMS password
3. Navigate to `Project` → `Compute` → `Access & Security` → `Key Pairs`.
4. In the Key Pairs screen, click the `Import Key Pair` button.

The screenshot shows the 'Key Pairs' tab selected in the navigation bar. Below the navigation bar are buttons for 'Filter' and 'Actions'. There are two main sections: 'Key Pair Name' and 'Fingerprint'. Under 'Key Pair Name', there is a table with one row containing 'blackmesakey'. Under 'Fingerprint', there is a table with one row containing 'b3:8b:de:91:90:3e:8b:47:9d:9c:54:23:f0:72:44:b2'. To the right of the tables is a red 'Delete Key Pairs' button. At the bottom right of the table area is a red 'Delete Key Pair' button. An orange arrow points to the 'Import Key Pair' button, which is located at the top right of the table area.

5. In the resulting dialog, enter a key pair name in the `Key Pair Name` field.
6. Copy your public key from the file (e.g., `~/.ssh/id_rsa.pub`) on your local machine.
7. Paste the public key in the `Public Key` text box.
8. Click the `Import Key Pair` button.

**Import Key Pair**

**Key Pair Name \***  
blackmesakey

**Public Key \***

```
ssh-rsa
AAAAB3NzaC1yc2EAAAQABAAQDKV3
oXnjn+GJis2v8PoFB6sSz9Dwt1hBeFVFFy'sM6H
HgTTsZlFFFoGrfjBHctL2FiPyqG2cG36gHOgph8wj
3rY88jQBUq5uat6z88jDJ1KBZFLONMvH3Wi0+qN
nkwlrr9pHdlvuKP5x5xswFwk+zmtPevabNIS-I+1
6Li2Rak9vdISOvBzXkCaaP5CbzipahUsYVP4T3K
9i43w4Q+HIZvUIMPwgzM+68T7MOOPQ2/bNgci
qmqYNUOTzPloS7u1e3Vj6uxWBHdvFhzx1u5D
SxFsQdHjFtcdkpQdDpJZfWONXHTpgJs5/yeifQbC
-----BEGIN RSA PRIVATE KEY-----
```

**Description:**

Key Pairs are how you login to your instance after it is launched.

Choose a key pair name you will recognise and paste your SSH public key into the space provided.

SSH key pairs can be generated with the ssh-keygen command:

```
ssh-keygen -t rsa -f cloud.key
```

This generates a pair of keys: a key you keep private (cloud.key) and a public key (cloud.key.pub). Paste the contents of the public key file here.

After launching an instance, you login using the private key (the username might be different depending on the image you launched):

```
ssh -i cloud.key <username>@<instance_ip>
```

**Cancel** **Import Key Pair**

9. Confirm your newly imported key appears in the Key Pair list.

## Connect to Your VM Instance Using SSH

All VM Instances have SSH access enabled by default. Use your key pair and the user name "cades" for authentication. SSH to your Instance using the Key Pair you configured above.

## Find Your Instance's IP

The IP address of your VM instance is shown in the Instances tab in the Horizon web GUI.

1. Navigate to Project → Compute → Instances .
2. From the Instances screen, you can copy the IP address for the desired VM Instance.

Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
black-mesa	m1.tiny	128.219.186.19	blackmesakey	Active	nova	None	Running	1 week, 3 days	Create Snapshot	

Displaying 1 item

## Connect to Your Instance

Once you have determined the IP address of your VM Instance and placed your keys in `~/.ssh/id_rsa` and `~/.ssh/id_rsa.pub`, you can connect using a simple SSH command where "cades" is always the user.

## Standard login

1. Open a Bash terminal.
2. Execute `ssh cades@128.219.186.19`.
  - Replace `128.219.186.19` with the IP address of your own Instance.
3. You should now be connected to your VM Instance via SSH.
  - Use the Bash terminal to install your software-stack and perform work.
  - You can also add user credentials to your VM Instance to grant access to other users.

**Note:** In a newly created Instance, "cades" is the only user name that will correctly authenticate over SSH (using an SSH key pair). A user can add more users, and add public keys for each, once logged in as "cades" ([instructions](#)).

## Long-form login

If you prefer to specify your key location during login (e.g., you downloaded your key from the Horizon web GUI), you can use a long-form SSH login.

1. Open a Bash terminal.
2. Execute `$ ssh -i ~/.ssh/blackmesakey.pem cades@128.219.186.19`.
  - Replace `~/.ssh/blackmesakey.pem` with the file path of your key.
  - Replace `128.219.186.19` with the IP address of your own Instance.
3. You should now be connected to your VM Instance via SSH.
  - Use the Bash terminal to install your software-stack and perform work.
  - You can also add user credentials to your VM Instance to grant access to other users.

**Note:** In a newly created Instance, "cades" is the only user name that will correctly authenticate over SSH (using an SSH key pair). A user can add more users, and add public keys for each, once logged in as "cades" ([instructions](#)).

## Extract Public Key

If you have several users, you can share your public key with them. New users must login using the username 'cades' to gain access to the instance, or you can [create another user account](#) for them. Once a user extracts the `.pub` file they can safely share with another OpenStack user. The receiving user can add that public key to the instance's `/home/cades/.ssh/authorized_keys` file.

To extract the public key from the private key:

```
openssl rsa -in privkey.pem -pubout > key.pub
```

In this way, through exchanging public keys, users can control who has SSH access to their instances.

Users should refrain from insecurely sending someone `.pem` files as they contains both public and private keys.

## Related Tutorials

- [Add More Users to VM Instances](#)
- [Access VM Instances Using PuTTY \(Windows\)](#)
- [Access Your VM Instances Using Horizon](#)



## Access Your VM Instance Using PuTTY (Windows)

Once you have created a VM Instance using Horizon, you can access this VM Instance through a secure shell (SSH) using an SSH key pair. For Windows users, the CADES team recommends [PuTTY](#). The [PuTTY MSI installer](#) also includes PuTTYGen, which allows you to convert a `.pem` private key file to a `.ppk` PuTTY key file.

If you would like to access your VM Instance over SSH from *outside* of the ORNL network, you must first [create a firewall \(SAFER\) exception](#) for this purpose. All local (within the ORNL network) SSH connections are permitted by default.

## Table of Contents

- [Add SSH Keys to a VM Instance](#)
  - [Generate a New Key Pair](#)
  - [Import an Existing Key](#)
- [Download and Install PuTTY](#)
- [Connect to Your VM Instance Using PuTTY](#)
  - [Find Your Instance's IP](#)
  - [Connect to Your Instance](#)
- [Related Tutorials](#)

## Add SSH Keys to a VM Instance

If you do not have any existing SSH key pairs, or you wish to generate a new key pair for your VM Instance, then you should use the Horizon web GUI to [generate a new SSH key pair](#) (Option 1). If you *do* have an existing key pair on your machine, then you can [import the public key using the Horizon web GUI](#) (Option 2).

In either case, you will need to log into the Horizon web GUI to associate your SSH key with your VM Instance(s).

## Log in to Horizon

1. Navigate to the Horizon web interface at <https://cloud.cades.ornl.gov/>.
2. Log in with your UCAMS credentials.
  - Domain: `ornl`
  - User Name: `Your three-letter UCAMS ID`
  - Password: `Your UCAMS password`

## Option 1: Generate a New Key Pair

1. Within Horizon, navigate to `Project` → `Compute` → `Access & Security` → `Key Pairs`.
2. In the Key Pairs screen, click the `+ Create New Pair` button.

The screenshot shows the Red Hat OpenStack Platform's Horizon web interface. The top navigation bar includes 'RED HAT OPENSTACK PLATFORM', 'Project', 'Identity', and a dropdown for 'Project'. Below this is a secondary navigation bar with 'Compute', 'Network', 'Orchestration', 'Object Store', and 'Data Processing'. Under 'Compute', 'Instances', 'Volumes', 'Images', and 'Access & Security' are listed. 'Access & Security' is currently selected. A sub-menu below it shows 'Security Groups', 'Key Pairs' (which is highlighted with an orange box), 'Floating IPs', and 'API Access'. At the bottom of this menu are 'Filter', a search icon, '+ Create Key Pair', 'Import Key Pair', and 'Delete Key Pairs'.

- In the resulting dialog, name your Key Pair, and click `Create Key Pair` when you're finished. We went with `blackmesakey` in this example.

The screenshot shows the 'Create Key Pair' dialog. It has a title bar 'Create Key Pair' with a close button. On the left, there is a 'Key Pair Name \*' field containing 'blackmesakey', which is also highlighted with an orange box. To the right, there is a 'Description:' section with explanatory text about key pairs and their usage. At the bottom right of the dialog is a blue 'Create Key Pair' button, which is also highlighted with an orange arrow.

- The private key will be downloaded to your local machine as a `.pem` file. The public key will be stored in OpenStack.
- Place the downloaded private key in a directory of your choosing, e.g., `C:\Users\Username\SSH`.

**Note:** The private key is in a `.pem` format and has to be converted to a `.ppk` file using PuTTYGen.

Having completed the procedure above, you can now [connect to your Instance using PuTTY](#).

## Option 2: Import an Existing Key

If you have an *existing* key pair that you would like to also use for your OpenStack Project, then you can import the public key using the Horizon web GUI.

- Within Horizon, navigate to `Project` → `Compute` → `Access & Security` → `Key Pairs`.
- In the Key Pairs screen, click the `Import Key Pair` button.

The screenshot shows the 'Key Pairs' screen. At the top, there are tabs for 'Security Groups', 'Key Pairs' (which is highlighted with an orange box), 'Floating IPs', and 'API Access'. Below the tabs are 'Filter', a search icon, '+ Create Key Pair', 'Import Key Pair' (which is highlighted with an orange arrow), and 'Delete Key Pairs'. The main area displays a table of key pairs. The first row has columns for 'Key Pair Name' (empty) and 'Fingerprint' (empty). The second row, for 'blackmesakey', has 'Key Pair Name' with a checkbox and 'Fingerprint' with the value 'b3:8b:de:91:90:3e:8b:47:9d:9c:54:23:f0:72:44:b2'. To the right of the table is an 'Actions' column with a 'Delete Key Pair' button.

- In the resulting dialog, enter a key pair name in the `Key Pair Name` field.
- On your local machine, locate your public key file and open it with a text editor.
- Copy the key to your clipboard using the text editor.
- Back in Horizon, paste the public key into the `Public Key` text box.
- Click the `Import Key Pair` button.

**Import Key Pair**

**Key Pair Name \***  
blackmesakey

**Public Key \***

```
ssh-rsa
AAAAB3NzaC1yc2EAAAQABAAQDKV3
oXnjn+GJis2v8PoFB6sSz9Dwt1hBeFVFFy'sM6H
HgTtsZlFFFoGrfjBHctL2FlPyqG2cG36gHOgph8wj
3rY88jQBUq5uat6z88jDJ1KBZFLONMvH3Wi0+qN
nkwlrr9pHdlvuKP5x5xswFwk+zmtPevaBNISI+1
6Lj2Rak9vdISOvBzXkCaaP5CbzipahUsYVP4Ts3K
9i43w4Q+HiZyvUIMPwazM+68T7MOOPQ2/bNgci
qmqYNUOTzPloS7u1e3Vj6uxWBHdvFhzx1u5D
SxFsQdHjFtcdkpQdDpJzFWONXHTpgJs5/yeifQbC
-----END-----
```

**Description:**

Key Pairs are how you login to your instance after it is launched.

Choose a key pair name you will recognise and paste your SSH public key into the space provided.

SSH key pairs can be generated with the ssh-keygen command:

```
ssh-keygen -t rsa -f cloud.key
```

This generates a pair of keys: a key you keep private (cloud.key) and a public key (cloud.key.pub). Paste the contents of the public key file here.

After launching an instance, you login using the private key (the username might be different depending on the image you launched):

```
ssh -i cloud.key <username>@<instance_ip>
```

**Cancel** **Import Key Pair**

8. Confirm your newly imported key appears in the Key Pair list.

Having completed the procedure above, you can now [connect to your Instance using PuTTY](#).

## Download and Install PuTTY

1. Navitage to the [official PuTTY download page](#).
2. Download the appropriate MSI file for your Windows system (32 bit or 64 bit).
  - o Alternatively, if you don't want to "install" PuTTY on your system, you can download `putty.exe` and `puttygen.exe` from the "Alternative binary files" list and execute them as needed.
3. Run the PuTTY MSI installer. Note the destination directory.
4. Confirm installation of PuTTY executables.

## Connect to Your VM Instance Using PuTTY

### Find Your Instance's IP

The IP address of your VM instance is shown in the Instances tab in the Horizon web GUI.

1. Navigate to `Project` → `Compute` → `Instances`.
2. From the Instances screen, you can copy the IP address for the desired VM Instance.

The screenshot shows the Red Hat OpenStack Platform interface. The top navigation bar includes 'RED HAT OPENSTACK PLATFORM', 'Project', 'Identity', 'Compute', 'Network', 'Orchestration', 'Object Store', and 'Data Processing'. Below this is a secondary navigation bar with 'Overview', 'Instances' (which is selected), 'Volumes', 'Images', and 'Access & Security'. The main content area is titled 'Instances' and displays a table with the following data:

Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
black-mesa	m1.tiny	128.219.186.19		blackmesakey	Active	nova	None	Running	1 week, 3 days	<button>Create Snapshot</button>

Displaying 1 item

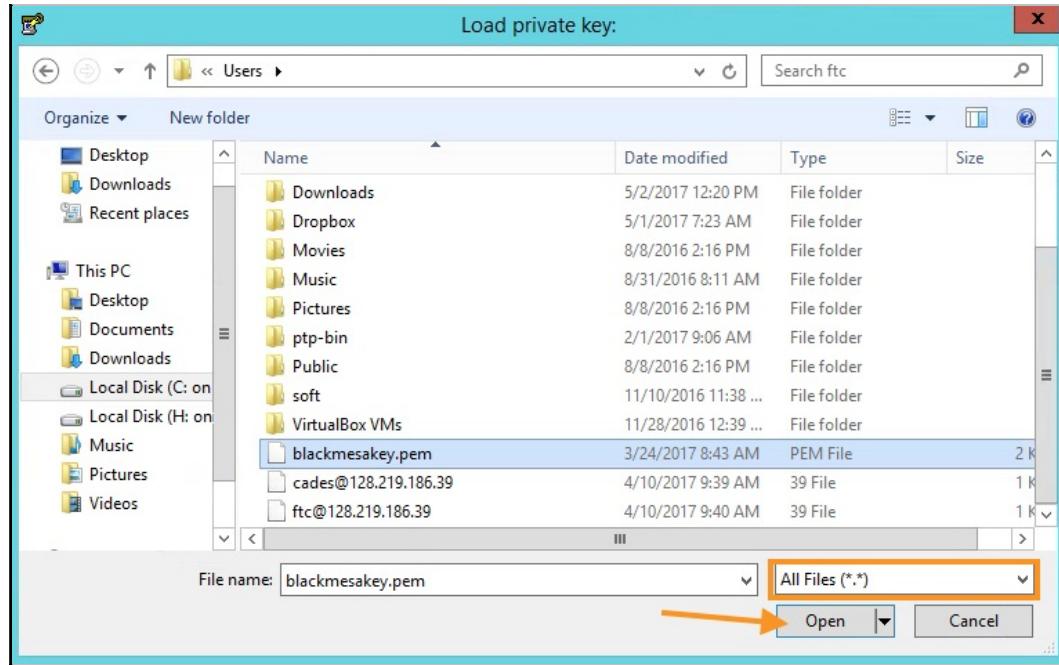
## Convert Your Private Key to PuTTY Format

If you generated a key in the `.pem` file format, then you need to convert that key to the `.ppk` format using PuTTYGen.

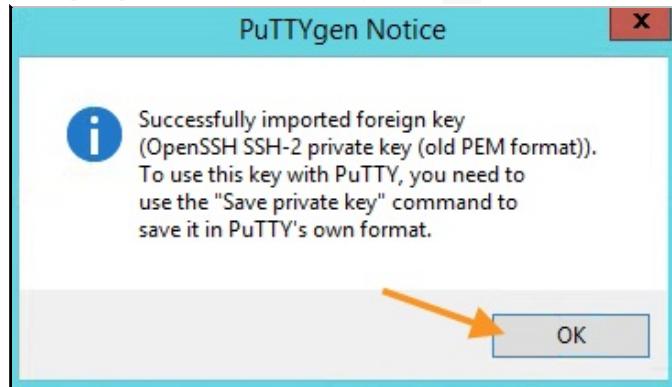
1. Locate and execute `PuTTYGen.exe`.
2. Click the `Load` button.

The screenshot shows the PuTTY Key Generator application window. The title bar says 'PuTTY Key Generator'. The menu bar includes 'File', 'Key', 'Conversions', and 'Help'. The main area has sections for 'Key' (which says 'No key.') and 'Actions'. The 'Actions' section contains buttons for 'Generate', 'Load', 'Save public key', and 'Save private key'. An orange arrow points to the 'Load' button. Below these are sections for 'Parameters' where 'RSA' is selected as the key type, and a field for 'Number of bits in a generated key' set to 2048.

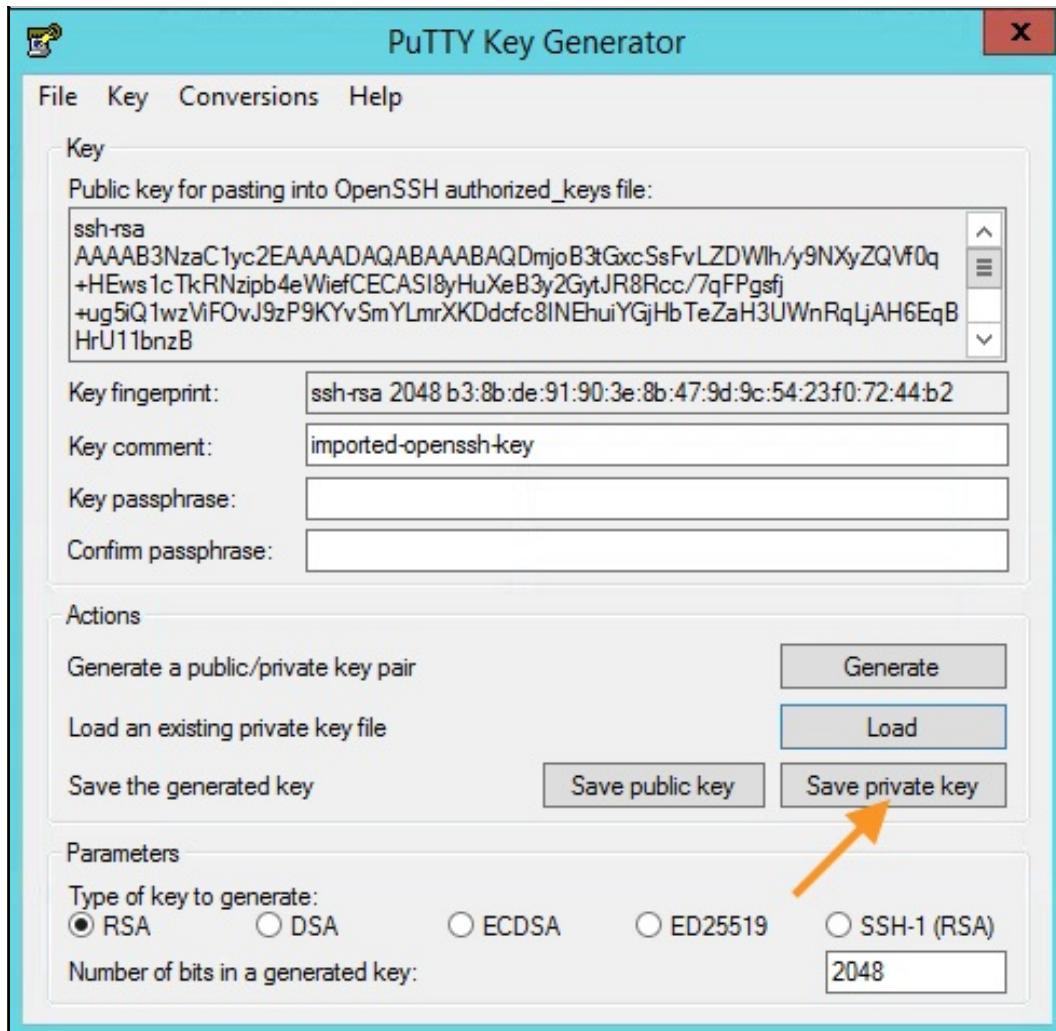
3. In the resulting dialog, navigate to the location of your key file.
4. Select `All Files (*.*)` to show `.pem` files in the dialog.
5. Select the appropriate key file and click `open`.



- When prompted by the success message, click `OK`.



- Save the loaded key by clicking `Save private key`.



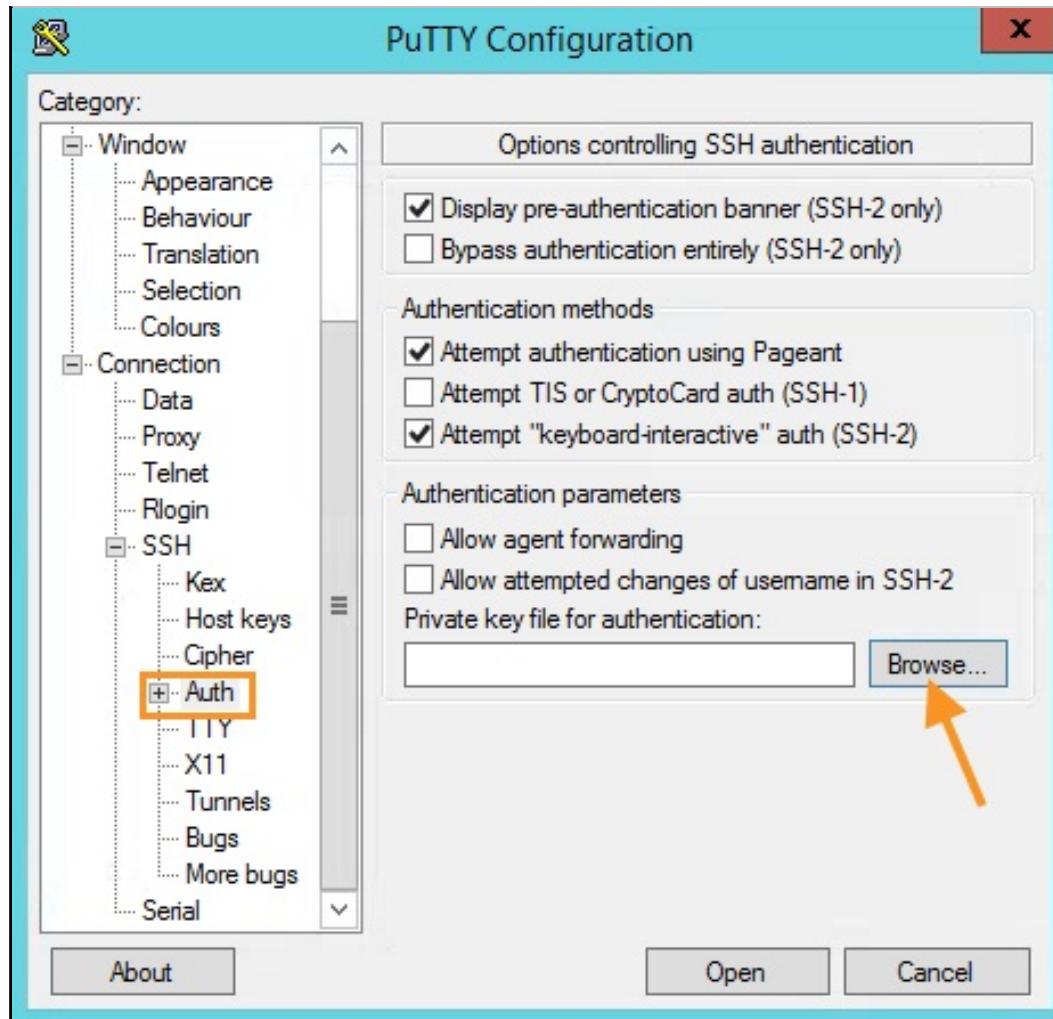
- Using the resulting dialog, save the key in an intuitive place (e.g., `C:\Users\Username\SSH`).

Your key is now saved as a `.ppk` file that can be loaded directly into PuTTY, and we can now use PuTTY to connect to your VM Instance, without a password, using the "cades" username.

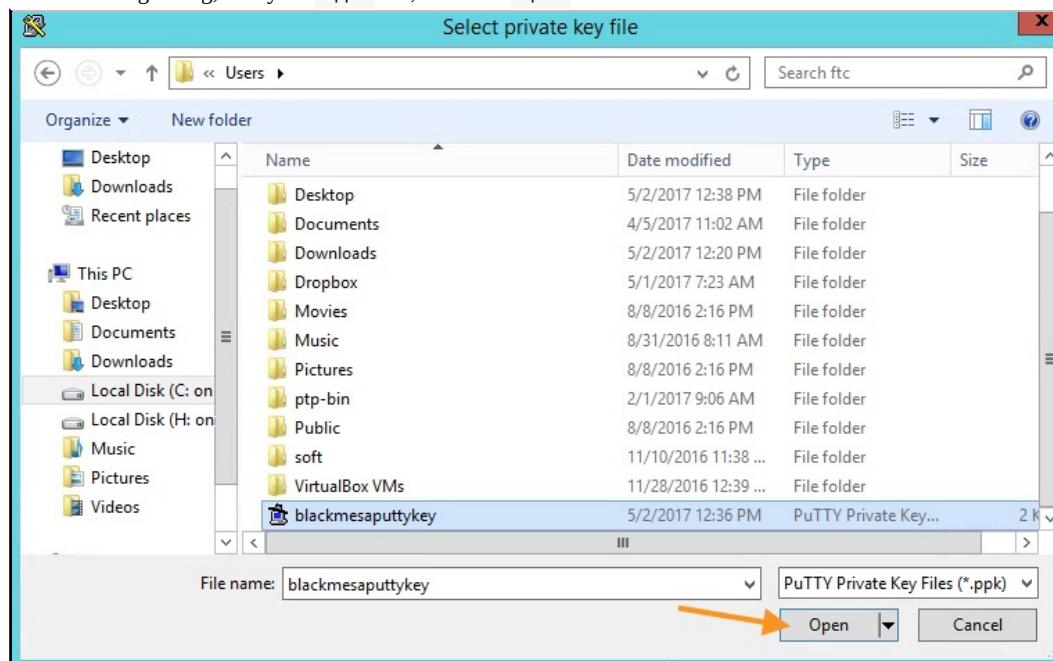
## Connect to Your Instance

First we're going to load the `.ppk` key file into PuTTY.

- Locate and execute `PuTTY.exe`.
- In PuTTY's navigation pane, go to `Connection` → `SSH` → `Auth`.
- In the "Options for controlling SSH authentication" screen, click `Browse`.



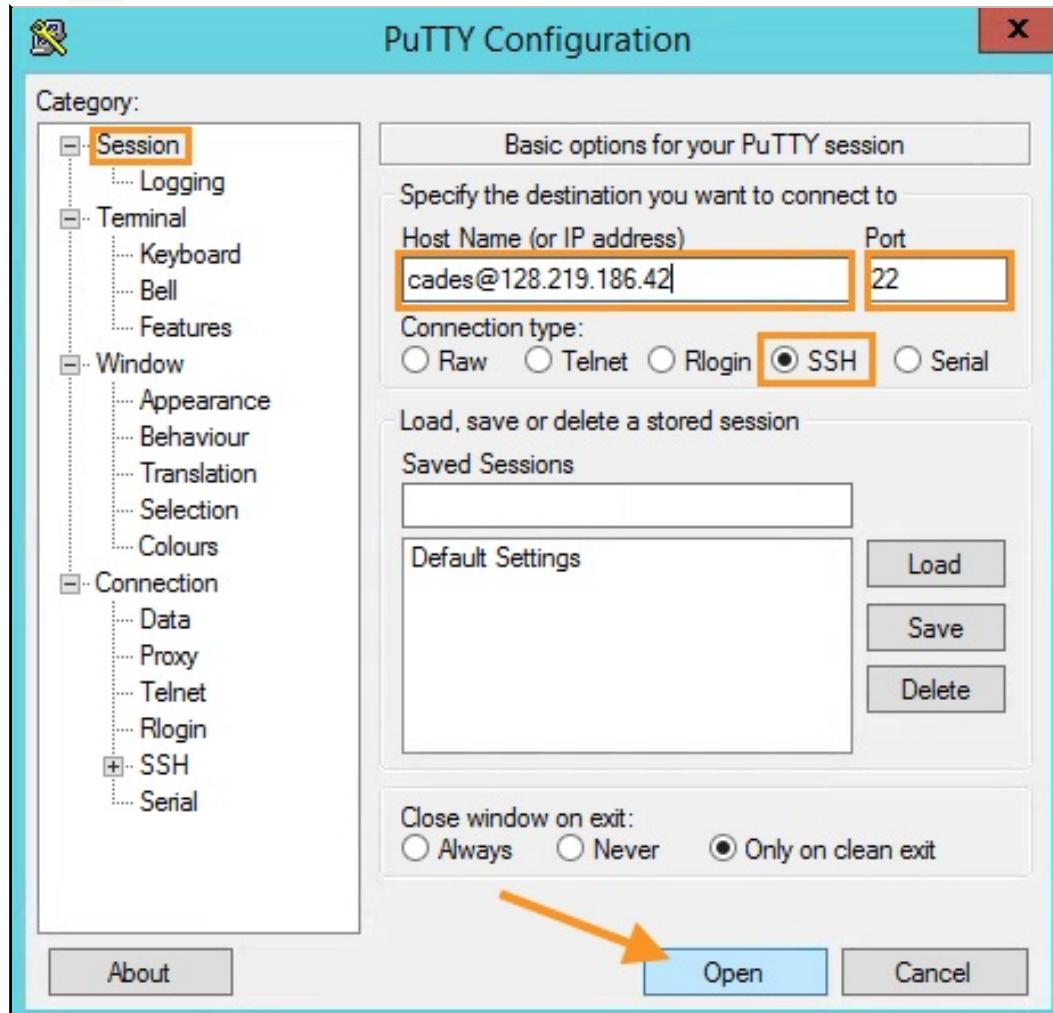
4. In the resulting dialog, find your .ppk file, and click open .



Your key file is now loaded into PuTTY. Next, we need to add your host information.

1. In PuTTY's navigation pane, click `Session` .

2. In the `Host Name (or IP address)` field, enter the IP address of your VM Instance preceded by "cades" (e.g., `cades@128.219.186.42`).
3. Ensure that the `SSH` radio button is selected and the port is set to `22`.
4. Click `open` to connect to your VM Instance using SSH.



5. If prompted, you can choose to cache the server's fingerprint (click `yes`), not to cache the server's fingerprint for this session (click `No`), or to reject the connection outright (click `cancel`). We're going to click `Yes`.



6. You should now have access to your VM Instance's Bash terminal.

## Related Tutorials

- Add More Users to VM Instances
  - Access Your VM Instances Using Horizon
  - Access Your VM Instances Using SSH



CADES → User Documentation → Birthright Cloud User Guide → Access Your VM Instance → Access Your VM Instance Using Horizon

## Access Your VM Instance Using Horizon

The Horizon web GUI also has a built-in console from which you can access your VM Instance once you have [added a generic user over SSH](#). Note that this is not the preferred method, and that the CADES team recommends using [the traditional SSH connection for accessing your VM Instance](#).

### Prerequisites

The Horizon console uses a simple username/password authentication model, and cannot leverage SSH key pairs for authentication. This means that you cannot use the console with the "cades" username or your UCAMS credentials.

As a result, to access your VM Instance using the Horizon console, you must first use SSH to create a non-UCAMS user. More info on creating users is available [here](#).

### Log in to Horizon

1. Navigate to the Horizon web interface at <https://cloud.cades.ornl.gov/>.
2. Log in with your UCAMS credentials.
  - o Domain: `ornl`
  - o User Name: `Your three-letter UCAMS ID`
  - o Password: `Your UCAMS password`

We can now use Horizon to access your VM Instances using the Console option.

### Launch Console

1. Navigate to `Project` → `Compute` → `Instances`.
2. Click the drop-down menu of the Instance you would like to access.
3. Select `Console` from the resulting menu.

The screenshot shows the 'Instances' tab selected in the navigation bar. A table lists one instance: 'black-mesa' (Image Name: CADES\_Ubuntu16.04\_v20170111\_0). The context menu for this instance is open, with 'Console' highlighted and an orange arrow pointing to the screenshot below.

4. In the console, input your user credentials.
5. Upon successful login, you can execute commands on your VM Instance.

The screenshot shows a VNC session to the 'black\_mesa' VM. The terminal window displays the following text:

```

Connected (encrypted) to: QEMU (Instance-0000087b)
Unauthorized or improper use of this system may result in administrative
disciplinary action and civil and criminal penalties. By continuing to use
this system you indicate your awareness of and consent to these terms and
conditions of use. LOG OFF IMMEDIATELY if you do not agree to the
conditions stated in this warning.

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
Welcome to OpenStack

Support:
Documentation/SLA: https://code.ornl.gov/cades-ua/user-documentation/
Email support: cades-help@email.ornl.gov
Chat/community: https://cades.slack.com/

sudo:
The 'cades' user on this VM has full sudo.
You may become your ucams user via:
sudo su - ucamsID
(Provide ucams password)

This message:
Served courtesy of /etc/profile.d/motdUserInfo.sh
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
sam@black-mesa:"$ _

```

**Note:** To use the Horizon console, you must first add a user and password to the Instance's operating system (via SSH) to enable access to the VM without an SSH key.

## Related Tutorials

- [Add More Users to VM Instances](#)
- [Access Your VM Instances Using SSH](#)
- [Access Your VM Instances Using PuTTY \(Windows\)](#)



## Add More Users to Your VM Instance

Once you have gained [SSH access](#) to your Instance (as "cades"), additional users can be created through the Bash terminal. The CADES OpenStack Instances leverage ORNL's UCAMS system to enable customers to create UCAMS users and home areas on their Instances. The following procedures will enable you to create a new user on your VM Instance and grant access using SSH key pairs.

### Add a UCAMS User

Because your Instance can access ORNL's UCAMS system, you can easily add a UCAMS user to your Instance. You can start with your own UCAMS ID to verify functionality and then move on to adding the UCAMS IDs of your collaborators (using their public SSH keys that they must provide to you).

1. Open a Bash terminal.
2. Upload a copy of your (or your collaborator's) public key (`blackmesakey.pub` in this case), `$ scp ~/.ssh/blackmesakey.pub cades@128.219.186.39:/home/cades`. Replace `128.219.186.39` with the IP address of your Instance.
3. Using SSH, log in to your VM Instance as "cades".
4. Once logged in, become root, `$ sudo -s`.
5. Create the home directory for the UCAMS ID, `$ su - UCAMS`. Replace "UCAMS" with the desired UCAMS ID.
6. Use `cat` and `>>` to copy the contents of your public SSH key to `/home/UCAMS/.ssh/authorized_keys`.

SSH commands for the above procedure are provided below. First:

```
$ scp ~/.ssh/blackmesakey.pub cades@128.219.186.39:/home/cades
```

Then:

```
$ ssh cades@128.219.186.39
$ sudo -s
$ su - UCAMS
$ cat /home/cades/blackmesakey.pub >> /home/UCAMS/.ssh/authorized_keys
```

Properly configured, you can now access your VM Instance over SSH using your UCAMS ID (instead of "cades"). Execute the command shown below (replace "UCAMS" with your own UCAMS ID).

```
$ ssh UCAMS@128.219.186.39
```

**Note:** You will not be prompted for your UCAMS password. This process uses SSH key pairs for authentication.

### Add a Non-UCAMS User

In some cases, it may be useful to have a generic user (not affiliated with UCAMS) on your VM Instance. The following procedure outlines how to create such a user.

1. Open a Bash terminal.
2. Upload a copy of your (or your collaborator's) public key (`blackmesakey.pub` in this case), `$ scp ~/.ssh/blackmesakey.pub cades@128.219.186.39:/home/cades`. Replace `128.219.186.39` with the IP address of your

Instance.

3. Using SSH, log in to your VM Instance as "cades".
4. Once logged in, become root, `$ sudo -s`.
5. Add a new user, `useradd USERNAME`.
6. Create a password for this user, `passwd USERNAME`.
  - Enter desired password. Reenter password to confirm.
7. Create the home directory for the generic user, `$ su - USERNAME`. Replace "USERNAME" with the desired user name.
8. Use `cat` and `>>` to copy the contents of your public SSH key to `/home/USERNAME/.ssh/authorized_keys`. Replace "USERNAME" with the desired user name.

SSH commands for the above procedure are provided below. First:

```
$ scp ~/.ssh/blackmesakey.pub cades@128.219.186.39:/home/cades
```

Then (replacing "USERNAME" with your desired username):

```
$ ssh cades@128.219.186.39
$ sudo -s
$ useradd USERNAME
$ passwd USERNAME
$ su - USERNAME
$ exit

$ sudo -s
$ cat /home/cades/blackmesakey.pub >> /home/USERNAME/.ssh/authorized_keys
```

Properly configured, you can now access your VM Instance over SSH using your newly created username (instead of "cades"). Execute the command shown below (replace "USERNAME" with your new username).

```
$ ssh USERNAME@128.219.186.39
```

See the [Access Your VM Instance documentation](#) for instructions on how to use SSH and Horizon to access your Instance.

## Related Tutorials

- [Access Your VM Instances Using SSH](#)
- [Access Your VM Instances Using Horizon](#)

## Manage Your VM Instances

The Horizon web GUI enables users to view and manage their VM Instances directly from a web browser. Using the following procedure, users can manage existing instances and create new ones through the Instances management screen.

1. Navigate to the Horizon web interface at <https://cloud.cades.ornl.gov/>.
2. Log in with your UCAMS credentials.
  - Domain: `ornl`
  - User Name: `UCAMS ID`
  - Password: `UCAMS password`
3. Click on the `Project` tab on the top left.
4. Select the `Compute` sub tab.
5. Finally, click on the `Instances` sub tab.

From the Instances screen, you can:

- [Launch a new VM Instance from an Image](#)
- [Delete old VM Instances](#)
- [Delete a Volume](#)
- [Resize a VM](#)
- [Create a Snapshot](#)
- [Modify Security Groups](#)

## Delete a VM Instance from your Project

At some point you may wish to delete a VM Instance from your project. This can help a user free up space in his or her allocation or clear out old Instances that have outlived their usefulness. Deleting a VM Instance is a simple process, outlined in the procedure below.

### Log in to Horizon

1. Navigate to the Horizon web interface at <https://cloud.cades.ornl.gov/>.
2. Log in with your UCAMS credentials.
  - o Domain: `ornl`
  - o User Name: `Your three-letter UCAMS ID`
  - o Password: `Your UCAMS password`

From here, you can manage your VM Instance(s) from within Horizon.

### Delete an Instance

1. Navigate to `Project` → `Compute` → `Instances`.
2. Click the drop-down menu of the Instance you wish to delete.
3. Select `Delete Instance` from the resulting menu.

The screenshot shows the Horizon Compute Instances page. At the top, there are navigation tabs: Compute, Network, Orchestration, Object Store, and Data Processing. Below that is a secondary navigation bar with tabs: Overview, Instances (which is selected), Volumes, Images, and Access & Security. The main area is titled 'Instances' and displays two items:

<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"/> <code>glados</code>	CADES_CentOS-7.3_v20170310_0	128.219.186.31	<code>m1.tiny</code>	blackmesakey	Active	nova	None	Running	5 minutes	<code>Create Snapshot</code> <span style="border: 1px solid orange; padding: 2px;">▼</span>
<input type="checkbox"/> <code>black-mesa</code>	CADES_Ubuntu16.04_v20170111_0	128.219.186.29	<code>m1.tiny</code>	blackmesakey	Active	nova	None	Running	6 hours, 18 minutes	<code>Associate Floating IP</code> <code>Attach Interface</code> <code>Detach Interface</code> <code>Edit Instance</code> <code>Update Metadata</code> <code>Edit Security Groups</code> <code>Console</code> <code>View Log</code> <code>Pause Instance</code> <code>Suspend Instance</code> <code>Shelve Instance</code> <code>Resize Instance</code> <code>Lock Instance</code> <code>Unlock Instance</code> <code>Soft Reboot Instance</code> <code>Hard Reboot Instance</code> <code>Shut Off Instance</code> <code>Rebuild Instance</code> <code>Delete Instance</code>

Below the table, it says 'Displaying 2 items'. To the right of the table, there is a vertical 'Actions' dropdown menu. The 'Delete Instance' option is highlighted with an orange border. An orange arrow points from the text in step 3 to this highlighted option.

4. Click `Delete Instance` in the dialog.
5. Confirm that the Instance is scheduled for deletion.

**Note:** Deleting a VM Instance does not delete the volume associated with the Instance. To remove both, you must follow the [Delete a VM Instance](#) procedure *and* the [Delete a Volume](#) procedure.

## Related Tutorials

- [Delete a Volume](#)
- [Launch a VM Instance](#)

## Delete a Volume from Your Project

If you decide to delete a VM Instance, you may still have its associated Volume left behind (i.e., `Delete Volume on Instance` was set to `No` when you launched the Instance). Use the following procedure if you wish to delete a Volume.

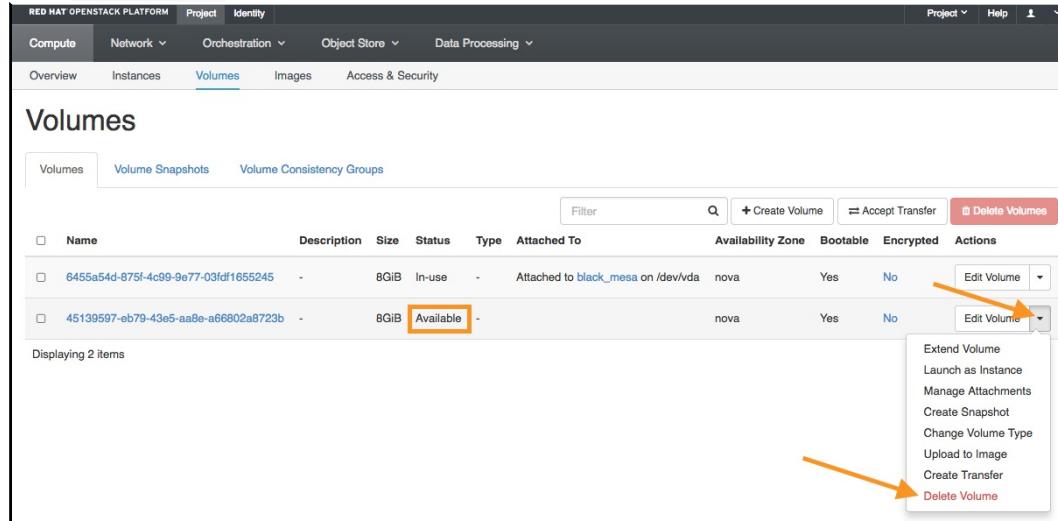
## Log in to Horizon

1. Navigate to the Horizon web interface at <https://cloud.cades.ornl.gov/>.
2. Log in with your UCAMS credentials.
  - Domain: `ornl`
  - User Name: `Your three-letter UCAMS ID`
  - Password: `Your UCAMS password`

From here, you can delete the Volume from your Project.

## Delete a Volume

1. Navigate to `Project` → `Compute` → `Volumes`.
2. In the Volumes screen, ensure that the Volume is not in use by any VM Instance.
3. Click the drop-down menu of the volume you wish to delete.
4. Select `Delete Volume` from the resulting menu.



5. Click `Delete Volume` in the resulting dialog.
6. Confirm that the Volume is no longer listed in the Volumes screen.

## Related Tutorials

- [Delete a VM Instance](#)
- [Launch a VM Instance](#)



## Resize an existing VM Instance

Sometimes users may wish to add CPU or Memory resources to an existing VM instance. This can be accomplished easily through the Horizon interface:

1. Navigate to your Instances List at <https://cloud.cades.ornl.gov/dashboard/project/instances/>.
2. Under `Actions` on the right-hand side, select `Resize Instance` for the instance you wish to resize.
3. You will be prompted to select a new `Flavor`.
4. As long as the new flavor fits in your `Allocation`, click `Resize` in the bottom right corner and your instance will begin resizing.
5. Your instance will reboot into the new flavor. Once this is complete, click `Confirm Resize or Migration`

At this point, your instance will be up and running at the size you selected.

## Create a Snapshot of a VM Instance

A VM snapshot is an instantaneous duplicate of an instance. A snapshot of an instance can be used to back-up data, to create a restore point, or as the basis of an instance and booted up at a later time. A snapshot can be booted as a new VM Instance and contains an image of the state of the filesystem at the moment that the snapshot is taken.

## Log in to Horizon

1. Navigate to the Horizon web interface at <https://cloud.cades.ornl.gov/>.
2. Log in with your UCAMS credentials.
  - o Domain: `ornl`
  - o User Name: `Your three-letter UCAMS ID`
  - o Password: `Your UCAMS password`

From here, you can manage your VM Instance(s) from within Horizon.

## Create a Snapshot

1. Navigate to `Project` → `Compute` → `Instances`.
2. Click on `Create Snapshot` for the Instance.

Status	Availability Zone	Task	Power State	Time since created	Actions
Active	nova	None	Running	2 months	<button>Create Snapshot</button>
Active	nova	None	Running	2 months	<button>Create Snapshot</button>

3. In the window that appears, choose a descriptive name for your snapshot and then click `Create Snapshot`.
4. Once the snapshot is created, a list of snapshot will appear. You can later navigate to this list by navigating to `Project` → `Compute` → `Images`.
5. From the `Images` screen you can launch, edit, or delete the snapshot.

## Related Tutorials

- [Delete a VM Instance](#)
- [Launch a VM Instance](#)

## OpenStack Security Groups

At their core, the OpenStack Security Groups are iptable-based firewalls built around an Instance at the hypervisor level. The Security Groups can be used in conjunction with the OS-level firewalls (e.g., FirewallD, iptables) but do not overlap with them (see [Important Notes](#)).

**Note:** If you wish to run services on your VM Instance that should be available outside of ORNL's network, ensure that you select the External Network option when setting up your VM Instance and that you also add a rule to your Security Group for that particular service.

## Important Notes for OpenStack Security Groups

- IPV6 is not currently supported in OpenStack.
- Changes to Security Groups take effect immediately.
- Unlike normal Linux firewall rules, the rule order does not matter in OpenStack Security Groups.
- By default, all Instances within the same Project can communicate with each other.
- Using 160.91.8.218:6556 to access ORNL's Check\_MK service is allowed but not enabled by default. For monitoring of uptime and basic metrics, please contact the [CADES team](#) for assistance.
- No firewall is enabled in the CADES-provided operating system (OS) images. Instead, we rely on the OpenStack Security Groups. The user is responsible for enabling and configuring extra OS-level firewall rules as desired.
- User-added firewall and iptable rules supersede rules set in OpenStack Security Groups. For example, ingress access enabled by a rule in the OpenStack Security Group that are otherwise blocked at the OS level using the firewall or iptables will be ineffective, and that traffic will still be blocked.
- By default, all newly created Security Groups allow all outbound IPV4 and IPV6 (enabled but not functional). By default, no inbound traffic is allowed.
- The CADES team recommends that users leave existing Security Group rules in place as many of these rules are used by the CADES support team (e.g., for monitoring and metrics).

CADES → User Documentation → Birthright Cloud User Guide → Manage Your VM Instances → OpenStack Security Groups → Modify Security Groups

## Modify the Default Security Group

1. Navigate to the Horizon web interface at <https://cloud.cades.ornl.gov/>.
2. Log in with your UCAMS credentials.
  - Domain: ornl
  - User Name: Your three-letter UCAMS ID
  - Password: Your UCAMS password
3. Navigate to Project → Compute → Access & Security → Security Groups .
4. On this screen, look for the row labeled "default." Click the Manage Rules button.

Name	Description	Actions
default	Default security group	<a href="#">Manage Rules</a>

From here you can [create a new rule](#create-a-new-rule) or [remove rules](#remove-rules).

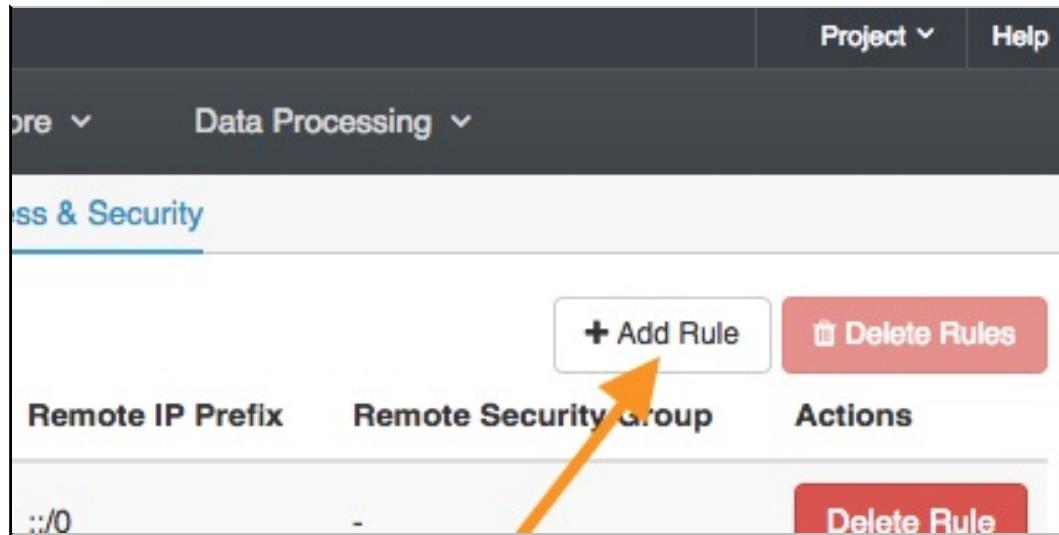
Direction	Ether Type	IP Protocol	Port Range	Remote IP Prefix	Remote Security Group	Actions
Egress	IPv6	Any	Any	::/0	-	<a href="#">Delete Rule</a>
Egress	IPv4	Any	Any	0.0.0.0/0	-	<a href="#">Delete Rule</a>
Ingress	IPv4	Any	Any	-	default	<a href="#">Delete Rule</a>
Ingress	IPv6	Any	Any	-	default	<a href="#">Delete Rule</a>
Ingress	IPv4	ICMP	Any	0.0.0.0/0	-	<a href="#">Delete Rule</a>
Ingress	IPv4	TCP	22 (SSH)	0.0.0.0/0	-	<a href="#">Delete Rule</a>
Ingress	IPv4	TCP	6556	160.91.8.218/32	-	<a href="#">Delete Rule</a>

## Create a New Rule

You can add a new rule to your Security Group [using the built-in rules](#) provided in OpenStack or you can create your own [custom rule](#).

### Create a Rule Using the Built-in Rules

1. Navigate to Project → Compute → Access & Security → Security Groups .
2. Click the Add Rule button.

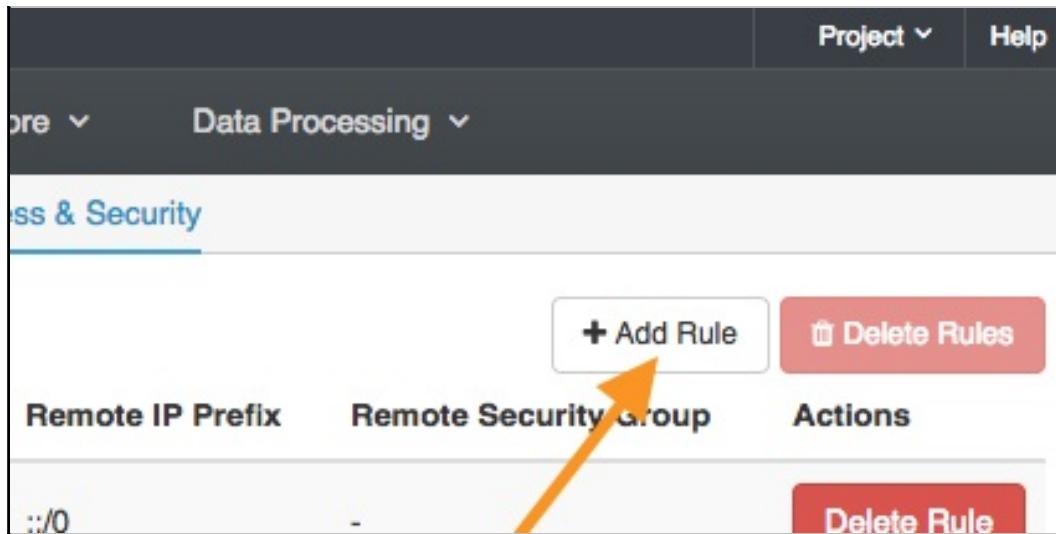


3. In the resulting dialog, click the drop-down field under Rule .
4. Choose a rule from the list that fits your needs (DNS, HTTP, HTTPS, etc.).
5. In the Remote box directly under Rule , choose either CIDR or Security Group .
  - o If you selected CIDR , enter the desired inter-domain range in the CIDR box directly under the Remote box. See [CIDR examples](#).
  - o If you select Security Group , choose a security group shared by the Instance(s) with which you wish to communicate. You can also do this via their IP addresses using the CIDR option.

**Note:** this option only allows access to the Instances within that security group. This differs from CIDR.
6. Click Add at the bottom of the dialog box to implement your rule.
7. Confirm the new rule is displayed in the Manage Rules screen.

### Create a Custom Rule

1. Navigate to Project → Compute → Access & Security → Security Groups .
2. Click the Add Rule button.



3. In the resulting dialog, click the drop-down field under `Rule`.
4. Choose the rule type from the drop-down list (e.g., `Custom TCP|ICMP|UDP Rule`).
5. Set the preferred direction in the `Direction` field (i.e., Ingress or Egress).
6. Choose either a single port or a range of ports in the `Open Port` section.
7. Enter the port or port range in the respective field.
8. In the `Remote` box, choose either `CIDR` or `Security Group`.
  - If you select `CIDR`, enter the desired inter-domain range in the `CIDR` field. See [CIDR examples](#).
  - If you select `Security Group`, choose a security group shared by the Instance(s) with which you wish to communicate.

**Note:** This option only allows access to the Instances within that security group. This differs from CIDR.
9. Click `Add` at the bottom of the dialog box to implement your custom rule.

**Add Rule**

<p><b>Rule *</b></p> <input type="text" value="Custom UDP Rule"/> <p><b>Direction</b></p> <input type="text" value="Ingress"/> <p><b>Open Port *</b></p> <input type="text" value="Port"/> <p><b>Port</b></p> <input type="text" value="27015"/> <p><b>Remote *</b></p> <input type="text" value="Security Group"/> <p><b>Security Group</b></p> <input type="text" value="default (current)"/> <p><b>Ether Type</b></p> <input type="text" value="IPv4"/>	<p><b>Description:</b></p> <p>Rules define which traffic is allowed to instances assigned to the security group. A security group rule consists of three main parts:</p> <p><b>Rule:</b> You can specify the desired rule template or use custom rules, the options are Custom TCP Rule, Custom UDP Rule, or Custom ICMP Rule.</p> <p><b>Open Port/Port Range:</b> 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.</p> <p><b>Remote:</b> 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.</p>
--	--

10. Confirm your new custom rule is displayed in the Manage Rules screen.

## Remove Rules

### Remove an existing rule

1. Navigate to Project → Compute → Access & Security → Security Groups .
2. Select the rule that you would like to remove.
3. Click the Delete Rule button on the far right of the selected rule.
4. Confirm deletion of the rule.

### Remove multiple rules

1. Navigate to Project → Compute → Access & Security → Security Groups .
2. Toggle the check box next to each rule you would like to remove.
3. Click the Delete Rules button in the top-right corner of the Manage Rules screen.
4. Confirm deletion of the rules.

## Rename a Security Group

**Note:** You cannot rename the default security group.

1. In the Security Groups table, select the drop-down menu on the far right of the row.
2. Select Edit Security Group .

The screenshot shows the 'Access & Security' interface with the 'Security Groups' tab selected. The table displays two items: 'Custom' and 'default'. For the 'Custom' item, a context menu is open, showing options: 'Manage Rules', 'Edit Security Group' (which is highlighted with an orange arrow), and 'Delete Security Group'. The status bar at the bottom indicates 'Displaying 2 items'.

<input type="checkbox"/>	Name	Description	Actions
<input type="checkbox"/>	Custom	Custom security group	<button>Manage Rules</button>
<input type="checkbox"/>	default	Default security group	<button>Edit Security Group</button> <button>Delete Security Group</button>

3. In the resulting dialog, you can modify the name and description of the user-added Security Group.
4. Click Edit Security Group to save your changes.

**Edit Security Group**

**Name \***  
Custom

**Description**  
Custom security group

**Description:**  
Security groups are sets of IP filter rules that are applied to the network settings for the VM. Edit the security group to add and change the rules.

**Cancel** **Edit Security Group**



5. Confirm your changes in the Security Groups table.

## Delete a Security Group

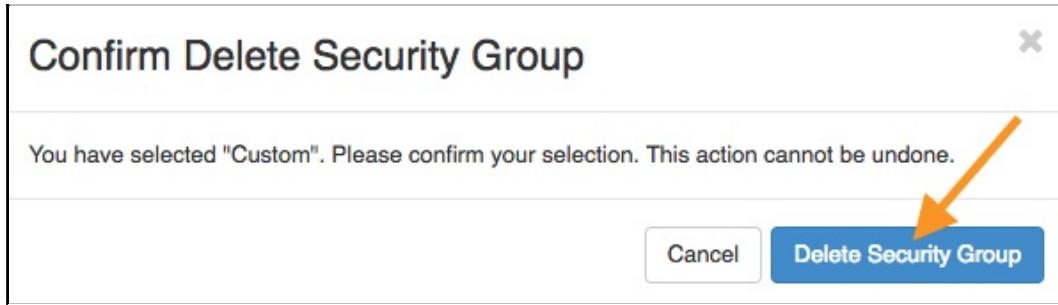
1. In the Security Groups table, find the Security Group you wish to delete, and select the drop-down menu on the far right of its row.
2. Select `Delete Security Group`.

**Access & Security**

Security Groups	Key Pairs	Floating IPs	API Access
<input type="button" value="Filter"/> <input type="button" value="Create Security Group"/> <input type="button" value="Delete Security Groups"/>			
<input type="checkbox"/> Name	Description	Actions	
<input type="checkbox"/> Custom	Custom security group	<input type="button" value="Manage Rules"/>	
<input type="checkbox"/> default	Default security group	<input type="button" value="Edit Security Group"/> <input type="button" value="Delete Security Group"/>	
Displaying 2 items			



3. Click `Delete Security Group` in the resulting dialog.



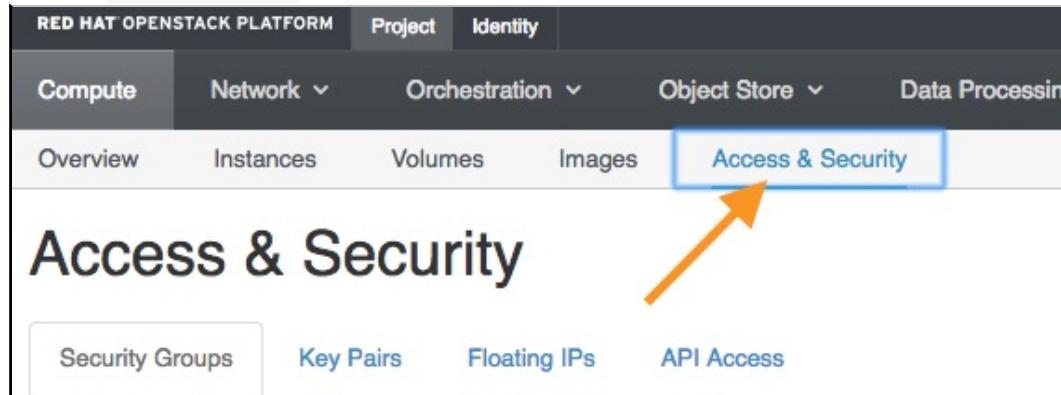
4. Confirm deletion of the Security Group.

CADES → User Documentation → Birthright Cloud User Guide → Manage Your VM Instances → OpenStack Security Groups  
→ Create New Security Groups

## Create a New Security Group

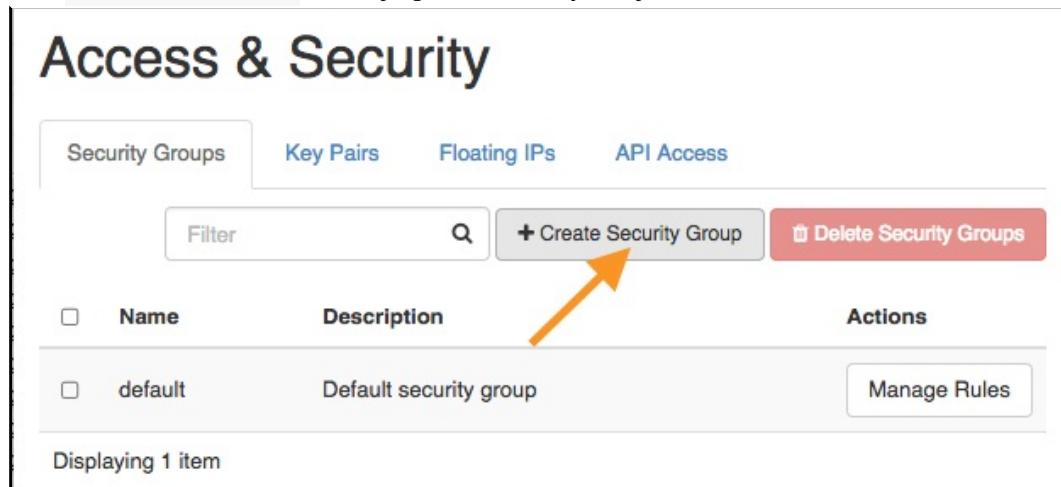
**Note:** By default, all newly created Security Groups allow all outbound IPV4 and IPV6 (enabled but not functional). By default, no inbound traffic is allowed.

1. Navigate to the Horizon web interface at <https://cloud.cades.ornl.gov/>.
2. Log in with your UCAMS credentials.
  - Domain: ornl
  - User Name: Your three-letter UCAMS ID
  - Password: Your UCAMS password
3. Click on the Project tab on the top left.
4. Select the Compute sub tab.
5. Select the Access & Security sub tab.



The screenshot shows the Red Hat OpenStack Platform dashboard. At the top, there is a navigation bar with tabs: RED HAT OPENSTACK PLATFORM, Project, Identity, Compute, Network, Orchestration, Object Store, and Data Processing. Below this, there is a secondary navigation bar with tabs: Overview, Instances, Volumes, Images, and Access & Security. The Access & Security tab is highlighted with a blue box and has an orange arrow pointing towards it from the bottom right. The main content area is titled "Access & Security" and contains four sub-tabs: Security Groups, Key Pairs, Floating IPs, and API Access.

6. Select the Security Groups sub tab to view a table/list of the Security Groups.
7. Click Create Security Group at the top right of the Security Groups table.



The screenshot shows the "Access & Security" table. At the top, there are four tabs: Security Groups, Key Pairs, Floating IPs, and API Access. Below these tabs is a search bar labeled "Filter" with a magnifying glass icon and a "Create Security Group" button with a plus sign icon. An orange arrow points towards the "Create Security Group" button. The table itself has columns for Name, Description, and Actions. One row is visible, showing "default" as the Name and "Default security group" as the Description. A "Manage Rules" button is located in the Actions column for this row. At the bottom of the table, it says "Displaying 1 item".

8. In the resulting dialog, fill out the Name and Description (optional) fields.
9. Click the Create Security Group button to complete the creation of the new Security Group.

**Create Security Group**

**Name \***  
Custom

**Description**  
Custom security group

**Description:**  
Security groups are sets of IP filter rules that are applied to the network settings for the VM. After the security group is created, you can add rules to the security group.

**Create Security Group**



Your new Security Group should now be available in the Security Groups table.

<b>Access &amp; Security</b>			
Security Groups		Key Pairs	Floating IPs
		Filter	+ Create Security Group
<input type="checkbox"/>	<b>Name</b>	<b>Description</b>	<b>Actions</b>
<input type="checkbox"/>	Custom	Custom security group	Manage Rules ▾
<input type="checkbox"/>	default	Default security group	Manage Rules
Displaying 2 items			

Just as you can with the default Security Group, you can create and manage rules for user-created Security Groups from the Manage Rules screen.

## Security Group CIDR Examples

When [adding a new rule to a Security Group](#), you can also specify CIDR configurations for each rule. Some examples are provided below.

- `0.0.0.0/0` – This CIDR configuration leaves traffic open to the world. However, other firewalls between the CIDR and the remote machine can still block traffic.
- `216.37.64.68/32` – This CIDR configuration only allows 216.37.64.68 access to the selected port(s). The `/32` is used to specify traffic for only the preceding IP address.
- `192.168.1.0/24` – This CIDR configuration allows all IPs between 192.168.1.1 and 192.168.1.254 access to the selected port(s). The `/24` is used to specify this traffic range.

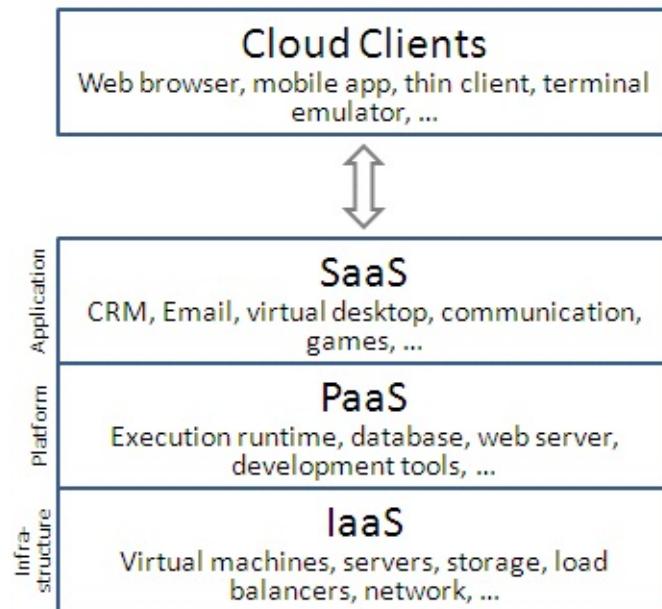
# Birthright Cloud User Guide

CADES (Compute and Data Environment for Science) provides eligible customers with an OpenStack-based cloud computing solution with customizable Virtual Machines (VM). This new resource, called "Birthright Cloud", enables customers in [science and technology directorates](#) to leverage self-service portals to rapidly request VMs for production, testing, and development. This documentation will walk you through how to configure and use your CADES Birthright Cloud allocation. The CADES Birthright Cloud allocations are intended and configured to be integrated within the ORNL network.

## What is cloud computing?

Cloud computing provides an efficient pool of on-demand, self-managed virtual infrastructure, consumed as a service.

As shown in the figure below, classic cloud clients interact with three layers in the cloud environment.



First, the Software as a Service (SaaS) layer that presents software tools and frameworks such as emails, enterprise systems to users. Second, the Platform as a Service (PaaS) that presents the runtime services such as web servers, programming environment etc. to user. Third, the Infrastructure as a Service (IaaS) layer that provides hardware and firmware such as storage, drivers and load balancers to users.

The CADES Birthright Cloud allocations provide:

- **Self Service** – Through the Horizon web interface, users can create, manage, and delete VMs.
- **Portable** – Operations can be performed using any local ORNL system that provides a Bash terminal and SSH functionality.
- **Elasticity** – Users can create VMs on demand and delete them when they are no longer desired.

## What is OpenStack?

[OpenStack](#) is an open-source cloud computing software framework that allows administrators (the CADES team) to create individual "Project" allocations for their customers. The customers/users can then fill these Project allocations with their own VMs without further intervention from CADES administrators—a true self-service implementation.



# Available VM Images and Configurations

The CADES OpenStack allocations can use either CADES-provided images or user-added images for the VM Instances. However, for full compatibility and best reliability, we strongly recommend that customers use one of the CADES images.

## CADES Images

The CADES team currently provides two Linux images for use in OpenStack.

- CentOS 7.x
- Ubuntu 16.04 LTS

## Features of CADES Images

CADES images have been preconfigured with the following features and modifications.

- The default user has been changed to "cades" and given full sudo privileges.
- A user named "cades-ops" has been added to ensure that the CADES team has adequate access to provide support.
- Limited Lightweight Directory Access Protocol (LDAP) functionality has been added to enable the use of UCAMS IDs after the initial key-based authentication as the "cades" user. To add your UCAMS ID, simply execute `sudo su - YOUR_UCAMS_ID` in the terminal.
  - You will not be prompted for a password.
  - A local home directory will be created for your user.
  - [Click here for more detailed instructions for adding UCAMS users to your Instance.](#)
- Yum/Apt repos and the Network Time Protocol (NTP) have been configured to use local ORNL resources.
- The minimum disk size required for each image is 8 GB.

## Naming Scheme for CADES Images

The CADES naming scheme is based around a `CADES_$OSRELEASE_v$BUILDDATE_${RELEASE}` formatting. The `RELEASE` field is always a single digit, with a `1` indicating a production version of an image and a `0` indicating a development version of the image.

## User Images

While you can run user-provided images in OpenStack, the CADES team strongly recommends that customers use CADES-provided images for best reliability and integration in the ORNL environment. **CADES will not provide support for user-provided images.** If you still want to run a custom cloud image please [contact the CADES team](#) for your request.

## Important Notes for Available VM Images

Once an image is launched as a volume, the image is no longer tied to the volume. If the base image is updated, those updates would not propagate to the Instance and vice versa.

## Available VM Instance Flavors

Birthright Cloud allocations are available in a variety of sizes to fit your needs. The size options can be viewed during VM setup.

If you see that your needs cannot be met by one of our preset configurations, feel free to [email us](#).

## Software & Hardware Details

### Software Stack

- **Host operating system:** Red Hat OpenStack Platform 9
- **Available VM operating systems:** CentOS 7.x and Ubuntu 16.04 LTS

### Hardware Configuration

#### RAM Information

- **Make:** Samsung
- **Model:** M393A2G40DB0-CPB
- **Speed:** DDR4 2133
- **Error correction:** Registered ECC

#### CPU Information

The following CPUs are used in the "nova" and "Lustre-OpenStack" Availability Zones.

- **Make:** Intel
- **Model:** Xeon E5-2698 v3
- **Speed:** 2.30 GHz base clock, 3.60 GHz Turbo Boost clock
- **Instruction set (VM Instances):** [CPUID Instance Codes](#)
- **Instruction set (Hypervisors):** [CPUID Hypervisor Codes](#)

Detailed CPU information is also available through CPUID.

| How to: [Install CPUID](#)

#### CPU Layout



# Network and Storage Details

## Network Configuration

The CADES Birthright Cloud allocations consist of two primary network environments or subnets—an external subnet and an internal subnet—described below. While either subnet can be used for a VM Instance, **only one IP address can be allocated to each subnet per VM Instance.**

**Note:** If you wish to run services on your VM Instance that should be available outside of ORNL's network, ensure that you select the [External Network option](#) when setting up your VM Instance and that you also add a rule to your Security Group for that particular service.

### External Network

- general\_extnetwork1: 128.219.184.0/21.
- **Required for services to be available outside of ORNL's network.**
- Outward-facing services (e.g., a web server) will require ORNL firewall exceptions ([instructions](#)).
- Routed to most "open" networks at ORNL.
- Outbound access is allowed per existing open research firewall exceptions.

### Internal Network

- general\_intnetwork1 = 172.22.0.0/20.
- **Services will not be available outside of ORNL's network.**
- Outward-facing services (e.g., a web server) will only be available from within ORNL.
- Routed to most "open" networks at ORNL.
- Outbound access is allowed through existing open research firewall exceptions.

### Important Notes for OpenStack Network Design

- IPV6 is not currently supported.
- In all cases, accessing a VM Instance via SSH from outside of ORNL's network requires a SAFER exception to allow inbound traffic on port 22 (SSH).
- SSH access *from* an instance to a destination outside of ORNL's network may route through the ORNL SSH proxy service ([instructions](#)), or the user can request an outbound exception using SAFER.
- If you wish to run services on your VM Instance that should be available outside of ORNL's network, ensure that you also select the [External Network option](#) when setting up your VM Instance.

## Storage Configuration

- **NFS** – Birthright Projects use a Network File System (NFS).
- **Lustre** – Lustre allocations are available upon request.

**Note:** No moderate/confidential data should be mounted or copied to your Birthright Cloud VM Instance. Birthright Cloud VM Instances are for open science.



[CADES](#) → [User Documentation](#) → [Birthright Cloud User Guide](#) → [Additional Cloud Resources](#)

## Additional Resources

There are some topics that are optional and their use is based on how you chose to use your Birthright VMs.

In this section you can:

- [Request Firewall Exception](#)
- [Run a Simple Web Server](#)
- [SSL - Let's Encrypt](#)
- [Install CPUID](#)
  - [CPUID Hypervisor Codes](#)
  - [CPUID Instance Codes](#)

Also, you can peruse the User-Contributed Tutorial section which currently covers the following topics:

- [Launch a Docker Container](#)
- [Launch Shiny within Docker](#)
- [Eclipse in CADES HPC](#)
- [Allinea DDT in CADES HPC](#)

**Note:** Content contributed by the community is not supported by CADES.

If there is something missing, please [email CADES](#) to let us know. You can also help us by contributing your own content. See the [Contributing](#) section for more information.

CADES → User Documentation → Birthright Cloud User Guide → Additional Cloud Resources → ORNL Firewall Configuration

## ORNL Firewall Configuration (SAFER)

In some cases, a customer may want to run a service on their VM Instance that should be available outside of ORNL's network (e.g., SSH, web server, GitLab service, Docker), which requires an exception in ORNL's firewall. The procedure below will show how to set up a firewall exception for a service running on your VM Instance.

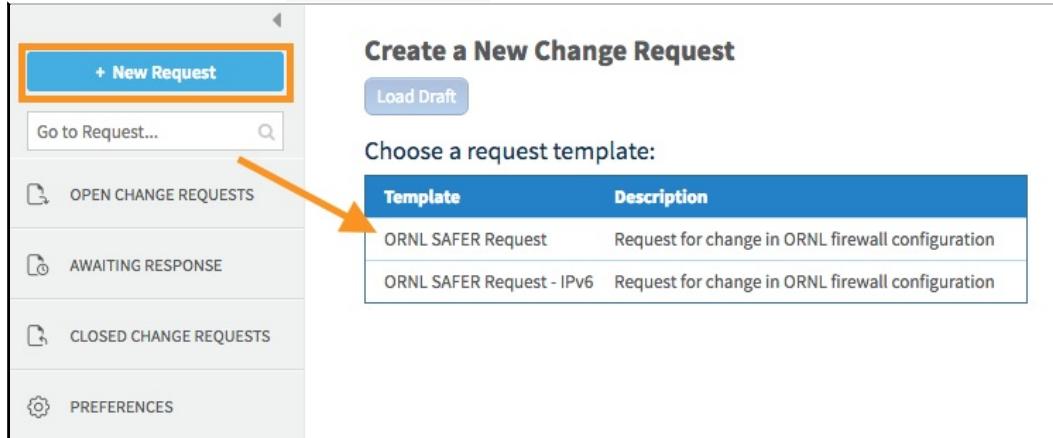
If you have further questions about getting your ORNL (SAFER) firewall rules in place, e-mail the SAFER team directly at [security@ornl.gov](mailto:security@ornl.gov).

If you wish to run services on your VM Instance that should be available outside of ORNL's network, ensure that you select the [External Network option](#) when setting up your VM Instance and that you also add a rule to your Security Group for that particular service. Read more about the CADES Birthright Cloud network design [here](#).

## Request a Firewall Exception

For the purposes of this procedure, we will set up an exception for a web server running on port 80.

1. Navigate to <https://safer.ornl.gov>.
2. Log in to the SAFER interface using your UCAMS credentials.
3. Click `+ New Request` at the top left of the screen.
4. On the resulting page, choose `ORNL SAFER Request`.



5. In the resulting request dialog, we need to fill out the following fields:
  - `Subject` – A simple subject will do. We're going with `VM web server`.
  - `Authorization` – Set to `None`.
  - `Change Request Justification` – Provide the reason for your firewall exception.
  - `Expires` – Leave this blank to make this exception indefinite. Otherwise, choose a date for the exception to expire.
  - `Source` – The source IP or IP range (with CIDR notation if range) of the server for which you would like to make an exception. We're going to use the single IP address of our VM Instance, `128.219.186.29`.
  - `Destination` – For this example, we're going to make the source and the destination the same IP address, `128.219.186.29`.
  - `Service` – This can be formatted using the protocol/port (e.g., TCP/80) or you can choose from a list of common multi-port services in the drop-down menu. We're using `TCP/80`.
  - `Service Name` – User-defined name of the rule. We're calling ours `blackmesa_web`.
6. Once filled out, click `Next` to submit your request.

**Create a New Change Request**

[Back](#) [Save Draft](#) [Next](#)

**General**

Subject	crawfordst@ornl.gov
Authentication	<input checked="" type="radio"/> None
	<input type="radio"/> RSA
	<input type="radio"/> Badge
	<input type="radio"/> UCAMS
	<input type="radio"/> XCAMS
	<input type="radio"/> DCAMS
Change request justification	Working with external developers who need to see and provide feedback on the website before it goes live on another server.
E-mail notification list (opt.)	

**Traffic \***

[Set traffic values](#) [Import traffic from csv](#)

Source	Destination	Service	Action
128.219.186.29	128.219.186.29	tcp/80	Allow
NAT settings			
Service Name blackmesa_web			

[+ Add more traffic](#)

[Back](#) [Save Draft](#) [Next](#)

You will receive an e-mail confirmation of your request. You can also view the status of your exception request at any time by logging into the [SAFER interface](#).

## Related Tutorials

- [Run a Web Server on Your VM Instance](#)

## Run a Simple Web Server

The following documentation will show you how to launch an Instance using the default Security Group. Once running, we will enable a basic HTTP service on port 80 by adding an additional Security Group.

## Launch an Instance Using Horizon

1. Navigate to the Horizon web interface at <https://cloud.cades.ornl.gov/>.
2. Log in with your UCAMS credentials.
  - Domain: ornl
  - User Name: Your three-letter UCAMS ID
  - Password: Your UCAMS password
3. Navigate to Project → Compute → Instances .
4. Click the Launch Instance button, and complete the launch instance wizard.
  - For this exercise use the Default security group.

If you have never launched a VM Instance before, check out the tutorial linked below before proceeding.

How to: [Launch a VM Instance from an Image](#)

## Add Rule for TCP Traffic

Once we have our VM Instance up and running, we need to make sure that the Instance can receive incoming traffic on port 80. We'll create a custom Security Group, add the required rule for HTTP traffic to our new (or existing) Security Group, and assign that Security Group to our VM Instance.

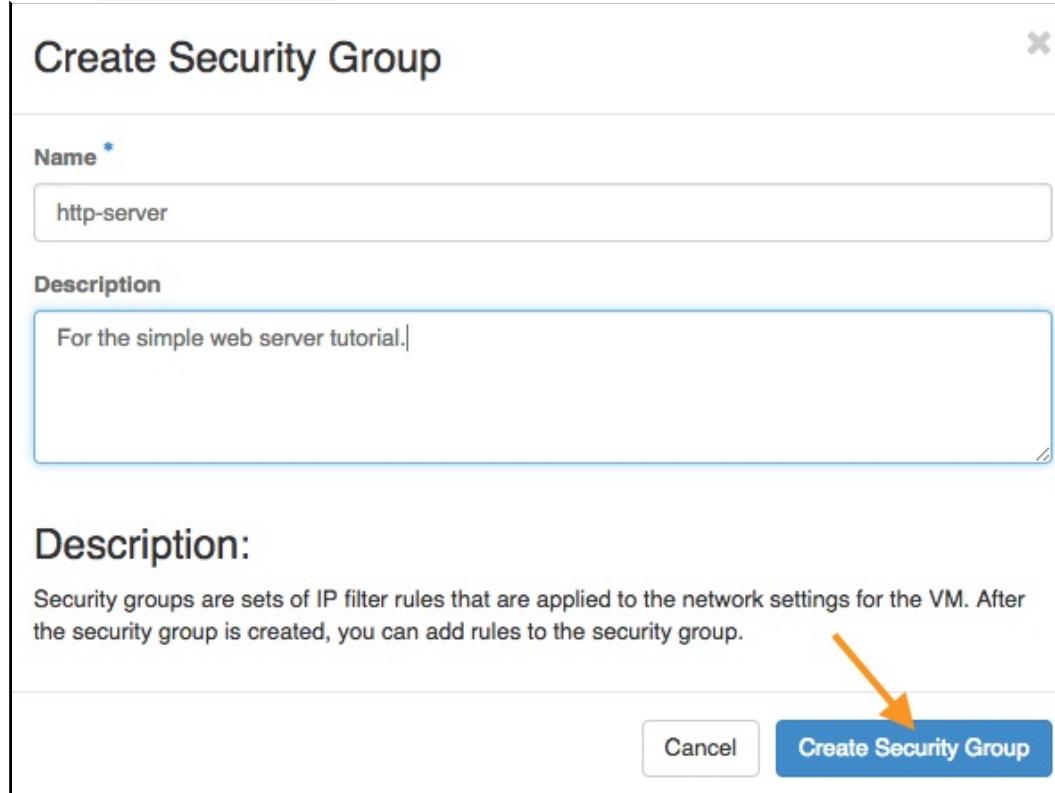
## Create Custom Security Group

1. Navigate to Project → Compute → Access & Security → Security Groups .
2. Click + create Security Group at the top right of the Security Groups table.

The screenshot shows the 'Access & Security' interface with the 'Security Groups' tab selected. At the top, there are tabs for 'Security Groups', 'Key Pairs', 'Floating IPs', and 'API Access'. Below these are buttons for 'Filter', a search icon, and '+ Create Security Group'. A red button labeled 'Delete Security Groups' is also visible. The main area displays a table with columns for 'Name', 'Description', and 'Actions'. One row is shown for a 'default' security group, which is described as a 'Default security group'. A 'Manage Rules' button is located in the 'Actions' column for this row. At the bottom of the table, it says 'Displaying 1 item'.

3. In the resulting dialog, fill out the Name and Description (optional) fields.
  - We're using http-server for this example.

- Click the `Create Security Group` button.



Your new Security Group should now be available in the Security Groups table.

Alternatively, you can modify an existing group using the corresponding `Manage Rules` button.

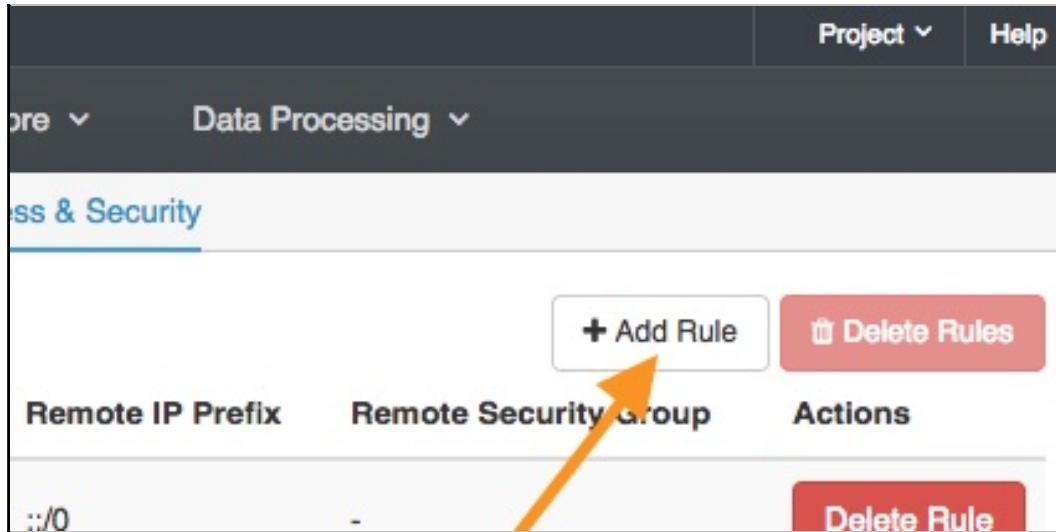
## Add Rule to Custom Security Group

- Find the newly created `http-server` rule on the `Security Groups` tab.

<b>Access &amp; Security</b>			
<b>Security Groups</b>		<b>Key Pairs</b>	<b>Floating IPs</b>
		<b>Filter</b>	<b>+ Create Security Group</b>
<input type="checkbox"/>	<b>Name</b>	<b>Description</b>	<b>Actions</b>
<input type="checkbox"/>	Custom	Custom security group	<b>Manage Rules</b> ▾
<input type="checkbox"/>	default	Default security group	<b>Manage Rules</b>
<input type="checkbox"/>	http-server	For the simple web server tutorial.	<b>Manage Rules</b> ▾

Displaying 3 items

- Click the `Manage Rules` button.
- On the Manage Rules screen, click the `Add Rule` button.



4. In the resulting dialog, click the drop-down field under `Rule`.
5. Choose the `HTTP` rule template from the drop-down list.
  - Choosing the HTTP template will automatically set the port to `80` and set the direction to `Ingress`.
6. In the `Remote` box, choose `CIDR` (preferred) and leave the field as `0.0.0.0/0` or enter the desired inter-domain range. See [CIDR examples](#) for more information.
7. Click `Add` at the bottom of the dialog box to implement your custom rule.

The screenshot shows the 'Add Rule' dialog box. On the left, there are input fields for 'Rule \*' (set to 'HTTP') and 'Remote \*' (set to 'CIDR'). Below these are descriptive sections: 'Description:' which explains what rules are and how they work, and 'Open Port/Port Range' and 'Remote' sections which provide specific instructions for those fields. At the bottom right of the dialog are 'Cancel' and 'Add' buttons, with an orange arrow pointing to the 'Add' button.

8. Confirm your new custom rule is displayed in the Manage Rules screen.

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

Displaying 3 items

## Add New Security Group to Your VM Instance

1. Navigate to `Project` → `Compute` → `Instances`, and find the Instance on which you would like to run your HTTP server.
2. Click the drop-down menu on your Instance's listing, and select `Edit Security Groups` from the resulting menu.

The screenshot shows the OpenStack Instances page. At the top, there are tabs for Overview, Instances, Volumes, Images, and Access & Security. Below the tabs, there is a search bar for 'Instance Name' and buttons for 'Launch Instance' and 'Delete Instances'. The main table lists two instances: 'aperture' and 'black-mesa'. For the 'black-mesa' instance, a context menu is open, with the 'Edit Security Groups' option highlighted and surrounded by an orange box. An orange arrow points from the 'Edit Security Groups' option to the 'Edit Security Groups' button in the 'Edit Instance' dialog.

<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"/>	aperture	-	128.219.186.46	m1.tiny	blackmesakey	Active	nova	None	Running	3 weeks, 5 days	<button>Create Snapshot</button>
<input type="checkbox"/>	black-mesa	-	128.219.186.42	m1.tiny	blackmesakey	Active	nova	None	Running	1 month	<button>Create Snapshot</button>

Displaying 2 items

3. In the resulting window, click `+` to add the `http-server` security group to your VM Instance.
4. Click `Save` to associate your `http-server` security group with your Instance.

The screenshot shows the 'Edit Instance' dialog. At the top, there are tabs for 'Information \*' and 'Security Groups', with 'Information \*' being active. Below the tabs, there is a note: 'Add and remove security groups to this instance from the list of available security groups.' There are two main sections: 'All Security Groups' and 'Instance Security Groups'. In the 'All Security Groups' section, there is a list with 'Custom' at the top and 'http-server' listed below it. An orange arrow points from the '+' button next to 'http-server' to the 'Save' button at the bottom right of the dialog. Another orange arrow points from the 'Save' button to the 'Save' button in the 'Edit Security Groups' dropdown menu on the Instances page.

## Connect to Your Instance Using SSH

1. Open a Bash terminal.

2. Execute `$ ssh cades@128.219.186.42` .
    - o Replace `128.219.186.42` with the IP address of your own Instance.
  3. You should now be connected to your VM Instance via SSH.

For more detailed information on connecting to your VM Instance using SSH, please see the [SSH tutorials](#), linked below.

## How to: Access Your VM Instance Using SSH

For Windows users, we have developed a separate tutorial that walks you through connecting to your VM Instance using PuTTY's SSH client.

How to: Access Your VM Instance Using PuTTY (Windows)

# Set up Your Web Server

For the web server, we need to create a web directory and populate it with a basic `index.html` page.

1. Execute the following in the VM's Bash terminal.

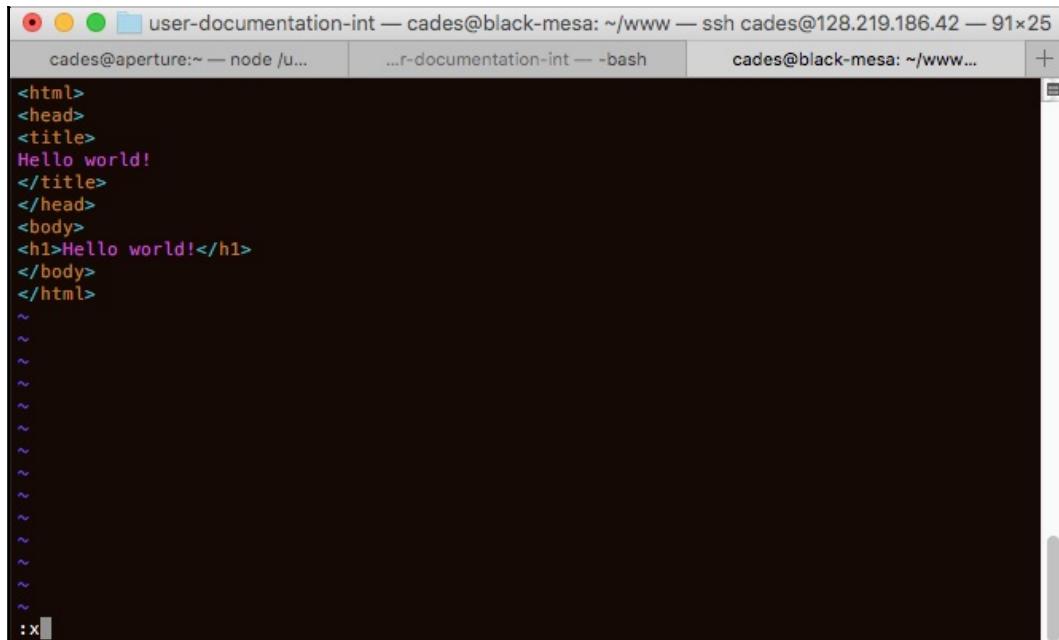
```
$ mkdir www  
$ cd www
```

This will create a `www` directory. Next, we need to open a new file (`index.html`) in the VIM text editor and populate it with some basic HTML markup.

2. In the Bash terminal, execute `$ vim index.html` to create the new file and open it in VIM.
  3. Once in VIM, add the following lines to your file.

```
<html>
<head>
<title>
Hello world!
</title>
</head>
<body>
<h1>Hello world!</h1>
</body>
</html>
```

4. Save your new file: press the `Esc` key, type `:x`, and then hit the `Return` key.

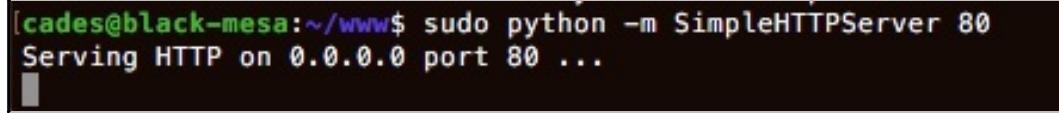


A screenshot of a terminal window titled "user-documentation-int — cades@black-mesa: ~/www — ssh cades@128.219.186.42 — 91x25". The window has three tabs: "index.html" (active), "...r-documentation-int — -bash", and "index.html" (background). The active tab displays the content of the "index.html" file, which contains the following HTML code:

```
<html>
<head>
<title>
Hello world!
</title>
</head>
<body>
<h1>Hello world!</h1>
</body>
</html>
~
```

Now that we have our files in place, we can start a simple web server using Python.

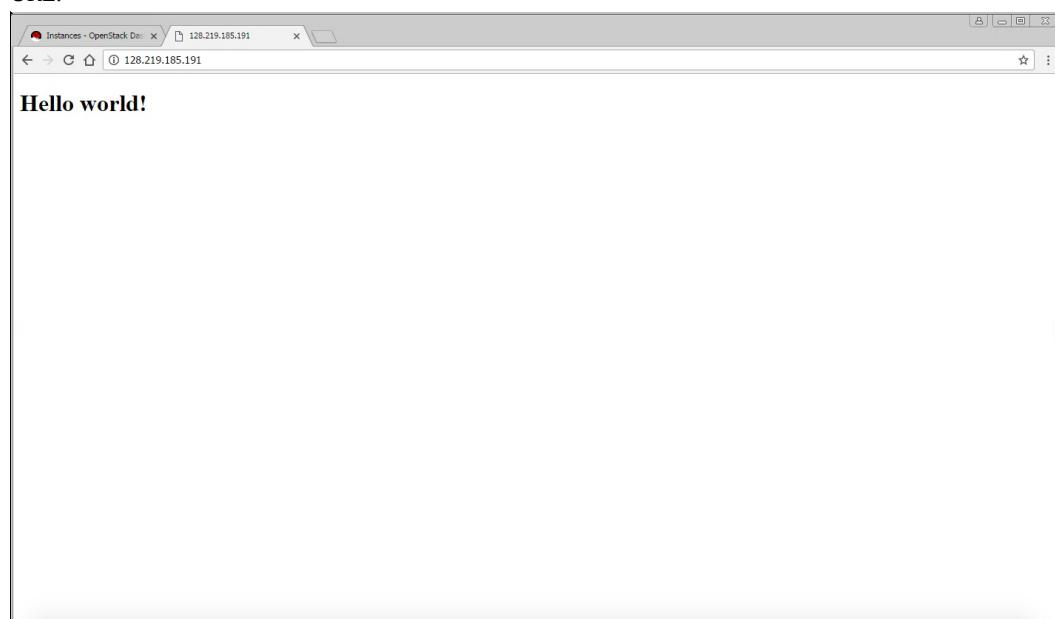
1. Ensure that you are still in the `www` directory where you placed your `index.html` file.
2. Execute `$ sudo python -m SimpleHTTPServer 80` in your VM's Bash terminal. A message will confirm that the server is running.



```
[cedes@black-mesa:~/www$ sudo python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
```

## Test the Web Server

You should now be able to access your "hello world" page from a web browser by entering the IP address of your instance as the URL.



## Troubleshooting

If you get an error when attempting to navigate to your Instance's IP using a browser, ensure that:

- You used the external network option when you configured your Instance.
- You are within the ORNL network or you have added a SAFER firewall exception for your VM Instance's IP address on Port 80 (or whatever port you specified in the Python command).
- You have a Rule in your Security Group that grants access to your VM Instance on Port 80 (or whatever port you specified in the Python command).
- Your Python server is active on your VM Instance.

## Related Tutorials

- [Launch a VM Instance](#)
- [Access VM Instance Using SSH](#)

# Adding a Security Certificate to Your Website

Obtaining a security certificate from a Certificate Authority enables you to use https on your website hosted by CADES. Utilizing https maintains the confidentiality of the transferred information by using a set of encryption keys. Additionally, this document will cover requirements for Firewall exceptions, cipher specifications, and HSTS preloading conditions.

## Caveat: Auto SSL Creation for CADES VMs

In order to expedite ORNL's required compliance with [DHS BOD 18-01](#) as many sites as possible were automatically adjusted to defaulting to HTTPS with a Let's Encrypt certificate.

### Directive Overview

Pursuant to [DHS BOD 18-01](#) all federal websites must be migrated to HTTPS. The prescribed implementation pattern is as follows

- Redirect HTTP traffic to HTTPS via [301 Moved Permanently](#)
- HTTPS should be configured with a trusted TLS certificate
- All of the following should be disallowed:
  - SSL v2
  - SSL v3
  - RC4, DES, and 3DES ciphers
- [HSTS](#) header set with `max-age` equal to 1 year.

## Table of Contents

### Prerequisites

#### Part 1: Modifying the OpenStack Horizon Security Group

- Add Rule for Secured TCP Traffic
- Include new rule in your Instance's Security Group

#### Part 2: Obtaining the Security Certificate for your Site

- Accessing your VM via SSH
- Using Certbot's automated client (Ubuntu or CentOS)
- Setting Up Autorenewal

#### Part 3: Updating Firewall Exception

- Requesting new firewall exception

#### Part 4: Configuring your Webserver

#### Part 5: Certificate Auto-Renewal

---

---

## Prerequisites

- A website hosted by CADES resources
- Bash access to the VM hosting the site
- Enable virtual hosts in your apache config file and ensure the default site is served as a virtual host.
  - [CentOS Apache Virtual Host Configuration](#)
  - [Ubuntu Apache Virtual Host Configuration](#)

## Part 1: Modifying the OpenStack Horizon Security Group

### View the Instance Using Horizon

1. Navigate to the Horizon web interface at <https://cloud.cades.ornl.gov/>.
2. Log in with your UCAMS credentials.
  - Domain: ornl
  - User Name: Your three-letter UCAMS ID
  - Password: Your UCAMS password
3. Navigate to Project → Compute → Instances .
4. Click on the existing instance that hosts your web server if you'd like to review the settings.

If you have never launched a VM Instance before, check out the tutorial linked below before proceeding.

How to: [Launch a VM Instance](#)

### Add Rule for Secured TCP Traffic

When you originally set up your web server, the settings allowed incoming traffic on port 80. Now we need to allow secure traffic. Then we'll make sure the rule is properly assigned to your Instance.

1. Navigate to Project → Compute → Access & Security → Security Groups .
2. Find the existing Security Group that contains your http access rules and click Manage Rules on the right side of the screen.
3. In the resulting window, click + Add Rule .
4. In the resulting dialog, click the drop-down field under Rule .
5. Choose the HTTPS rule template from the drop-down list.
  - Choosing the HTTPS template will automatically set the port to 443 and set the direction to Ingress .
6. In the Remote box, choose CIDR (preferred) and leave the field as 0.0.0.0/0 or enter the desired inter-domain range. See [CIDR examples](#) for more information.
7. Click Add at the bottom of the dialog box to implement your custom rule.
8. Confirm your new custom rule is displayed in the Manage Rules screen.

More info: [Security Groups](#)

### Add Rule to the Instance's Security Group

1. Navigate to Project → Compute → Instances .
2. On the right, click the down arrow () next to Create Snapshot .
3. Select Manage Security Groups .
4. Click the + next to your new HTTPS rule to add it to your Instance.

## Part 2: Obtaining the Security Certificate for your Site

The process of obtaining the SSL certificate is automated via the command line using Let's Encrypt and Certbot.

# Access your VM via ssh.

1. Open a Bash terminal.
  2. Execute `ssh cades@128.219.186.42` .
    - o Replace `128.219.186.42` with the IP address of your own Instance.
  3. You should now be connected to your VM Instance via SSH.

For more detailed information on connecting to your VM Instance using SSH, please see the [SSH tutorials](#), linked below.

## How to: Access Your VM Instance Using SSH

For Windows users, we have developed a separate tutorial that walks you through connecting to your VM Instance using PuTTY's SSH client.

How to: Access Your VM Instance Using PuTTY (Windows)

## **Installing Certbot for Ubuntu (see below for CentOS)**

Certbot is an automated client that obtains and implements SSL certificates for your website.

- #### 1. Obtain the necessary Certbot packages.

```
sudo apt-get update  
sudo apt-get install software-properties-common  
sudo add-apt-repository ppa:certbot/certbot  
sudo apt-get update  
sudo apt-get install python-certbot-apache
```

- ## 2. Use Certbot's Apache plugins to automate the certificate process.

- There are two options. Option A should be used if *cannot* stop and restart your web server. Option B should be used if you *can* start and stop your web server. If you are not sure, choose Option A.

- Option A

```
sudo certbot --authenticator webroot --installer apache
```

- Option B

```
sudo certbot --authenticator standalone --installer apache --pre-hook "apachectl -k stop" --post-hook
```

- ```
"apachectl -k start"
```
- You will be asked to enter your domain name (e.g. `example.com`) and your webroot (the folder where your website lives, e.g. `www`).

## Installing Certbot for CentOS (see above for Ubuntu)

Certbot is an automated client that obtains and implements SSL certificates for your website.

- Obtain the necessary Certbot packages.

```
wget https://dl.eff.org/certbot-auto
chmod a+x certbot-auto
```

- Use Certbot's plugins to automate the certificate process. (Fill in the `/path/to/` with your specific path.)

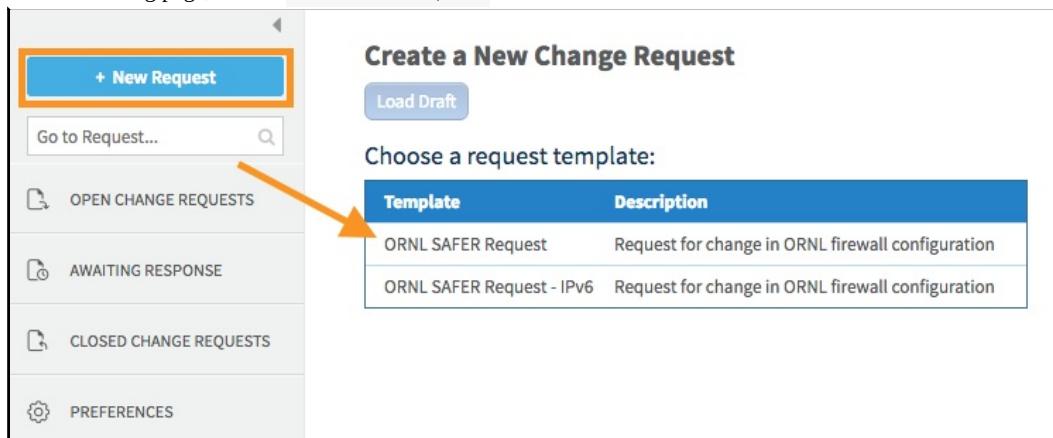
```
sudo ./path/to/certbot-auto certonly
```

- Select `2` when prompted to place Certbot's files in the webroot directory.
- Please also provide your email address and your domain name, e.g. `example.com`.
- Next, you'll be asked to provide the webroot for your domain (the folder where your website lives), e.g. `www`.
- The next command will stop and restart your server. Please be aware of the temporary service interruption.

```
sudo certbot --authenticator standalone --installer apache --pre-hook "apachectl -k stop" --post-hook
"apachectl -k start"
```

## Part 3: Updating Firewall Exception (for External-Facing Sites)

- Navigate to <https://safer.ornl.gov>.
- Log in to the SAFER interface using your UCAMS credentials.
  - User Name: Your three-letter UCAMS ID
  - Password: Your UCAMS password
- Click `+ New Request` at the top left of the screen.
- On the resulting page, choose `ORNL SAFER Request`.



- In the resulting request dialog, we need to fill out the following fields:
  - `Subject` – A simple subject will do. We're going with `VM web server`.
  - `Authorization` – Set to `None`.
  - `Change Request Justification` – Provide the reason for your firewall exception.

- **Expires** – Leave this blank to make this exception indefinite. Otherwise, choose a date for the exception to expire.
  - **Source** – The source IP or IP range (with CIDR notation if range) of the server for which you would like to make an exception. We're going to use the single IP address of our VM Instance, 128.219.186.29.
  - **Destination** – For this example, we're going to make the source and the destination the same IP address, 128.219.186.29.
  - **Service** – This can be formatted using the protocol/port (e.g., TCP/443) or you can choose from a list of common multi-port services in the drop-down menu. Please use TCP/443 as the secure traffic port.
  - **Service Name** – User-defined name of the rule. We're calling ours `blackmesa_web`.
6. Once filled out, click `Next` to submit your request.

**Create a New Change Request**

**General**

Subject: VM web server

Requestor: crawfordst@ornl.gov

Authentication: None

Expires:

Change request justification: Working with external developers who need to see and provide feedback on the website before it goes live on another server.

E-mail notification list (opt.):

**Traffic \***

| Source         | Destination    | Service | Action |
|----------------|----------------|---------|--------|
| 128.219.186.29 | 128.219.186.29 | tcp/80  | Allow  |

Service Name: blackmesa\_web

**Next**

You will receive an e-mail confirmation of your request. You can also view the status of your exception request at any time by logging into the [SAFER interface](#).

## Part 4: Webserver configuration

Your webserver will need to be configured to do the following:

- Redirect HTTP traffic to HTTPS
- Set up the HSTS header and preloading for external-facing sites ([see ORNL documentation](#))
- Use the TLS certificate designated for your site
- Allow only approved ciphers ([see ORNL documentation](#))

The exact details will vary between different web and application servers, but Mozilla provides a simple interactive tool that can help you get started:

<https://mozilla.github.io/server-side-tls/ssl-config-generator/>

## Part 5: Certificate Auto-Renewal

Using a simple script we can automate the certificate renewal process. Although this certificate lasts for 90 days, running this cron job often will ensure your certificate stays up to date.

```
@weekly python -c 'import random; import time; time.sleep(random.random() * 3600)' && certbot renew
```

If you find that you need more guidance setting up automatic renewal, you can visit [this site](#).

## Install CPUID

Installing CPUID on your VM Instance is a fairly simple task. Once connected to your Instance using SSH, use the Bash terminal to install and run CPUID. Below you will find a procedure for each CADES-provided operating system.

## Prerequisites

To install CPUID on your VM Instance, you must first have access to your Instance through SSH ([instructions](#)).

## Ubuntu

1. Update Ubuntu.

```
$ sudo apt-get update
```

2. Install CPUID using the package manager.

```
$ sudo apt-get install cpuid
```

3. CPUID is now installed. To run CPUID, execute the following.

```
$ cpuid
```

## CentOS

1. Update CentOS.

```
$ sudo yum check-update
```

2. Install CPUID using the package manager.

```
$ sudo yum install cpuid
```

3. When prompted, confirm that you wish to install CPUID. `$ Is this ok [y/d/N]: y`

4. CPUID is now installed. To run CPUID, execute the following.

```
$ cpuid
```

## Supported CPUID Codes

- CPUID Codes supported by Hypervisor CPU (Not all in Birthright)
  - fpu: Onboard FPU (floating point support)
  - eagerfpu: Non lazy FPU restore
  - vme: Virtual 8086 mode enhancements
  - de: Debugging Extensions (CR4.DE)
  - smx: Safer mode: TXT (TPM support)
  - pse: Page Size Extensions (4MB memory pages)
  - tsc: Time Stamp Counter (RDTSC)
  - constant\_tsc: TSC ticks at a constant rate
  - nonstop\_tsc: TSC does not stop in C states
  - ptsc: performance time-stamp counter
  - msr: Model-Specific Registers (RDMSR, WRMSR)
  - nodeid\_msr: NodeId MSR
  - pae: Physical Address Extensions (support for more than 4GB of RAM)
  - mce: Machine Check Exception
  - cx8: CMPXCHG8 instruction (64-bit compare-and-swap)
  - apic: Onboard APIC
  - x2apic: x2APIC
  - extapic: Extended APIC space
  - sep: SYSENTER/SYSEXIT
  - mttr: Memory Type Range Registers
  - k6\_mttr: AMD K6 nonstandard MTRRs
  - pge: Page Global Enable (global bit in PDEs and PTEs)
  - mca: Machine Check Architecture
  - smca: Scalable MCA
  - cmov: CMOV instructions (conditional move) (also FCMOV)
  - pat: Page Attribute Table
  - pse36: 36-bit PSEs (huge pages)
  - clflush: Cache Line Flush instruction
  - dts: Debug Store (buffer for debugging and profiling instructions)
  - acpi: ACPI via MSR (temperature monitoring and clock speed modulation)
  - mmx: Multimedia Extensions
  - cxmmx: Cyrix MMX extensions
  - fxsr: FXSAVE/FXRSTOR, CR4.OSFXSR
  - sse: Intel SSE vector instructions
  - misalignsse: indicates if a general-protection exception (#GP) is generated when some legacy SSE instructions operate on unaligned data. Also depends on CR0 and Alignment Checking bit
  - sse2: SSE2
  - ss: CPU self snoop
  - ht: Hyper-Threading
  - tm: Automatic clock control (Thermal Monitor)
  - rtm: Restricted Transactional Memory
  - pbe: Pending Break Enable (PBE# pin) wakeup support
  - syscall: SYSCALL (Fast System Call) and SYSRET (Return From Fast System Call)
  - nx: Execute Disable
  - pdpe1gb: One GB pages (allows hugepagesz=1G)
  - rdtscp: Read Time-Stamp Counter and Processor ID

- lm: Long Mode (x86-64: amd64, also known as Intel 64, i.e. 64-bit capable)
- lahf\_lm: Load AH from Flags (LAHF) and Store AH into Flags (SAHF) in long mode
- constant\_tsc: TSC ticks at a constant rate
- arch\_perfmon: Intel Architectural PerfMon
- pebs: Precise-Event Based Sampling
- bts: Branch Trace Store
- rep\_good: rep microcode works well
- nopl: The NOPL (0F 1F) instructions
- xtopology: cpu topology enum extensions
- nonstop\_tsc: TSC does not stop in C states
- aperfmpf: APERFMPERF
- eagerfpu: Non lazy FPU restore
- pni: SSE-3 (“Prescott New Instructions”)
- pclmulqdq: Perform a Carry-Less Multiplication of Quadword instruction — accelerator for GCM
- dtes64: 64-bit Debug Store
- monitor: Monitor/Mwait support (Intel SSE3 supplements)
- ds\_cpl: CPL Qual. Debug Store
- vmx: Hardware virtualization: Intel VMX
- smx: Safer mode: TXT (TPM support)
- est: Enhanced SpeedStep
- tm2: Thermal Monitor 2
- ssse3: Supplemental SSE-3
- fma: Fused multiply-add
- cx16: CMPXCHG16B
- xtpr: Send Task Priority Messages
- pdcm: Performance Capabilities
- pcid: Process Context Identifiers
- invpcid: Invalidate Processor Context ID
- dca: Direct Cache Access
- sse4\_1: SSE-4.1
- sse4\_2: SSE-4.2
- x2apic: x2APIC
- movbe: Move Data After Swapping Bytes instruction
- popcnt: Return the Count of Number of Bits Set to 1 instruction (Hamming weight, i.e. bit count)
- tsc\_deadline\_timer: Tsc deadline timer
- xsave: Save Processor Extended States: also provides XGETBY,XRSTOR,XSETBY
- avx: Advanced Vector Extensions
- f16c: 16-bit fp conversions (CVT16)
- rdrand: Read Random Number from hardware random number generator instruction
- lahf\_lm: Load AH from Flags (LAHF) and Store AH into Flags (SAHF) in long mode
- abm: Advanced Bit Manipulation
- ida: Intel Dynamic Acceleration
- arat: Always Running APIC Timer
- epb: IA32\_ENERGY\_PERF\_BIAS support
- pln: Intel Power Limit Notification
- pts: Intel Package Thermal Status
- tpr\_shadow: Intel TPR Shadow
- vnmi: Intel Virtual NMI
- flexpriority: Intel FlexPriority
- ept: Intel Extended Page Table
- vpid: Intel Virtual Processor ID

- fsgsbase: {RD/WR}{FS/GS}BASE instructions
- tsc\_adjust: TSC adjustment MSR
- bmi1: 1st group bit manipulation extensions
- avx2: AVX2 instructions
- smep: Supervisor Mode Execution Protection
- bmi2: 2nd group bit manipulation extensions
- erms: Enhanced REP MOVSB/STOSB
- invpcid: Invalidate Processor Context ID
- cqm: Cache QoS Monitoring
- xsaveopt: Optimized XSAVE
- cqm\_llc: LLC QoS
- cqm\_occup\_llc: LLC occupancy monitoring

## Supported CPUID Instruction Set Codes

- CPUID Instruction Set Codes Supported by VM Instances
  - fpu: Onboard FPU (floating point support)
  - eagerfpu: Non lazy FPU restore
  - vme: Virtual 8086 mode enhancements
  - de: Debugging Extensions (CR4.DE)
  - smx: Safer mode: TXT (TPM support)
  - pse: Page Size Extensions (4MB memory pages)
  - tsc: Time Stamp Counter (RDTSC)
  - constant\_tsc: TSC ticks at a constant rate
  - nonstop\_tsc: TSC does not stop in C states
  - ptsc: performance time-stamp counter
  - msr: Model-Specific Registers (RDMSR, WRMSR)
  - nodeid\_msr: NodeId MSR
  - pae: Physical Address Extensions (support for more than 4GB of RAM)
  - mce: Machine Check Exception
  - cx8: CMPXCHG8 instruction (64-bit compare-and-swap)
  - apic: Onboard APIC
  - x2apic: x2APIC
  - extapic: Extended APIC space
  - sep: SYSENTER/SYSEXIT
  - mttr: Memory Type Range Registers
  - k6\_mttr: AMD K6 nonstandard MTRRs
  - pge: Page Global Enable (global bit in PDEs and PTEs)
  - mca: Machine Check Architecture
  - smca: Scalable MCA
  - cmov: CMOV instructions (conditional move) (also FCMOV)
  - pat: Page Attribute Table
  - pse36: 36-bit PSEs (huge pages)
  - clflush: Cache Line Flush instruction
  - mmx: Multimedia Extensions
  - cxmmx: Cyrix MMX extensions
  - fxsr: FXSAVE/FXRSTOR, CR4.OSFXSR
  - sse: Intel SSE vector instructions
  - misalignsse: indicates if a general-protection exception (#GP) is generated when some legacy SSE instructions operate on unaligned data. Also depends on CR0 and Alignment Checking bit
  - sse2: SSE2
  - ss: CPU self snoop
  - syscall: SYSCALL (Fast System Call) and SYSRET (Return From Fast System Call)
  - nx: Execute Disable
  - pdpe1gb: One GB pages (allows hugepagesz=1G)
  - rdtscp: Read Time-Stamp Counter and Processor ID
  - lm: Long Mode (x86-64: amd64, also known as Intel 64, i.e. 64-bit capable)
  - lahf\_lm: Load AH from Flags (LAHF) and Store AH into Flags (SAHF) in long mode
  - constant\_tsc: TSC ticks at a constant rate
  - rep\_good: rep microcode works well
  - nopl: The NOPL (0F 1F) instructions
  - eagerfpu: Non lazy FPU restore

- pni: SSE-3 (“Prescott New Instructions”)
- pclmulqdq: Perform a Carry-Less Multiplication of Quadword instruction — accelerator for GCM)
- ssse3: Supplemental SSE-3
- fma: Fused multiply-add
- cx16: CMPXCHG16B
- pcid: Process Context Identifiers
- invpcid: Invalidate Processor Context ID
- sse4\_1: SSE-4.1
- sse4\_2: SSE-4.2
- x2apic: x2APIC
- movbe: Move Data After Swapping Bytes instruction
- popcnt: Return the Count of Number of Bits Set to 1 instruction (Hamming weight, i.e. bit count)
- tsc\_deadline\_timer: Tsc deadline timer
- xsave: Save Processor Extended States: also provides XGETBY,XRSTOR,XSETBY
- avx: Advanced Vector Extensions
- f16c: 16-bit fp conversions (CVT16)
- rdrand: Read Random Number from hardware random number generator instruction
- hypervisor: Running on a hypervisor
- lahf\_lm: Load AH from Flags (LAHF) and Store AH into Flags (SAHF) in long mode
- abm: Advanced Bit Manipulation
- fsgsbase: {RD/WR}{FS/GS}BASE instructions
- bmi1: 1st group bit manipulation extensions
- avx2: AVX2 instructions
- smep: Supervisor Mode Execution Protection
- bmi2: 2nd group bit manipulation extensions
- erms: Enhanced REP MOVSB/STOSB
- invpcid: Invalidate Processor Context ID
- xsaveopt: Optimized XSAVE

CADES → User Documentation → User-Provided Tutorials for Birthright Cloud → Launch a Docker Container

Tutorial contributed by Drew Schmidt. **Note:** User-provided tutorials are not supported by CADES.

## Background

Docker is a [container](#) architecture and ecosystem. A [linux.com article](#) nicely summarizes Docker as follows:

Docker is a tool that can package an application and its dependencies in a virtual container that can run on any Linux server. This helps enable flexibility and portability on where the application can run, whether on premises, public cloud, private cloud, bare metal, etc.

Containers have somewhat similar goals to a virtual machine (vm). However, a Docker container is not a vm. You are probably aware that vm's have some performance overhead compared to running things natively. However, it is worth noting that the applications that run inside of Docker containers actually run *natively*. Your Docker containers share the kernel with their host operating system. So there is no double overhead in running a container inside our vm. However, we still suffer some performance penalty by having virtualized in the first place.

Many of the applications you will be interested in deploying are already configured for very easy use with Docker. You can find public repositories of many of your favorite applications set up on [Docker Hub](#).

## First Steps

We will assume that you are already reasonably familiar with the [Birthright Cloud](#) system. If not, consider reading the [user documentation](#). Make sure you read the section titled: [Launch a VM Instance from an Image](#)

However, you get there, launch an Ubuntu 16.04 vm.

| Image Name                    | Type  | Status | Public | Protected | Format | Size   | Actions                 |
|-------------------------------|-------|--------|--------|-----------|--------|--------|-------------------------|
| CADES_CentOS-7.3_v20170310_0  | Image | Active | Yes    | No        | QCOW2  | 1.4 GB | <button>Launch</button> |
| CADES_Ubuntu16.04_v20170302_0 | Image | Active | Yes    | No        | QCOW2  | 1.4 GB | <button>Launch</button> |

Displaying 2 items

For now, you can keep the setup very basic, just following the instructions outlined in the "Launch a VM Instance" page linked above.

You may eventually be interested in more complicated configurations. For example, you may need to modify the security group details (say, for example, you want to run a docker container that runs a web server). For now, we will ignore those details. However, we provide a more complicated example in our Shiny tutorial.

## Install Docker

Next, you need to [ssh to your new vm](#). I named my instance `t1`, so when I login, it shows me as `cedes@t1`. Your prompt will show `cedes@whatever-you-named-your-vm`.

Having logged in, it's time to install Docker. The official Docker documentation [provides a lot of useful information](#) to this end. Below we summarize only the steps outlined in that article. If you wish to understand an individual step or if something goes wrong, please refer to the article.

Otherwise, run:

```
sudo apt-get install \
    apt-transport-https \
    ca-certificates \
    curl \
    software-properties-common

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

sudo apt-key fingerprint 0EBFCD88

sudo add-apt-repository \
    "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
    $(lsb_release -cs) \
    stable"

sudo apt-get update
sudo apt-get install -y docker docker.io
```

And if all is well, you should have Docker installed on your vm.

## Run a Test Container

While still ssh'd to your Birthright Cloud vm, you can test that your setup is working correctly by running:

```
sudo docker pull hello-world
sudo docker run hello-world
```

If all goes well, you will have a small "hello world"-like output and return to your terminal, and should look something like this:

```
cade$@t1:~$ sudo docker run hello-world
Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://cloud.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/engine/userguide/
```

cade\$@t1:~\$ █

That's it!

CADES → User Documentation → User-Provided Tutorials for Birthright Cloud → Launch Shiny within Docker

Tutorial contributed by Drew Schmidt. **Note:** User-provided tutorials are not supported by CADES.

## Background

[Shiny](#) is a web app framework for the statistical programming language R. And while that might sound crazy, it actually works very well!

Shiny is generally pretty easy to develop locally, but deploying it in Shiny Server can be a bit of a headache if you go about it the wrong way. This guide should help make the process as painless as possible.

## First Steps

For simplicity, we will be using Shiny inside of a Docker container. Since shiny is a web app, that might lead you to think this process is going to be considerably more complicated than running it natively. However, I can assure you that this is not the case, particularly since the [Rocker Project](#) has handled most of the configuration details for us. They provide a [shiny container](#) already set up for business. However, if you wish to install Shiny server natively for some reason, then you will find some useful details in their [Dockerfile](#).

So the first step is to set up an Ubuntu 16.04 vm on the Birthright Cloud with Docker configured. See the Docker tutorial to learn how to set up Docker.

## Open Port(s)

Since Shiny is a web app, we need to open some ports on the vm so that we can access it from a laptop. We also have to do something similar for the Docker container so it can communicate with the vm, but that is actually much easier (and explained in the next section).

We will be using port 80. If you need to open another port, you can, but it's not as simple because port 80 is a pre-configured rule set for you. The [Run a Simple Web Server](#) guide from the official Birthright Cloud documentation is quite helpful here.

The condensed version is you want to navigate to Access & Security and:

1. click Create Security Group and give it a name (I called mine `shiny`)
2. click Manage Rules
3. click Add Rule
4. select http (this will open port 80; if you want another port, set up a custom tcp rule)
5. navigate to your instance (under the Instances tab)
6. click the triangle button next to Create Snapshot, select Edit Security Groups
7. add your new security group by clicking the blue "plus" button

That may sound like a lot or feel a bit overwhelming, but it's not so bad. Give it a try!

## Run an Existing Shiny Container

### Example 1: k-means Demo

Ok, we're finally ready to start talking Shiny. Assuming you've got your vm with Docker running, we'll first run some example (already configured and built) apps.

The first is a simple demonstration of k-means, and is an official [Shiny Gallery](#) example. It is [already set up](#) on Docker Hub, so standing this up is a breeze. Simply ssh to your vm and run:

```
sudo docker pull wrathematics/shinykmeans
sudo docker run -i -t -p 80:3838 wrathematics/shinykmeans
```

If you chose a port other than 80 when setting up the security groups, then you will need to change the `80` above accordingly.

Now just point your web browser to the IP of your vm, which you can find from the Compute -> Instance tab on the Birthright Cloud dashboard. Example screenshot below (with actual IP addresses masked)

| Instance Name | Image Name                    | IP Address                                    | Size     | Key Pair | Status | Availability Zone | Task | Power State | Time since created  |
|---------------|-------------------------------|-----------------------------------------------|----------|----------|--------|-------------------|------|-------------|---------------------|
| t1            | CADES_Ubuntu16.04_v20170302_0 | [REDACTED]<br>or_provider_general_intnetwork1 | m1.small | mk       | Active | nova              | None | Running     | 2 hours, 46 minutes |

So if your ip is `1.2.3.4`, then you just need to go to `http://1.2.3.4`. If you chose to use a port other than 80, then you will need to append a colon `:` and that port to the end of your ip address. So if you chose port `5555`, then you would go to `http://1.2.3.4:5555`.

## Example 2: Plot Builder

Something that shows off the power of R and Shiny a bit more is the app "ggplotwithyourdata". For your convenience, this too has been [set up](#) on Docker Hub, and is similarly easy to use:

```
sudo docker pull wrathematics/ggplotwithyourdata-docker
sudo docker run -i -t -p 80:3838 wrathematics/ggplotwithyourdata-docker
```

As before, just point your web browser to your vm's url and you're good to go.

## Deploying Your Own Shiny App

Fortunately or unfortunately (depending on your perspective), this is the most tricky part. In truth, this goes a bit beyond the scope of this document. Only you really know the dependencies of your app, so we can't tell you exactly what to do.

But we can give some general advice:

1. You will need to create your own Dockerfile. This guide is also not the right place to learn all about the various options of Dockerfiles; [this is](#). However, generally speaking, a Dockerfile is not that far removed from a shell script. So knowing how to build the app natively is important (and really, no different) in understanding how to get it to build in the container.

2. See how things are done on existing Dockerfile configurations. The k-means configuration is available [here](#) and the plot one [here](#). These have been deliberately kept fairly simple.
3. Installing R packages from source is often much harder and always *much* more time consuming than installing binary packages. When you run `install.packages()` or `devtools::install_github()` or the like, you are installing a source package. However, there are many binary packages available in `apt`. These packages all have the prefix `r-cran-`, so you should check what is available before going the `install.packages()` route. However, sometimes the `r-cran-` packages are out of date, so check before you install.
4. You can run multiple apps in the same Docker container. You just place each one in its own subdirectory of `/srv/shiny-server`. So if you have apps `foo` and `bar` that you want to host in the same container (say they have very similar dependencies), you would put them in `/srv/shiny-server/foo/` and `/srv/shiny-server/bar/`. Then you would view them at `http://1.2.3.4/foo` and `http://1.2.3.4/bar` (replace the fake ip with your real one of course!).
5. For troubleshooting Shiny Server problems, see the [Shiny Server Open Source Administrator's Guide](#).

Tutorial contributed by Fengming Yuan. **Note:** User-provided tutorials are not supported by CADES.

## Using the Eclipse IDE

NOTE: Here is an example of create new fortran project, e.g. PFLOTRAN

### STEP 1. Log into CADES

by X-windows from your local terminal (e.g. XQuartz on Mac OSX, or Cygwin-X on Windows OS).

```
ssh -X or-condo-login02.ornl.gov
(input ucams/xcams id and pwd)
```

```
f9y@or-condo-login02:~$ ssh -X or-condo-login02.ornl.gov
key_load_public: invalid format
f9y@or-condo-login02.ornl.gov's password:
Last login: Tue Oct  3 10:43:00 2017 from dyn01015919673.dz.ornl.gov
[f9y@or-condo-login02 ~]$
```

### STEP 2. START eclipse

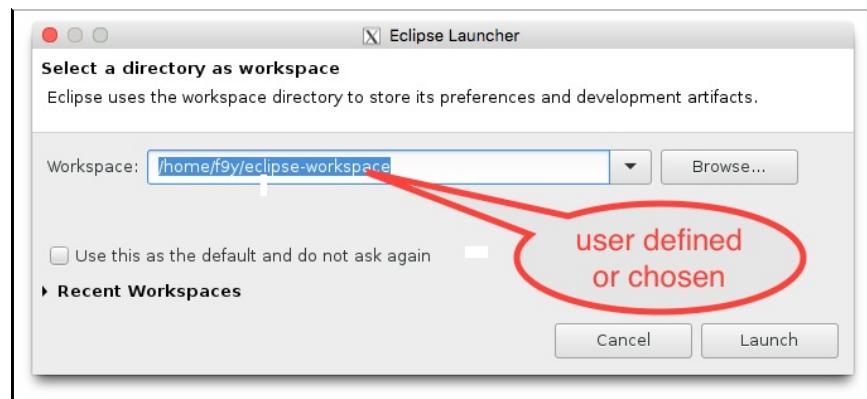
(note: you may create a link, such as optional command in the following, or add `export eclipse=/software/user_tools/current/cades-ccsi/eclipse` in `.bashrc` )

```
$ export eclipse=/software/user_tools/current/cades-ccsi/eclipse
(note: this is optional, otherwise, you can directly start the program)
```

```
[f9y@or-condo-login02 ~]$ export eclipse=/software/user_tools/current/cades-ccsi/eclipse
[f9y@or-condo-login02 ~]$ $eclipse
```

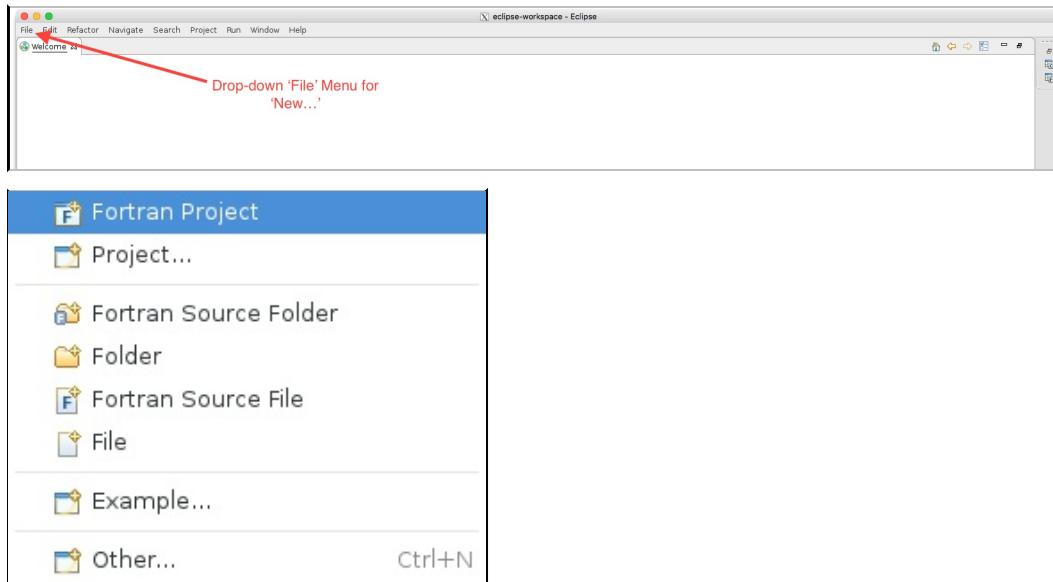
You may add this to your .bashrc, then each time you log in, 'eclipse' can be directly used as a command

```
$eclipse
#(this will start ECLIPSE in GUI shown in your desktop/laptop)
```

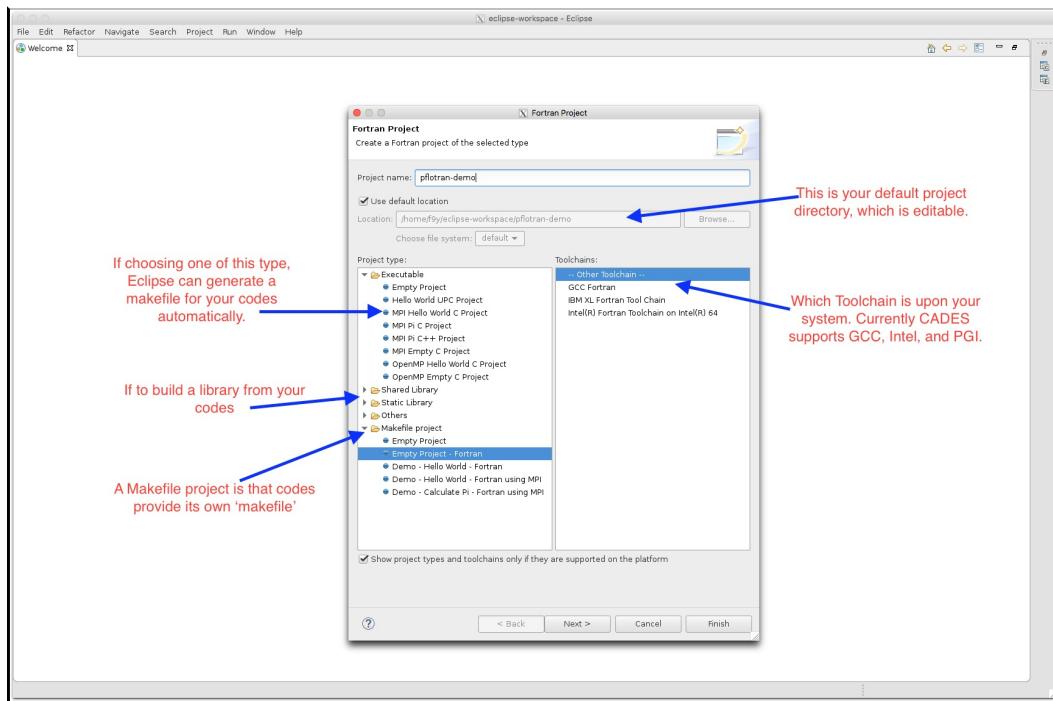


## STEP 3. Create a project

**WELCOME page (if first time) & FILE DROP-DOWN menu:**

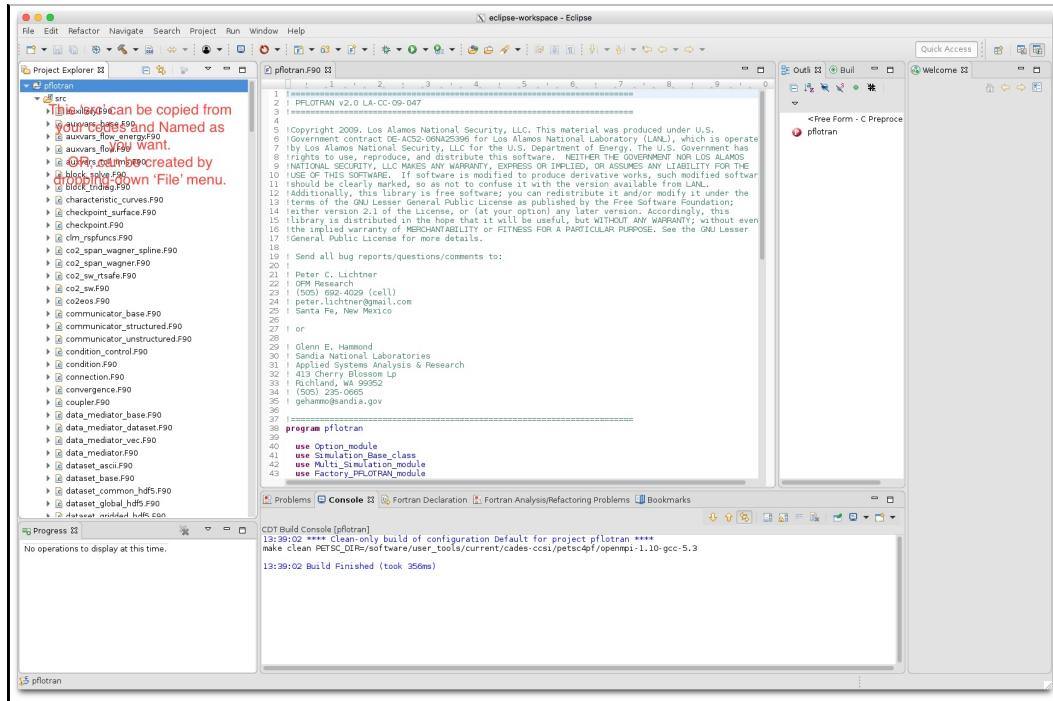


**NEW ... PROJECT configuration:**

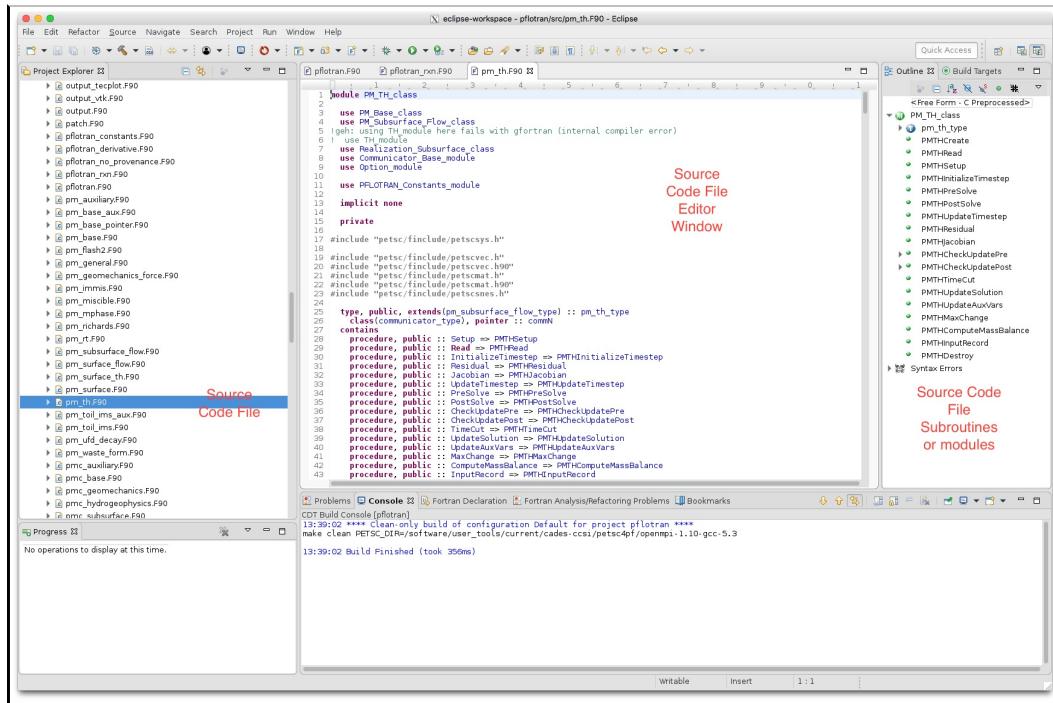


**GENERIC Interface/windows of ECLIPSE:**

(note on how to add/create source codes)



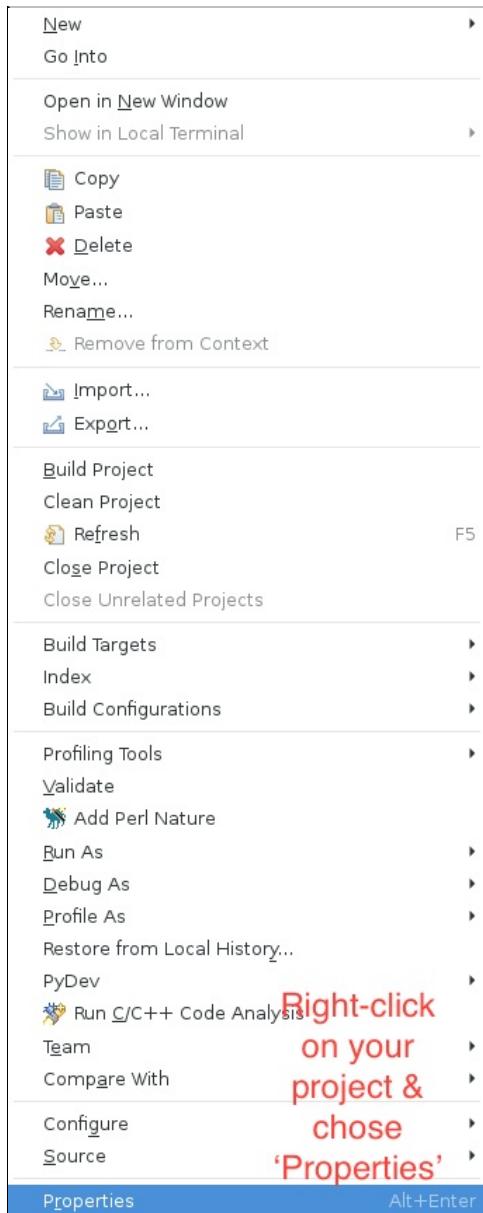
MORE on windows...



## STEP 4. Build project

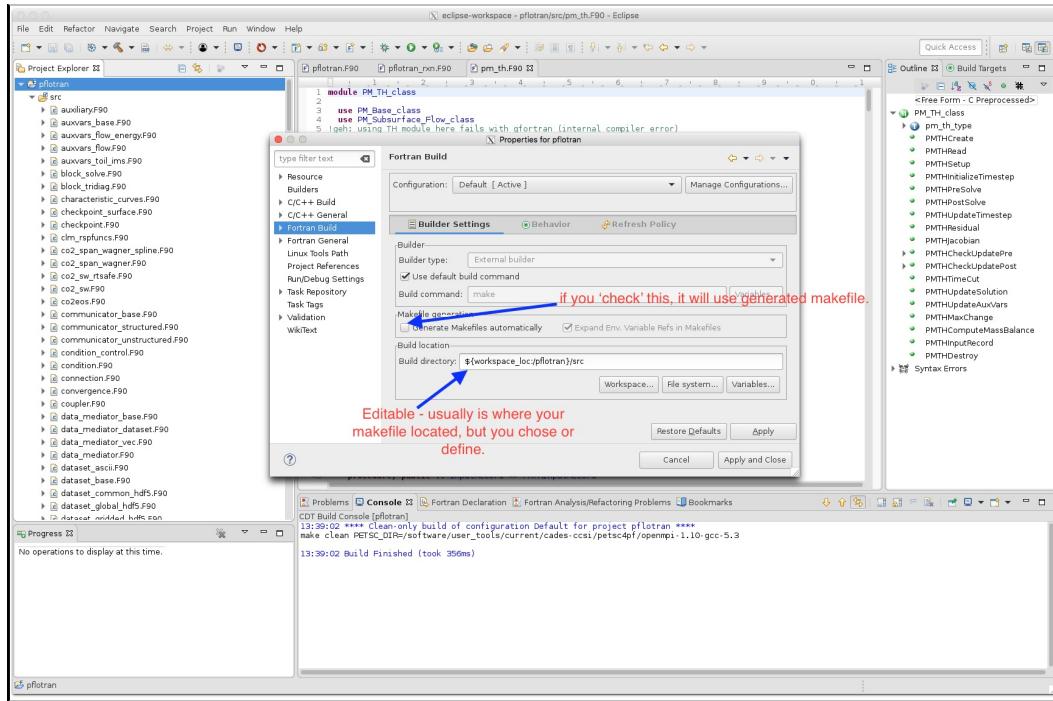
### 4.1 Project Properties Editing

Right-click on the Project in 'Project Explorer' window ...



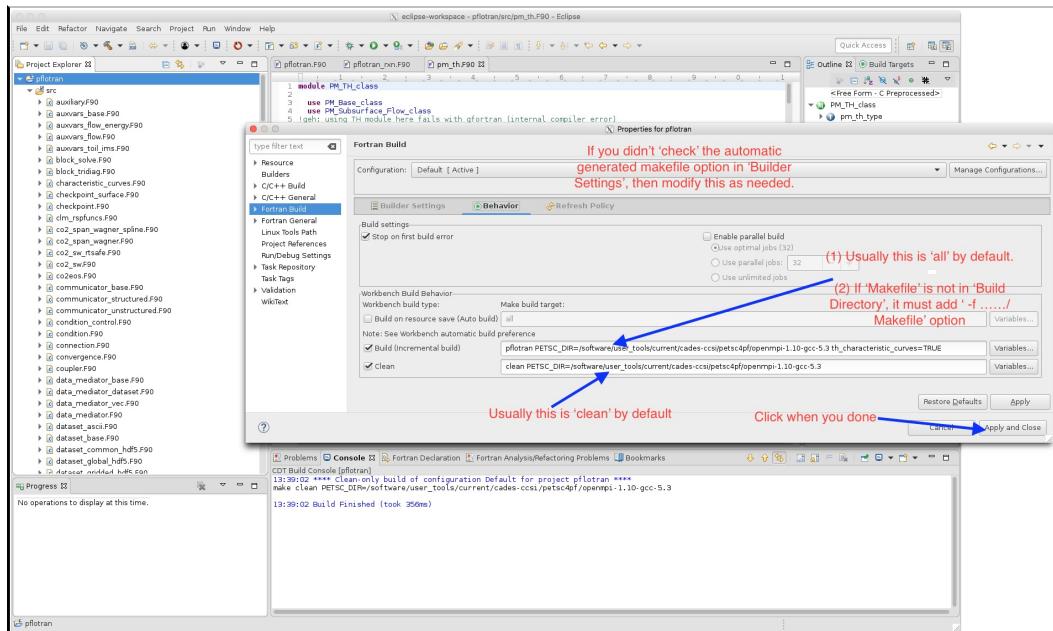
#### 4.2 Building settings

Editing as what you NEEDED ... e.g. Build Directory (NOTE: you may browse your file system)



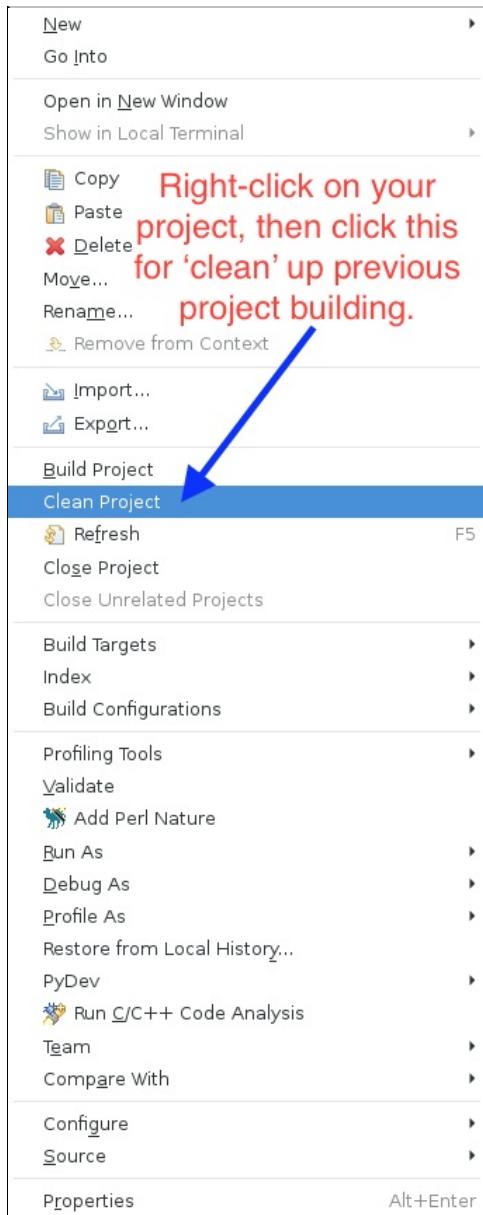
#### 4.3 Building behavior editing

AGAIN, as NEEDED.



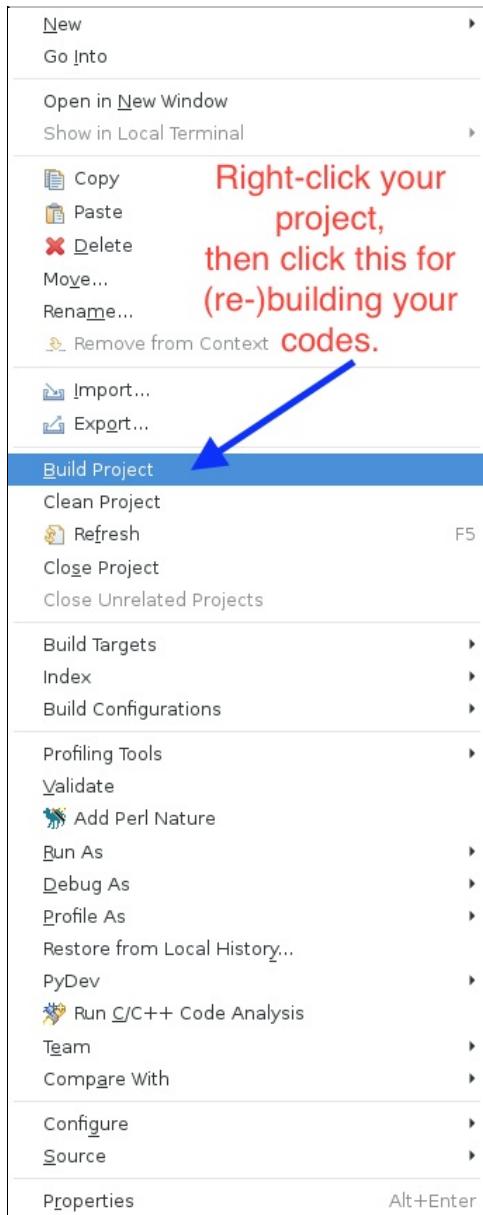
#### 4.4 Clean up (previous) build (OPTIONAL)

IF you would like to, clean previous build of project, by Right-click project, followed by clicking on 'Clean Project'

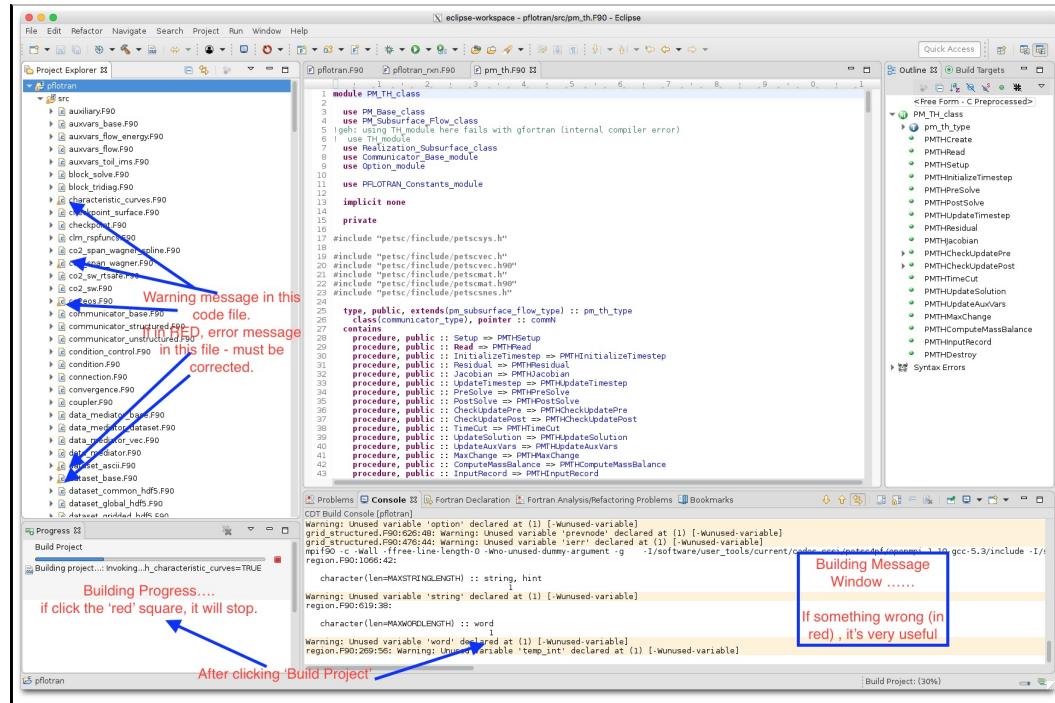


#### 4.5 Build your project

**Right-click on your project, then click on 'Build Project'**

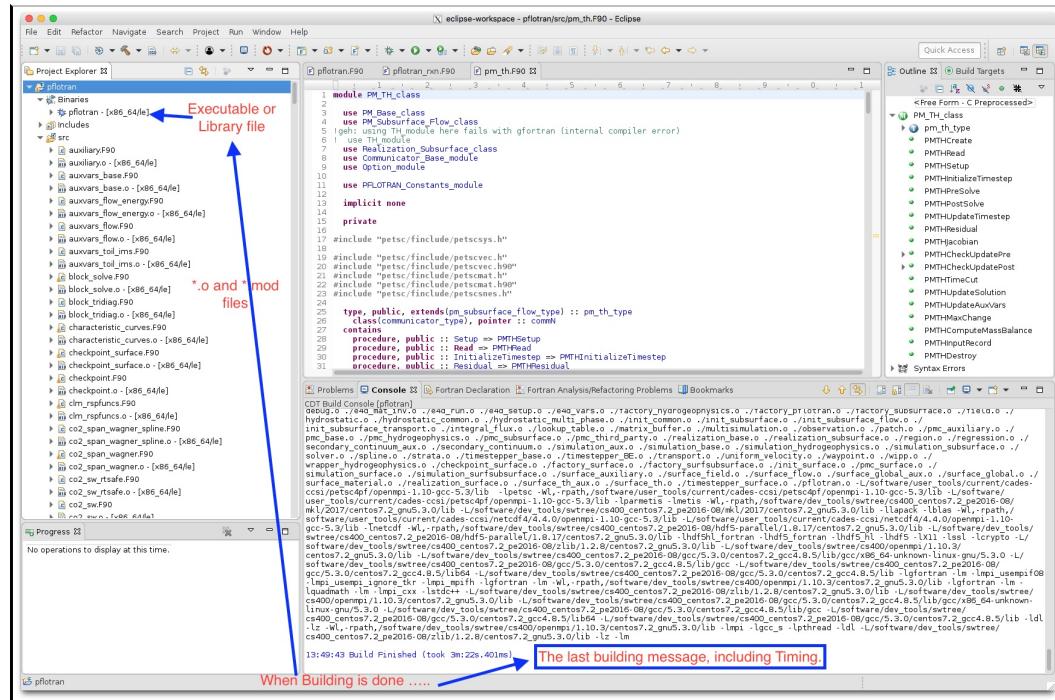


**Building progress .....**



### After Building Successfully.

IF NOT, clicking the 'error' code file and editing in the 'editor window'



## STEP 5. Run project

(NOTE: it's NOT allowed to directly run a program from the login node on CADES!)

**Run your executables in terminal, OR, include your built library (as USUALLY you do)**



Tutorial contributed by Fengming Yuan. **Note:** User-provided tutorials are not supported by CADES.

## Allinea Forge/DDT Client

### I. Allinea DDT Client

The Client software is free from: <https://www.allinea.com/products/forge/download#remote-client>.

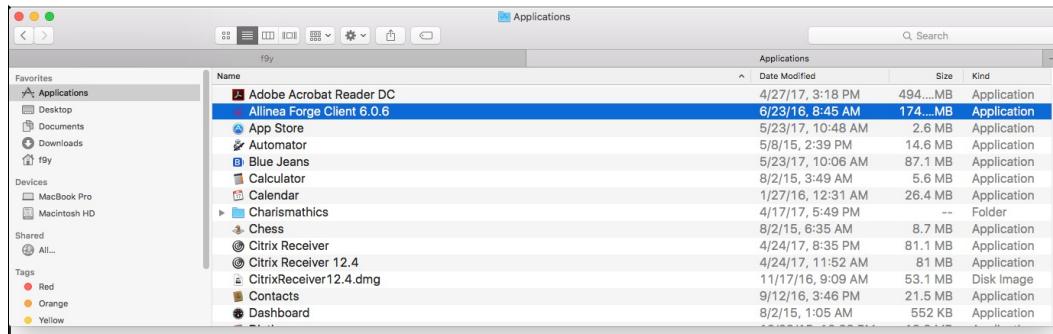
The version for matching with CADES' server-end is 6.0.6 (Currently).

### II. Configuration of Remote Server

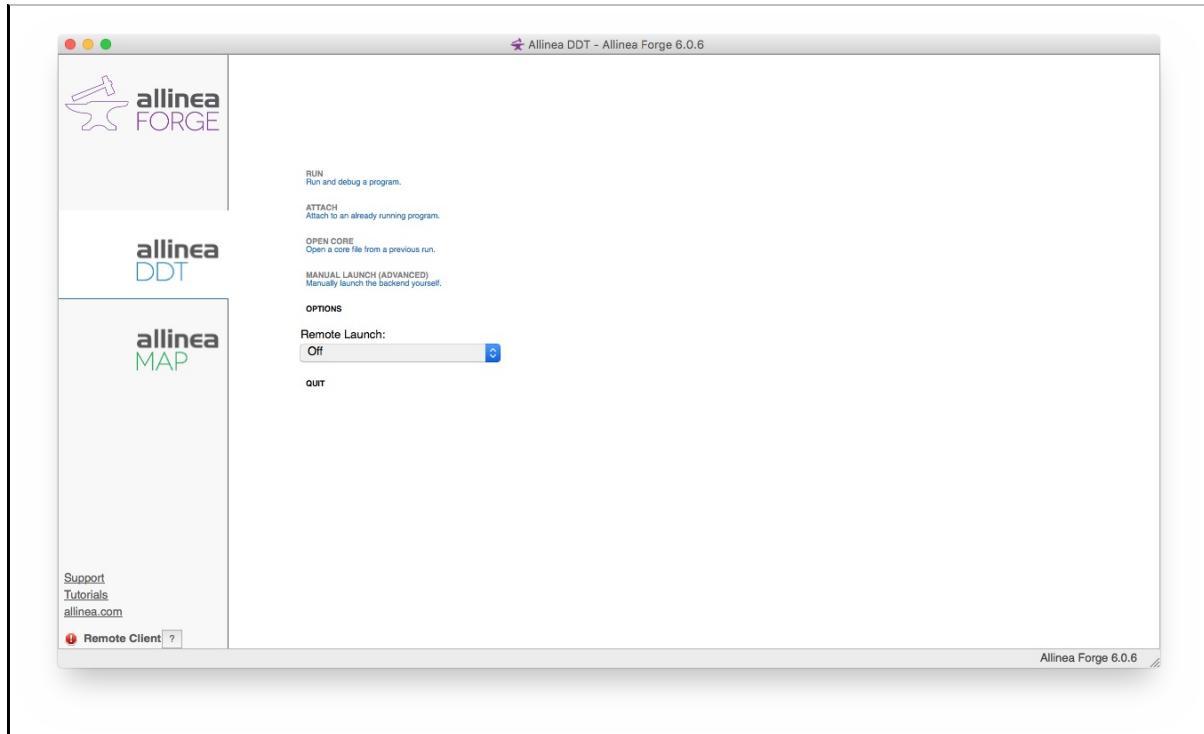
**Currently, CADES has 2 versions of Allinea DDT/Forge, the following is tested with v.6.0.6**

#### II-1. Start Allinea Forge Client locally

*Allinea Forge/DDT Client application installed .....*

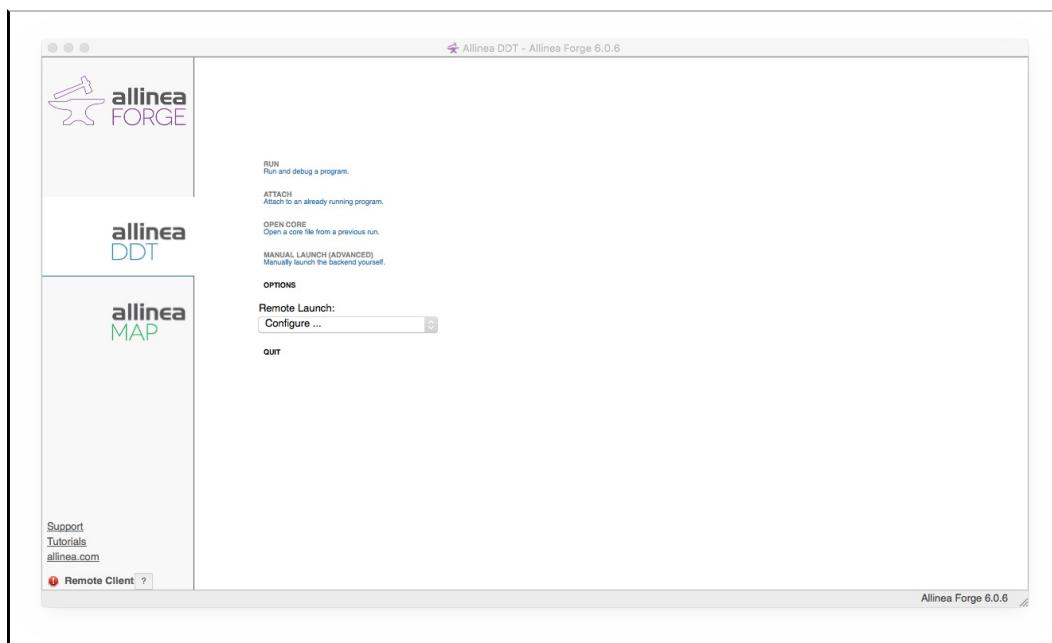


*On your screen .....*

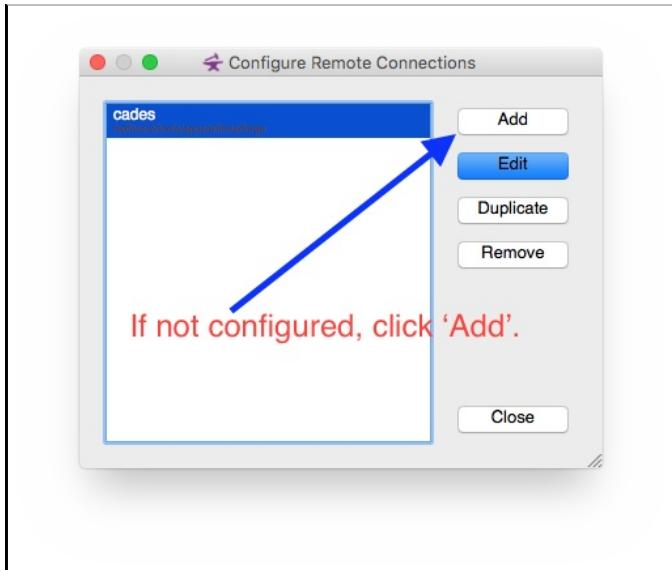


## **II-2. Configure Remote Launching**

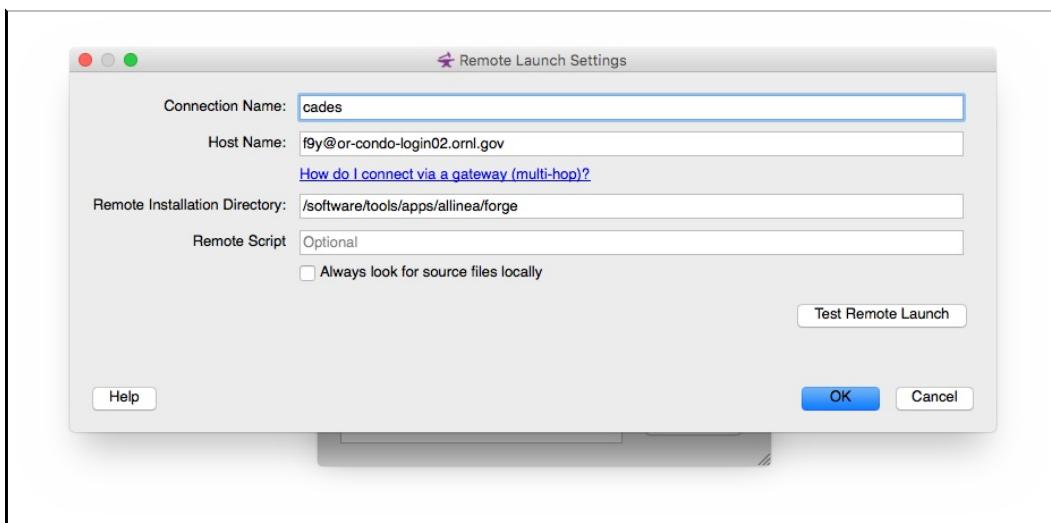
**Drop-down configue .....**



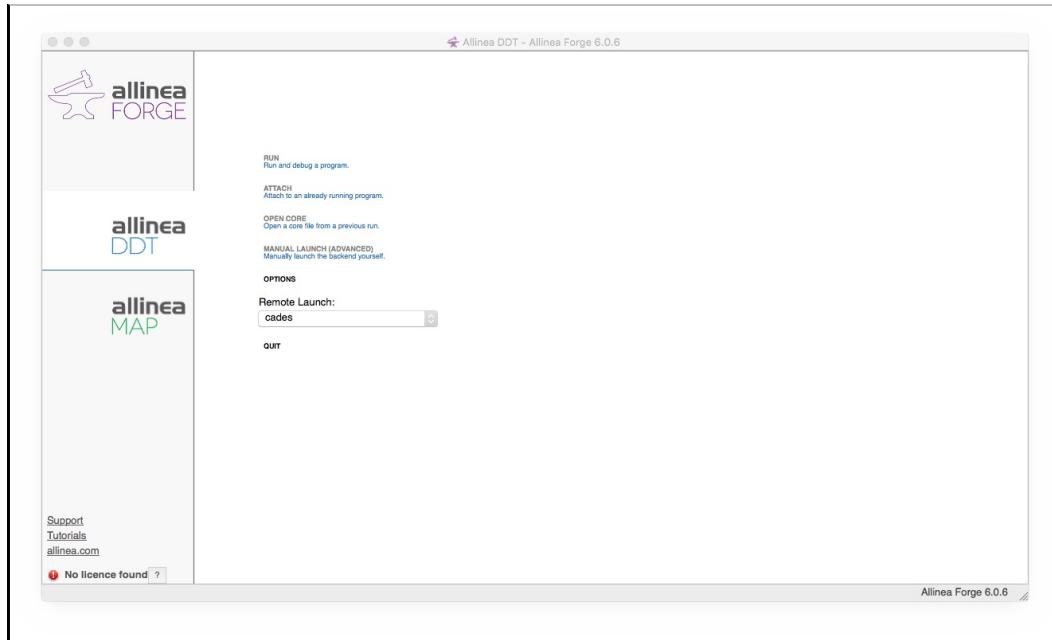
**Add or Modify .....**



**Remote Launch Settings.....**



**AND, .....**



## II-3. DDT template on CADES

You have to configure a DDT template file, to be used by Allinea Forge Client, as following:

e.g., here a file named as `ddt_template.qtf`, which MUST be put in your home directory.

```

#
# Name: Generic Torque
#
# WARNING: If you install a new version of Allinea Forge to the same
#           directory as this installation, then this file will be overwritten.
#           If you customize this script at all, please rename it.
#
# submit: qsub
# display: qstat
# job regexp: (.+)
# cancel: qdel JOB_ID_TAG
# show num_nodes: yes
#
# WALL_CLOCK_LIMIT_TAG: {type=text,label="Wall Clock Limit",default="00:30:00",mask="09:09:09"}
# QUEUE_TAG: {type=text,label="Queue",default=debug}
## Allinea Forge will generate a submission script from this by
## replacing these tags:
##      TAG NAME      |      DESCRIPTION      |      EXAMPLE
## -----
## PROGRAM_TAG      | target path and filename      | /users/ned/a.out
## PROGRAM_ARGUMENTS_TAG | arguments to target program | -myarg myval
## NUM_PROCS_TAG    | total number of processes   | 16
## NUM_NODES_TAG    | number of compute nodes     | 8
## PROCS_PER_NODE_TAG | processes per node        | 2
## NUM_THREADS_TAG  | OpenMP threads per proc   | 4
## DDT_DEBUGGER_ARGUMENTS_TAG | arguments to be passed to ddt-debugger
## MPIRUN_TAG       | name of mpirun executable   | mpirun
## AUTO_MPI_ARGUMENTS_TAG | mpirun arguments            | -np 4
## EXTRA_MPI_ARGUMENTS_TAG | extra mpirun arguments     | -x FAST=1

#!/bin/bash
#PBS -S /bin/bash
#PBS -m ae
#PBS -j oe
#PBS -M yuanf@ornl.gov
#PBS -N acme_debug

```

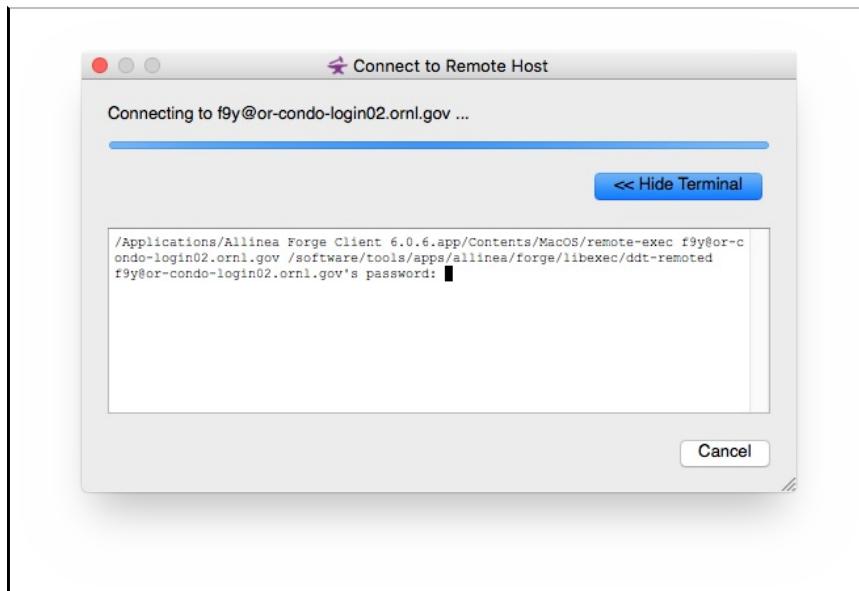
```
#####PBS -q QUEUE_TAG
#PBS -q batch
#PBS -l nodes=NUM_NODES_TAG:ppn=PROCS_PER_NODE_TAG
#PBS -l walltime=WALL_CLOCK_LIMIT_TAG
#PBS -W group_list=cades-ccsi
#PBS -A ccsi
#PBS -l qos=std
#PBS -l naccesspolicy=singlejob
export OMP_NUM_THREADS=1
#PBS -V
#PBS -o PROGRAM_TAG-ddt.output
#PBS -e PROGRAM_TAG-ddt.error
## The following line will use exactly the same default settings that
## Allinea Forge uses to launch without the queue.
module load env/cades-ccsi
module load vasp
AUTO_LAUNCH_TAG
## Replace the above for more complex situations - such as for passing unusual
## parameters to mpirun, like machine files or processes per node -- below is
## an example.
##
## if test "MPI_TAG" = "mpich 1 standard" ; then
##     MPIRUN_TAG -tv -np NUM_PROCS_TAG PROGRAM_TAG PROGRAM_ARGUMENTS_TAG
## else
##     if test DEBUG_STARTER_TAG -eq 1 ; then
##         DDT_CLIENT_TAG MPIRUN_TAG -np NUM_PROCS_TAG -machinefile $PBS_NODELIST PROGRAM_TAG PROGRAM_ARGUMENTS_TAG
##     else
##         MPIRUN_TAG -np NUM_PROCS_TAG -machinefile $PBS_NODELIST DDT_DEBUGGER_TAG PROGRAM_ARGUMENTS_TAG
##     fi
## fi
```

NOTE: (1)Please modify user information and others as NEEDED; (2) TIP: # or ## implies comments, EXCEPT for  
#PBS

### III. Step-by-Step Instruction on Debugging

#### III-1. Start 'Run and debug a program'

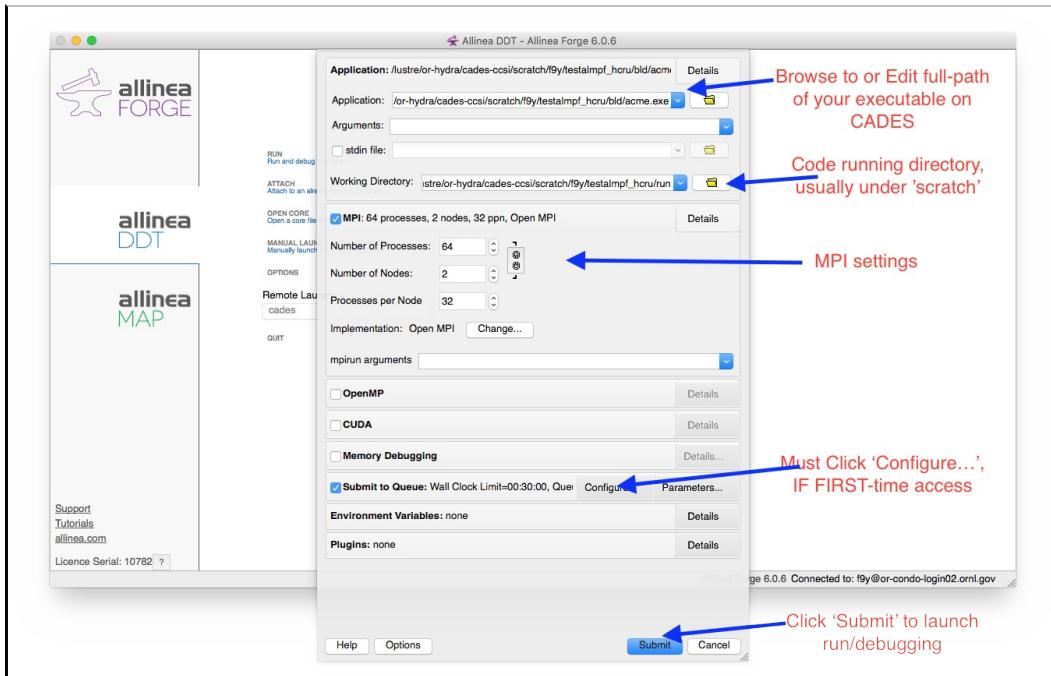
While 'Remote Launch: cades' is chosen, click "Run and debug a program" under "RUN" .....



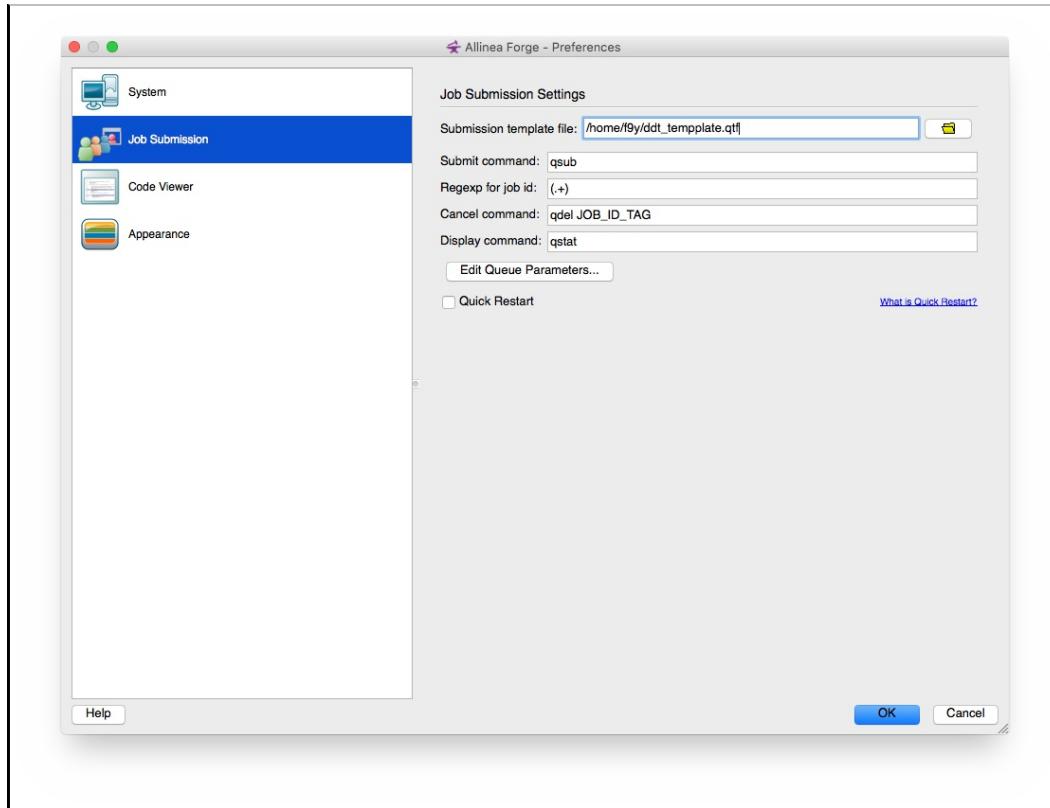
input your pwd, and ENTER .....

### III-2. Configure Allinea DDT job-submission to CADES

After successfully connecting to CADES, the following will pop up. THEN, you may add or edit a job-submission configuration



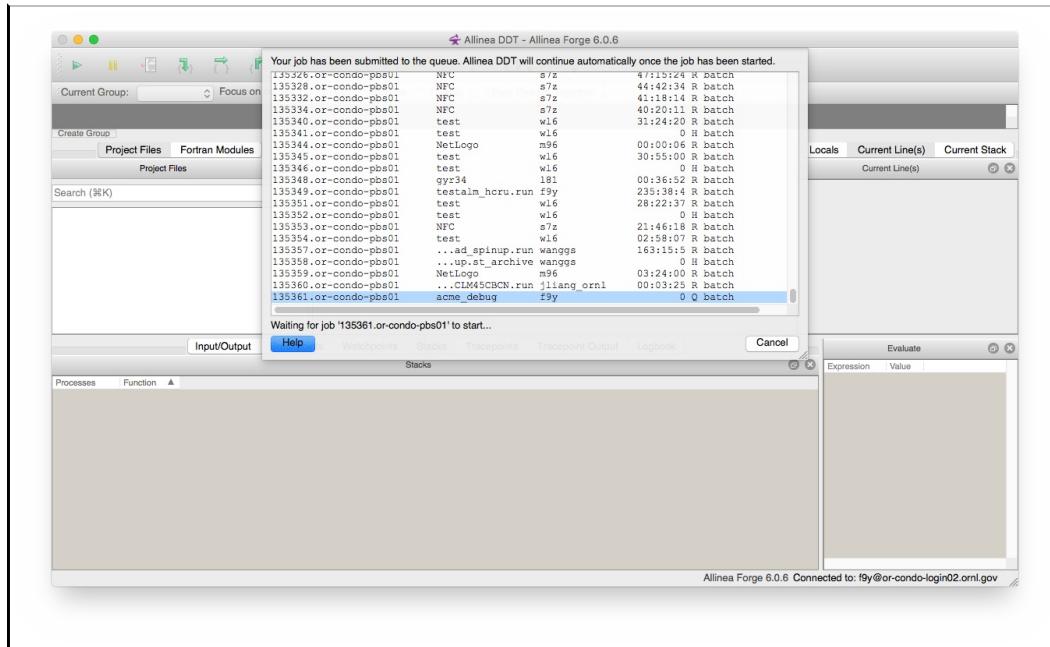
**IMPORTANT:** if this is your first time to access CADES for submitting a job to run & debug, you MUST Click 'Configure...' under 'Submit to Queue'. And then add the above template \*.qif file (in your home directory) and other settings, AS Following: .....



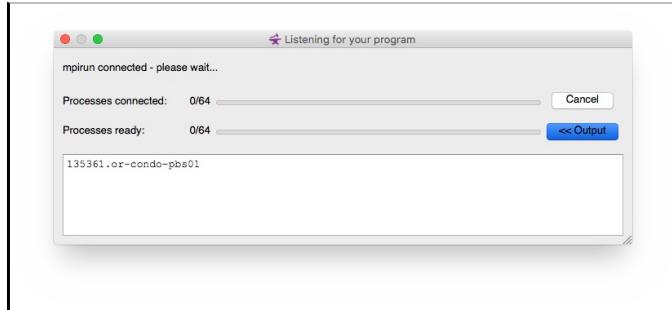
NOTE: This 'configuration' Window also can be used to configure other settings, as SHOWN. You may want to have some inspections by clicking each one\*

### III-3. Submit job and Run

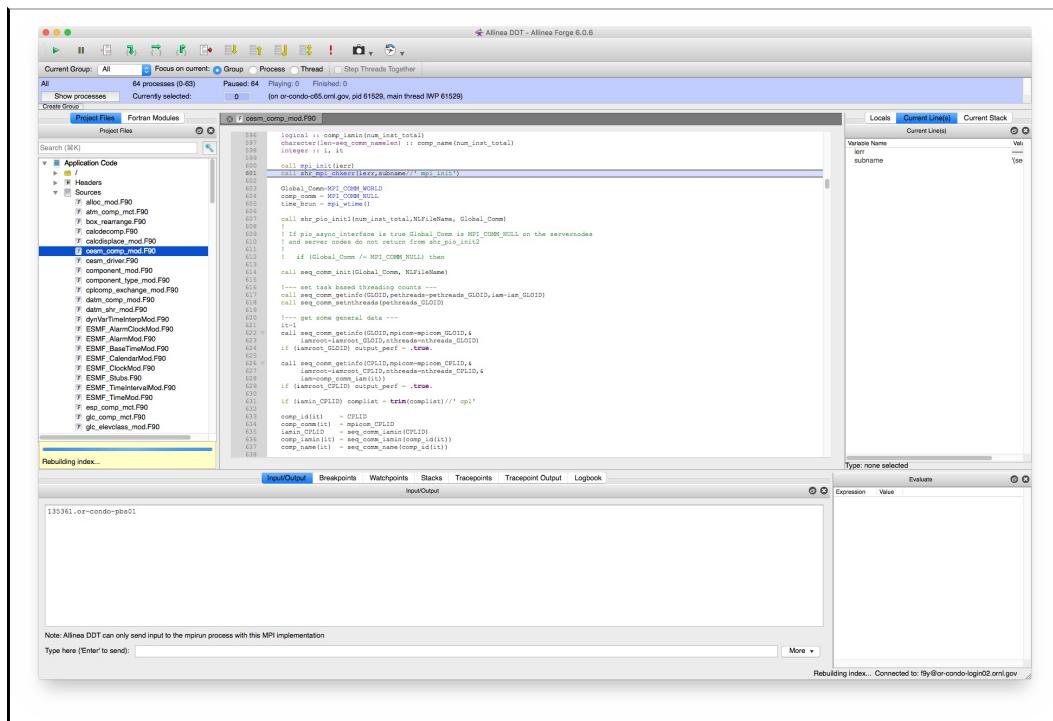
AFTER clicking 'SUBMIT' button .....



ONCE your job status is as "R", WINDOW will be switching and showing as .....(OTHERWISE, the above window would be hanging on)



**WHEN all requested Processes (e.g. here 64/64) connected and ready, the window will be changing into the run/debug views as FOLLOWING, and PAUSE at the main program entry point (usually).....**



### III-4. Debug The Program

#### PAUSE at Breakpoint(s)

Allinea DDT - Allinea Forge 6.0.6

Current Group: All Focus on current: Group Process Thread Step Threads Together

Show processes 51 52 53 54 55 56 57 58 59 60 61 62 63

\_locals Current Line(s) Current Stack

Variable Name Value  
c /> soilcount  
larea /> larea ([1])  
toparea\_dm\_loc /> toparea\_dm\_loc ([1])  
vlgcell\_sum /> vlgcell\_sum ([1])

```

1559         gout = 0
1560         do c = boundsdblegp, boundsendo
1561             g = c*gridsize(c)
1562             larea = 0.0
1563
1564             ! note that filtersoilcell includes 'lstart' and 'lstop'
1565             ! (TODO: checking colltype and lunktype - appears not match with each other, and colltype IS messy)
1566             if (lstart <= lstop .and. lstart >= lstop-1 .and. lstart >= lstop-2 .and. lstop >= lstop-1) then
1567                 gout = gout + 1
1568                 if (active(c) .and. c*gridsize(c)>=r8) then
1569                     gout = gout + 1
1570                     ! actually is the active soil column count
1571                     colcount = colcount + 1
1572                     larea = larea + g
1573
1574                     dsoil_cilm_loc(meillecount) = dsoil_cilm_loc(meillecount) + larea(g)
1575                     rsoil_cilm_loc(meillecount) = rsoil_cilm_loc(meillecount) + larea(g)
1576                     vlgcell_sum(meillecount) = vlgcell_sum(meillecount) + larea(g)
1577
1578                     ! top face areas, scaled by active column weight and land fraction
1579                     toparea_cilm_loc(meillecount) = vlgcell_sum(c)*ldomainfrac(g)*larea(g)*lstart*r8/lm2
1580
1581             ! after knowing 'ltoparea', we may get a pseudo 'dmx' and 'dmy' so that PF will not crash
1582             ! (note: PF needs these information otherwise throw-out error message, even with 'vertical_only' option)
1583             if (ldomainfrac(g) < 0) then ! lwgps: ldomainfrac missing in ACM1
1584                 ldomainfrac(g) = 0.0
1585             end if
1586             late = lstart+1
1587             late = late+1
1588             dsoil_cilm_loc(meillecount) = abelarea(l1+late(4)-late(2)-late(3))/2.0_r8
1589             ! note: since in 'column_wise' mode, the columns are in 1D array by x-axis,
1590             ! only need to scale 'r8' by column area fraction
1591             dsoil_cilm_loc(meillecount) = abelarea(l1+late(2)-late(3))/late(4)/2.0_r8
1592
1593             else
1594                 dsoil_cilm_loc(meillecount) = late(g)/(r8*c) * ldomainfrac(g)
1595                 dsoil_cilm_loc(meillecount) = late(g)/(r8*c) * ldomainfrac(g)
1596             endif
1597
1598         endifif
1599     endif

```

Type: none selected

Input/Output Breakpoints Watchpoints Stacks Tracepoints Tracepoint Output Logbook

Expression Value

proc = 31 begpr= 64413 endlpr= 64968 numnl= 556
 rtm decomp info proc = 31 begpr= 62927 endlpr= 64427 numnr= 3501
 proc = 1 begpr= 3017 endlpr= 5149 numnl= 2133
 rtm decomp info proc = 62 begpr= 1053 endlpr= 2977 numnr= 1925
 proc = 62 begpr= 83765 endlpr= 84031 numnl= 267
 rtm decomp info proc = 63 begpr= 23570 endlpr= 259200 numnr= 2458
 proc = 63 begpr= 174465 endlpr= 174680 numnl= 2225
 Opened existing file /lustre/or-hydras/cades-cscl/proj-shared/project\_acme/ACME\_inputdata/ind/clnd/rmdata/rdir0\_0.5x0.5\_smyr2000\_sipmxv1\_c120717.nc 786432
 Opened existing file /lustre/or-hydras/cades-cscl/proj-shared/project\_acme/ACME\_inputdata/rft/rmt/initdata/rmt1.II850CROCLM45HGC.0241-01-01.R05\_smyr1850\_c130515.nc 786432

Note: Allinea DDT can only send input to the mpnrun process with this MPI implementation  
Type here ('Enter' to send): More →

3 processes playing. Connected to: fly@or-condo-login02.ont.gov

CHECK variables: current line or locals for current subroutine, OR do what you want .....

Allinea DDT - Allinea Forge 6.0.6

Current Group: All Focus on current: Group Process Thread Step Threads Together

Show processes 51 52 53 54 55 56 57 58 59 60 61 62 63

\_locals Current Line(s) Current Stack

Variable Name Value  
c /> 31787  
larea /> larea ([1]) = 5.5262356876784097e-316  
toparea\_cilm\_loc /> toparea\_cilm\_loc ([1]) = 0.0, [2] = 0.0, [3] = 0.0, [4] = 0.0, [5] = 0.0, [6] = 0.0, [7] = 0.0, [8] = 0.0, [9] = 0.0, [10] = 0.0, [11] = 0.0, [12] = 0.0, [13] = 0.0, [14] = 0.0, [15] = 0.0, [16] = 0.0, [17] = 0.0, [18] = 0.0, [19] = 0.0, [20] = 0.0, [21] = 0.0, [22] = 0.0, [23] = 0.0, [24] = 0.0, [25] = 0.0, [26] = 0.0, [27] = 0.0, [28] = 0.0, [29] = 0.0, [30] = 0.0

```

1559         gout = 0
1560         do c = boundsdblegp, boundsendo
1561             g = c*gridsize(c)
1562             larea = 0.0
1563
1564             ! note that filtersoilcell includes 'lstart' and 'lstop'
1565             ! (TODO: checking colltype and lunktype - appears not match with each other, and colltype IS messy)
1566             if (lstart <= lstop .and. lstart >= lstop-1 .and. lstart >= lstop-2 .and. lstop >= lstop-1) then
1567                 gout = gout + 1
1568                 if (active(c) .and. c*gridsize(c)>=r8) then
1569                     gout = gout + 1
1570                     ! actually is the active soil column count
1571                     colcount = colcount + 1
1572                     larea = larea + g
1573
1574                     dsoil_cilm_loc(meillecount) = dsoil_cilm_loc(meillecount) + larea(g)
1575                     rsoil_cilm_loc(meillecount) = rsoil_cilm_loc(meillecount) + larea(g)
1576                     vlgcell_sum(meillecount) = vlgcell_sum(meillecount) + larea(g)
1577
1578                     ! top face areas, scaled by active column weight and land fraction
1579                     toparea_cilm_loc(meillecount) = vlgcell_sum(c)*ldomainfrac(g)*larea(g)*lstart*r8/lm2
1580
1581             ! after knowing 'ltoparea', we may get a pseudo 'dmx' and 'dmy' so that PF will not crash
1582             ! (note: PF needs these information otherwise throw-out error message, even with 'vertical_only' option)
1583             if (ldomainfrac(g) < 0) then ! lwgps: ldomainfrac missing in ACM1
1584                 ldomainfrac(g) = 0.0
1585             end if
1586             late = lstart+1
1587             late = late+1
1588             dsoil_cilm_loc(meillecount) = abelarea(l1+late(4)-late(2)-late(3))/2.0_r8
1589             ! note: since in 'column_wise' mode, the columns are in 1D array by x-axis,
1590             ! only need to scale 'r8' by column area fraction
1591             dsoil_cilm_loc(meillecount) = abelarea(l1+late(2)-late(3))/late(4)/2.0_r8
1592
1593             else
1594                 dsoil_cilm_loc(meillecount) = late(g)/(r8*c) * ldomainfrac(g)
1595                 dsoil_cilm_loc(meillecount) = late(g)/(r8*c) * ldomainfrac(g)
1596             endif
1597
1598         endifif
1599     endif

```

Type: real(kind=8), ALLOCATABLE (14235)

Input/Output Breakpoints Watchpoints Stacks Tracepoints Tracepoint Output Logbook

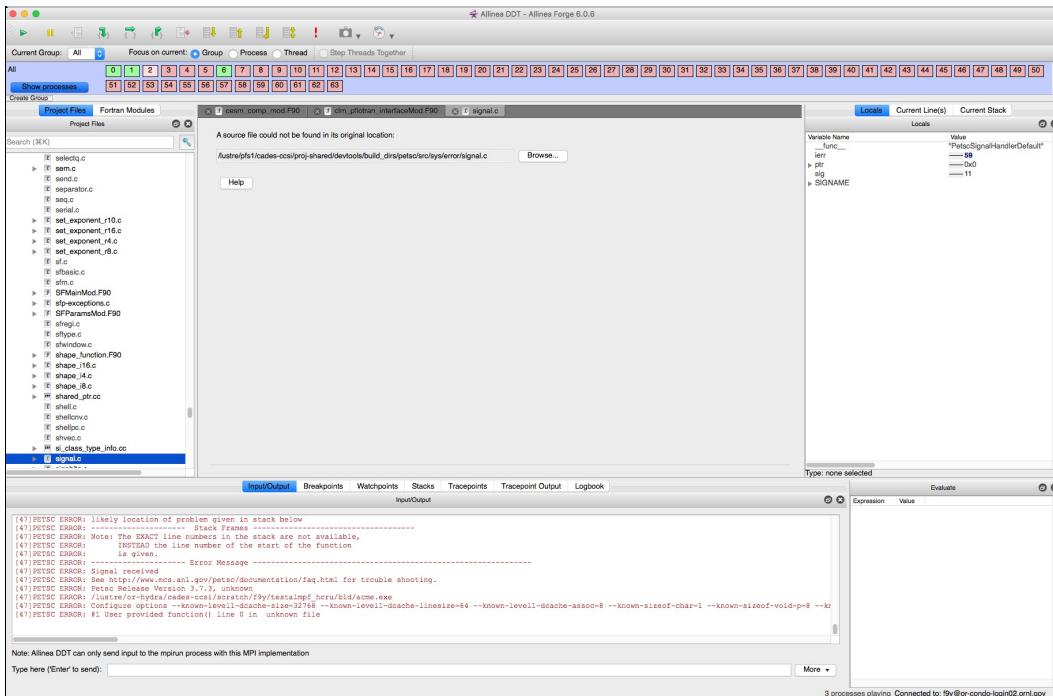
Expression Value

proc = 31 begpr= 64413 endlpr= 64968 numnl= 556
 rtm decomp info proc = 31 begpr= 62927 endlpr= 66427 numnr= 3501
 proc = 1 begpr= 3017 endlpr= 5149 numnl= 2133
 rtm decomp info proc = 62 begpr= 1053 endlpr= 2977 numnr= 1925
 proc = 62 begpr= 83765 endlpr= 84031 numnl= 267
 rtm decomp info proc = 63 begpr= 23570 endlpr= 259200 numnr= 2458
 proc = 63 begpr= 174465 endlpr= 174680 numnl= 2225
 Opened existing file /lustre/or-hydras/cades-cscl/proj-shared/project\_acme/ACME\_inputdata/ind/clnd/rmdata/rdir0\_0.5x0.5\_smyr2000\_sipmxv1\_c120717.nc 786432
 Opened existing file /lustre/or-hydras/cades-cscl/proj-shared/project\_acme/ACME\_inputdata/rft/rmt/initdata/rmt1.II850CROCLM45HGC.0241-01-01.R05\_smyr1850\_c130515.nc 786432

Note: Allinea DDT can only send input to the mpnrun process with this MPI implementation  
Type here ('Enter' to send): More →

3 processes playing. Connected to: fly@or-condo-login02.ont.gov

HIT a BUG, .....(NOT SO FUN?)



## Prerequisites

To properly utilize your CADES SHPC Condo allocation, you will need a couple of utilities loaded on your local workstation. These utilities are free and widely used for this type of application.

- Required: **SSH client**
- Recommended: **Bash terminal**

**Note:** CADES does not provide support for getting these utilities up and running on your workstation.

## Windows

Windows does not have a native SSH client or a native Bash terminal. Two recommended solutions are linked below.

**Required:** [PuTTY](#) – Windows users will need some form of SSH tool, and the CADES team recommends PuTTY for this purpose. The [PuTTY MSI installer](#) also includes PuTTYGen, which allows you to convert your private `.pem` SSH key to a `.ppk` PuTTY key file.

See the tutorial below on how to use the PuTTY client to connect to remote host via SSH.

| How to: [Access Your VM Instance Using PuTTY \(Windows\)](#)

**Recommended:** [Cygwin](#) – If you wish to have Bash-style functions on your Windows machine, then you should consider installing Cygwin, which ports the Portable Operating System Interface (POSIX) system calls and environment to Windows.

**Alternative:** The Windows Subsystem for Linux (WSL) may provide a more modern and customizable environment  
<https://msdn.microsoft.com/en-us/commandline/wsl/faq>

## macOS and Linux

Both macOS and Linux distributions include a Bash terminal and an SSH client by default. No additional software should be required to access SHPC Condos.

## Getting Started on CADES SHPC Condos

The CADES Scalable HPC (SHPC) Condos consist of two HPC clusters: one in the ORNL Moderate protection zone (CADES Mod) and one in the ORNL Open protection zone (CADES Open). The protection zones contain and control both the software base and the data produced on those systems.

This section outlines the most basic procedures for procuring and using a CADES SHPC Condo allocation.

### Step 1: Request Your SHPC Condo Allocation

You can request a CADES SHPC Condo allocation by clicking on the appropriate XCAMS registration [link for your group](#). If you do not see your group listed, please contact the [CADES team](#).

The XCAMS registration process will ask you to acknowledge the XCAMS User Agreement and register your UCAMS with an LDAP group or create a new XCAMS user ID and the register it with an LDAP group. Use the steps below to enter your request.

1. Navigate to the appropriate [XCAMS registration link for your group](#).
2. Enter your email address (your ORNL address if available) and click `Continue`.
3. Review the XCAMS user agreement, and select `Agree`.
4. Enter your UCAMS ID (or a new XCAMS user name).
5. Enter your UCAMS password (or a new XCAMS password).
6. Click `Submit` to complete the XCAMS request.

When your resources are ready for use, an activation notice will be dispatched to the email address entered above. This process can take up to 24 hours to complete.

### Step 2: Access Your SHPC Condo Allocation

Once your request for resources has been approved, you can access the SHPC Condo login nodes using SSH. Open and Moderate protection zones each have their own login node. Choose the login node for your protection zone.

#### Open Protection Zone

1. Open a Bash terminal (or PuTTY for Windows users).
2. Execute `ssh xcams@or-condo-login.ornl.gov`.
  - Replace "xcams" with the username you registered above.
3. When prompted, enter your XCAMS password.

#### Moderate Protection Zone

1. Open a Bash terminal (or PuTTY for Windows users).
2. Execute `ssh ucams@mod-condo-login.ornl.gov`.
  - Replace "ucams" with your ORNL UCAMS ID.
3. When prompted, enter your UCAMS password.

### Step 3: Execute a Job on Your SHPC Allocation

Now that you have access to your allocation through the login node, it is time to do some work. The tutorial linked below, intended for users who are new to the CADES SHPC Condo environment, outlines the basic steps to setting up and executing a job on the SHPC Condo compute nodes.

| How to: [Execute a Job on Your SHPC Condo Allocation](#)

**Note:** Do not execute jobs on the login nodes; only use the login nodes to access your compute nodes. Processor-intensive, memory-intensive, or otherwise disruptive processes running on login nodes will be killed without warning.

## Getting Help

If at any point you feel stuck and need some help figuring out your next move, please contact the CADES support team at [cades-help@ornl.gov](mailto:cades-help@ornl.gov) or join our Slack channel at <https://cades.slack.com/signup>.

# SHPC Condo Configuration

## Hardware Configuration

### Node Information

- **Make:** Cray
- **Model:** CS400

### RAM Information

- **Speed:** DDR4 2133
- **Error correction:** Registered ECC
- **Capacity:** 128–256 GB per node (GPU nodes and high memory nodes have 256 GB of RAM)

### CPU Information

- **Make:** Intel
- **Model:** Xeon E5-2698 v3
- **Speed:** 2.30 GHz base clock, 3.60 GHz Turbo Boost clock
- **Capacity:** 2 CPUs per node
- **CPU layout:** [Click to see image.](#)

### GPU Information

- **Make:** NVIDIA
- **Model:** Tesla K80 (2 GK210 GPUs on each K80)
- **Speed:** 560 MHz base clock
- **VRAM:** 24 GB of GDDR5
- **Error correction:** Registered ECC
- **Capacity:** 2 Tesla K80s per node, 2 GK210 GPUs per K80 (4 total GK210 GPUs per node)

## Storage Configuration

### Lustre

Lustre allocations are available in both Open and Moderate protection zones and are located in the paths given below.

- Open protection zone: `/lustre/or-hydra/LDAP-Group/XCAMS`
- Moderate protection zone: `/lustre/hydra/LDAP-Group/UCAMS`

Replace "LDAP-Group" with your LDAP group name. Replace "XCAMS" or "UCAMS" with your XCAMS or UCAMS ID. Click [here](#) to find your LDAP group.

## Software Stack

### Host Operating system

CentOS 7.x

## Available Modules

Located in `/software/tools/modules` :

| Module Name       | Version | Full Path                                        |
|-------------------|---------|--------------------------------------------------|
| Gromacs           | 5.1.1   | /software/tools/modules/apps/gromacs/gcc5/5.1.1  |
| NAMD2             | 2.11.0  | /software/tools/modules/apps/namd2/binary/2.11.0 |
| P3DFFT            | 2.7.5   | /software/tools/modules/apps/p3dfft/intel/2.7.5  |
| Intel Compiler    | 2016.1  | /software/tools/modules/compilers/intel/2016.1   |
| Intel Compiler    | 2017.0  | /software/tools/modules/compilers/intel/2017.0   |
| PGI Compiler      | 15.7    | /software/tools/modules/compilers/pgi/15.7       |
| CADES-ALICE       |         | /software/tools/modules/env/cades-alice          |
| CADES-ARM-USGS    |         | /software/tools/modules/env/cades-arm-usgs       |
| CADES-bellerophon |         | /software/tools/modules/env/cades-bellerophon    |
| CADES-BSD         |         | /software/tools/modules/env/cades-bsd            |
| CADES-CCSI        |         | /software/tools/modules/env/cades-ccsi           |
| CADES-CEES        |         | /software/tools/modules/env/cades-cees           |
| CADES-CNMS        |         | /software/tools/modules/env/cades-cnms           |
| CADES-datarep     |         | /software/tools/modules/env/cades-datarep        |
| CADES-EDDE        |         | /software/tools/modules/env/cades-edde           |
| CADES-ESGF        |         | /software/tools/modules/env/cades-esgf           |
| CADES-HACC        |         | /software/tools/modules/env/cades-hacc           |
| CADES-nucphys     |         | /software/tools/modules/env/cades-nucphys        |
| CADES-Ops         |         | /software/tools/modules/env/cades-ops            |
| CADES-SW          |         | /software/tools/modules/env/cades-sw             |
| CADES-Virtues     |         | /software/tools/modules/env/cades-virtues        |
| CNMS              |         | /software/tools/modules/env/cnms                 |
| MVAPICH           | 2.2rc1  | /software/tools/modules/mpi/mvapich-gnu/2.2rc1   |
| OpenMPI GCC       | 5.3.0   | /software/tools/modules/mpi/openmpi/gcc/5.3.0    |
| OpenMPI Intel     | 1.10.2  | /software/tools/modules/mpi/openmpi/intel/1.10.2 |
| Allinea           | 6.0.6   | /software/tools/modules/utils/allinea/6.0.6      |
| ATLAS             | 3.11.34 | /software/tools/modules/utils/atlas/3.11.34      |
| CUDA              | 7.5.17  | /software/tools/modules/utils/cuda/7.5.17        |
| GNUploat GCC      | 5.0.3   | /software/tools/modules/utils/gnuplot/gcc/5.0.3  |
| MKL               | 2016.1  | /software/tools/modules/utils/intel/mkl/2016.1   |

Located in `/software/dev_tools/swtree/cs400/modulefiles` :

| <b>Module Name</b>           | <b>Version</b>    | <b>Full Path</b>                                                           |
|------------------------------|-------------------|----------------------------------------------------------------------------|
| Amber                        | 14 (default)      | /software/dev_tools/swtree/cs400/modulefiles/amber/14(default)             |
| Amber                        | 16                | /software/dev_tools/swtree/cs400/modulefiles/amber/16                      |
| Anaconda2                    | 4.3.1             | /software/dev_tools/swtree/cs400/modulefiles/anaconda2/4.3.1               |
| ATLAS                        | 3.10.3            | /software/dev_tools/swtree/cs400/modulefiles/ATLAS/3.10.3                  |
| BOOST                        | 1.61.0 (default)  | /software/dev_tools/swtree/cs400/modulefiles/boost/1.61.0(default)         |
| BOOST                        | 1.64              | /software/dev_tools/swtree/cs400/modulefiles/boost/1.64                    |
| bzip2                        | 1.0.6 (default)   | /software/dev_tools/swtree/cs400/modulefiles/bzip2/1.0.6(default)          |
| Caffe                        | 1.0-conda         | /software/dev_tools/swtree/cs400/modulefiles/caffe/1.0-conda               |
| CMake                        | 3.5.2             | /software/dev_tools/swtree/cs400/modulefiles/cmake/3.5.2                   |
| CMake                        | 3.6.1             | /software/dev_tools/swtree/cs400/modulefiles/cmake/3.6.1                   |
| CUDA                         | 6.5               | /software/dev_tools/swtree/cs400/modulefiles/cuda/6.5                      |
| CUDA                         | 7.5 (default)     | /software/dev_tools/swtree/cs400/modulefiles/cuda/7.5(default)             |
| CUDA                         | 8.0               | /software/dev_tools/swtree/cs400/modulefiles/cuda/8.0                      |
| Exodus                       | 5.14              | /software/dev_tools/swtree/cs400/modulefiles/exodus/5.14(default)          |
| FFTW                         | 3.3.5 (default)   | /software/dev_tools/swtree/cs400/modulefiles/fftw/3.3.5(default)           |
| GCC                          | 4.9.4             | /software/dev_tools/swtree/cs400/modulefiles/gcc/4.9.4                     |
| GCC                          | 5.3.0 (default)   | /software/dev_tools/swtree/cs400/modulefiles/gcc/5.3.0(default)            |
| GCC                          | 6.2.0             | /software/dev_tools/swtree/cs400/modulefiles/gcc/6.2.0                     |
| Git                          | 2.11.0 (default)  | /software/dev_tools/swtree/cs400/modulefiles/git/2.11.0(default)           |
| Gromacs                      | 2016.3            | /software/dev_tools/swtree/cs400/modulefiles/gromacs/2016.3                |
| Gromacs                      | 2016.3-AVX2       | /software/dev_tools/swtree/cs400/modulefiles/gromacs/2016.3-AVX2           |
| Gromacs                      | 5.1.2             | /software/dev_tools/swtree/cs400/modulefiles/gromacs/5.1.2                 |
| GTK+                         | 2.24.31 (default) | /software/dev_tools/swtree/cs400/modulefiles/gtk+/2.24.31(default)         |
| Parallel HDF5                | 1.8.17 (default)  | /software/dev_tools/swtree/cs400/modulefiles/hdf5-parallel/1.8.17(default) |
| HDF5                         | 1.8.17 (default)  | /software/dev_tools/swtree/cs400/modulefiles/hdf5/1.8.17(default)          |
| HYPRE                        | 2.11.1 (default)  | /software/dev_tools/swtree/cs400/modulefiles/hypre/2.11.1(default)         |
| HYPRE                        | 2.6.0b            | /software/dev_tools/swtree/cs400/modulefiles/hypre/2.6.0b                  |
| ILAMB                        | 2.1 (default)     | /software/dev_tools/swtree/cs400/modulefiles/ilamb/2.1(default)            |
| Intel Compiler (Fortran & C) | 16.0.1 (default)  | /software/dev_tools/swtree/cs400/modulefiles/intel/16.0.1(default)         |
| Intel Compiler (Fortran & C) | 17.0.0            | /software/dev_tools/swtree/cs400/modulefiles/intel/17.0.0                  |
| Java                         | 1.8.0_131         | /software/dev_tools/swtree/cs400/modulefiles/java/1.8.0_131                |

|                      |                   |                                                                                   |
|----------------------|-------------------|-----------------------------------------------------------------------------------|
| Keras                | 1.8.0_131         | /software/dev_tools/swtree/cs400/modulefiles/keras/2.0.2-conda                    |
| LAMMPS               | 31Mar17           | /software/dev_tools/swtree/cs400/modulefiles/lammps/31Mar17                       |
| MAGMA                | 1.7.0 (default)   | /software/dev_tools/swtree/cs400/modulefiles/magma/1.7.0(default)                 |
| METIS                | 5.1.0 (default)   | /software/dev_tools/swtree/cs400/modulefiles/metis/5.1.0(default)                 |
| MKL                  | 2017 (default)    | /software/dev_tools/swtree/cs400/modulefiles/mkl/2017(default)                    |
| MPICH                | 3.2 (default)     | /software/dev_tools/swtree/cs400/modulefiles/mpich/3.2(default)                   |
| NAG                  | 6.0 (default)     | /software/dev_tools/swtree/cs400/modulefiles/nag/6.0(default)                     |
| NCL                  | 6.3.0 (default)   | /software/dev_tools/swtree/cs400/modulefiles/ncl/6.3.0(default)                   |
| NCO                  | 4.6.4             | /software/dev_tools/swtree/cs400/modulefiles/nco/4.6.4                            |
| NCO                  | 4.6.6             | /software/dev_tools/swtree/cs400/modulefiles/nco/4.6.6(default)                   |
| NCView               | 2.1.7 (default)   | /software/dev_tools/swtree/cs400/modulefiles/ncview/2.1.7(default)                |
| NCView               | 2.1.7 (default)   | /software/dev_tools/swtree/cs400/modulefiles/ncview/2.1.7(default)                |
| NetCDF HDF5 Parallel | 4.3.3.1 (default) | /software/dev_tools/swtree/cs400/modulefiles/netcdf-hdf5parallel/4.3.3.1(default) |
| NetCDF               | 4.3.3.1 (default) | /software/dev_tools/swtree/cs400/modulefiles/netcdf/4.3.3.1(default)              |
| NWChem               | 6.6 (default)     | /software/dev_tools/swtree/cs400/modulefiles/nwchem/6.6(default)                  |
| OpenBLAS             | 0.2.19            | /software/dev_tools/swtree/cs400/modulefiles/openBLAS/0.2.19                      |
| OpenMPI              | 1.10.3 (default)  | /software/dev_tools/swtree/cs400/modulefiles/openmpi/1.10.3(default)              |
| OpenMPI              | 2.0.0             | /software/dev_tools/swtree/cs400/modulefiles/openmpi/2.0.0                        |
| PE-gnu               | 1.0 (default)     | /software/dev_tools/swtree/cs400/modulefiles/PE-gnu/1.0(default)                  |
| PE-Intel             | 1.0 (default)     | /software/dev_tools/swtree/cs400/modulefiles/PE-intel/1.0(default)                |
| PE-nag               | 1.0 (default)     | /software/dev_tools/swtree/cs400/modulefiles/PE-nag/1.0(default)                  |
| PE-PGI               | 1.0 (default)     | /software/dev_tools/swtree/cs400/modulefiles/PE-pgi/1.0(default)                  |
| PGI                  | 15.7.0 (default)  | /software/dev_tools/swtree/cs400/modulefiles/pgi/15.7.0(default)                  |
| PGI                  | 15.7.0 (default)  | /software/dev_tools/swtree/cs400/modulefiles/pgi/15.7.0(default)                  |
| PostgreSQL           | 9.3.5 (default)   | /software/dev_tools/swtree/cs400/modulefiles/postgresql/9.3.5(default)            |
| Python               | 2.7.12 (default)  | /software/dev_tools/swtree/cs400/modulefiles/python/2.7.12(default)               |
| Python               | 2.7.13            | /software/dev_tools/swtree/cs400/modulefiles/python/2.7.13                        |
| Python               | 3.6.1             | /software/dev_tools/swtree/cs400/modulefiles/python/3.6.1                         |
| R                    | 3.3.2             | /software/dev_tools/swtree/cs400/modulefiles/R/3.3.2                              |
| sassena              | 1.4.2 (default)   | /software/dev_tools/swtree/cs400/sassena/1.4.2(default)                           |
| ScaLAPACK            | 2.0.2             | /software/dev_tools/swtree/cs400/scalapack/2.0.2                                  |
| Silo                 | 4.10.2            | /software/dev_tools/swtree/cs400/silo/4.10.2                                      |

|              |                      |                                                                 |
|--------------|----------------------|-----------------------------------------------------------------|
| Silo         | 4.9.1 (default)      | /software/dev_tools/swtree/cs400/silo/4.9.1(default)            |
| SPEC CPU2006 | 1.2                  | /software/dev_tools/swtree/cs400/SPEC2006/1.2                   |
| Subversion   | 1.9.4 (default)      | /software/dev_tools/swtree/cs400/subversion/1.9.4(default)      |
| SWIG         | 2.0.12<br>(default)  | /software/dev_tools/swtree/cs400/swig/2.0.12(default)           |
| Szip         | 2.1 (default)        | /software/dev_tools/swtree/cs400/szip/2.1(default)              |
| TensorFlow   | 0.11-cpu             | /software/dev_tools/swtree/cs400/tensorflow/0.11-cpu            |
| TensorFlow   | 0.11-cpu-<br>conda   | /software/dev_tools/swtree/cs400/tensorflow/0.11-cpu-conda      |
| TensorFlow   | 0.11-gpu             | /software/dev_tools/swtree/cs400/tensorflow/tensorflow/0.11-gpu |
| Theano       | 0.9-conda            | /software/dev_tools/swtree/cs400/theano/0.9-conda               |
| VinaMPI      | v2 (default)         | /software/dev_tools/swtree/cs400/vinampi/v2(default)            |
| VisIt        | 2.10.3<br>(default)  | /software/dev_tools/swtree/cs400/visit/2.10.3(default)          |
| VMD          | 1.9.3                | /software/dev_tools/swtree/cs400/vmd/1.9.3                      |
| wxPython     | 3.0.2.0<br>(default) | /software/dev_tools/swtree/cs400/wxpython/3.0.2.0(default)      |
| XALT         | 0.7.5 (default)      | /software/dev_tools/swtree/cs400/xalt/0.7.5(default)            |
| XALT         | 0.7.6                | /software/dev_tools/swtree/cs400/xalt/0.7.6                     |
| ZLIB         | 1.2.8 (default)      | /software/dev_tools/swtree/cs400/zlib/1.2.8(default)            |

## SHPC Condo LDAP Groups

To access CADES SHPC Condos, users need to be added to an XCAMS LDAP entry in the appropriate group. Find your group in the table below, and click on the respective XCAMS registration URL, as outlined in [Step 1: Request Your SHPC Cloud Allocation](#).

**Note:** If you do not see your group listed in the table, please [contact the CADES team](#) for registration assistance, and include the following information in your email:

- **Subject:** Help with SHPC Condo Registration
- **Email body:** UCAMS ID or XCAMS ID, contact information, reason for requesting an SHPC Condo allocation, and the name of your directorate and division.

| Division Name                                           | Division Approver                                      | XCAMS Registration URL         | Protection Zone | PBS Directives                                     |
|---------------------------------------------------------|--------------------------------------------------------|--------------------------------|-----------------|----------------------------------------------------|
| Computing and Computational Sciences Directorate (CCSD) | Jayson Hines<br>(hinesjb@ornl.gov)                     | <a href="#">Request Access</a> | Open            | #PBS -W<br>group_list=cades-ccsd<br>#PBS -A ccsd   |
| Spallation Neutron Source (SNS)                         | A.J. (Timmy) Ramirez-Cuesta<br>(ramirezcueaj@ornl.gov) | <a href="#">Request Access</a> | Open            | #PBS -W<br>group_list=cades-virtues<br>#PBS -A sns |
| Center for Nanophase Materials Sciences (CNMS)          | Bobby Sumpter<br>(sumpterbg@ornl.gov)                  | <a href="#">Request Access</a> | Open            | #PBS -W<br>group_list=cades-cnms<br>#PBS -A cnms   |
| Climate Change Science Institute (CCSI)                 | Dali Wang<br>(wangd@ornl.gov)                          | <a href="#">Request Access</a> | Open            | #PBS -W<br>group_list=cades-ccsi<br>#PBS -A ccsi   |
| Energy Dissipation to Defect Evolution (EDDE)           | Malcolm Stocks<br>(stocksgm@ornl.gov)                  | <a href="#">Request Access</a> | Open            | #PBS -W<br>group_list=cades-edde<br>#PBS -A edde   |
| Biosciences Division (BSD)                              | Jeremy Smith, Scott Painter, and Dan Jacobson          | <a href="#">Request Access</a> | Open            | #PBS -W<br>group_list=cades-bsd<br>#PBS -A bsd     |
| Nuclear Science and Engineering Directorate (NSED)      | Jeff Banta<br>(bantajp@ornl.gov)                       | <a href="#">Request Access</a> | Moderate        | #PBS -W<br>group_list=cades-nsed<br>#PBS -A nsed   |

[CADES](#) → [User Documentation](#) → [SHPC Condo Documentation](#) → [Troubleshooting](#)

## **Troubleshooting**

*This page is under construction*

## Execute a Job on Your SHCP Condo Allocation

The tutorial below shows you how to run Wes Kendall's basic "hello world" program, written in C, using the message passing interface (MPI) to scale across the SHPC Condo compute nodes [1]. This tutorial is intended for users who are new to the CADES SHPC Condo environment and leverages a portable batch system (PBS) script and a C source code.

**Note:** Do not execute jobs on the login nodes; only use the login nodes to access your compute nodes. Processor-intensive, memory-intensive, or otherwise disruptive processes running on login nodes will be killed without warning.

### Prerequisites

- Access to the login node (XCAMS/UCAMS authorization).
- A PBS script that specifies your conditions/variables and calls the binary/script you would like to execute on the compute nodes.
- A Binary file, source, or script for the problem that you would like to run on the compute nodes.

### Files

The steps below walk you through building a PBS script and compiling a C binary from source. However, if you wish to download the files used in this to tutorial files directly, you can do so using the links below.

- [hello-world.pbs](#) – PBS script used for the hello world batch job
- [MyMPIhw.c](#) – hello world C source code
- [hello-world](#) – hello world C binary (already compiled)

## Step 1: Connect to Your Allocation

Open and Moderate protection zones each have their own login node. Choose the login node for your protection zone.

**Note:** The Open protection zone can be accessed either using either XCAMS or UCAMS credentials. However, the Moderate protection zone requires an ORNL UCAMS ID.

### Open Protection Zone

1. Open a Bash terminal (or PuTTY for Windows users).
2. Execute `ssh xcams@or-condo-login.ornl.gov`.
  - Replace "xcams" with your XCAMS or UCAMS ID.
3. When prompted, enter your XCAMS or UCAMS password.

### Moderate Protection Zone

1. Open a Bash terminal (or PuTTY for Windows users).
2. Execute `ssh ucams@mod-condo-login.ornl.gov`.
  - Replace "ucams" with your UCAMS ID.
3. When prompted, enter your UCAMS password.

Once you have connected to the login node, you can proceed to Step 2 and begin assembling your PBS script.

## Step 2: Create Your PBS Script

Below is the PBS script we are using to run an MPI "hello world" program as a batch job. PBS scripts use variables to specify things like the number of nodes/cores used to execute your job, estimated wall time for your job, and which compute resources to use (e.g., GPU vs. CPU). The sections below feature an example PBS script for SHPC Condo resources, show you how to create/save your own PBS script, and show you how store the PBS script on an SHPC Condo file system.

Check out the [official Torque documentation](#) for a complete list of PBS variables</a>.

### Example PBS script

Here is an example PBS script for running a batch job on a CADES SHPC Condo allocation. We break down each command in the section below.

```
#!/bin/bash
#PBS -N MyMPIhw
#PBS -M YourEmailHere@ornl.gov
#PBS -l nodes=1:ppn=16
#PBS -l walltime=0:00:2:0
#PBS -W group_list=cades-birthright
#PBS -A birthright
#PBS -l qos=std
#PBS -q gpu
module purge
module load PE-gnu
module list
cd $PBS_O_WORKDIR
pwd
mpirun hello-world
```

Download this script (with explanatory comments) [here](#).

### PBS Script Breakdown

Here, we break down the essential elements of the above PBS script.

First, we're going to set the script type: Bash.

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

Next, you need to set the job name. Make it short and simple because your output files will share this name. We're going with **MyMPIhw**.

```
#PBS -N MyMPIhw
```

You can add your email address if you would like errors to be emailed to you directly.

```
#PBS -M YourEmail@ornl.gov
```

Set your node spec, including the number of nodes and processors per node that you want to use to run your job. In this case, we're using one node and 16 cores per node.

```
#PBS -l nodes=1:ppn=16
```

Tell PBS the anticipated runtime for your job, where `walltime=HH:MM:S`. The example below has the walltime set to 2 minutes.

```
#PBS -l walltime=0:00:2:0
```

Specify your LDAP group. The full list of SHPC Condo LDAP groups is [here](#). We're using `cades-birthright` in this case.

```
#PBS -W group_list=cades-birthright
```

You also need to specify your account type. We're also using `birthright` in this case.

```
#PBS -A birthright
```

Now we can set the quality of service (QOS). We can set this to `burst` or `std`.

Burst jobs allow a user to leverage more nodes/cores/GPUs than may be in their formal allocation. However, in exchange for this "resource burst" flexibility, your burst job may be preempted if the rightful owner of those resources needs them to complete his or her own jobs.

In most cases, a user will simply run a job with the QOS set to `std`.

```
#PBS -l qos=std
```

All Birthright SHPC Condo nodes have GPUs. Since this example uses a Birthright SHPC allocation, we're going to specify the use of GPUs.

```
#PBS -q gpu
```

Next we need to load the modules required for executing our batch job. First thing we'll do is clear any modules currently loaded that might result in a conflict.

```
module purge
```

With a clean slate, we can now load our programming environment using `module load`. For this particular example, all we need is the PE-gnu module, which loads OpenMPI, GCC, and XALT.

```
module load PE-gnu
```

The next line confirms the modules that were loaded.

```
module list
```

With our environment loaded, the PBS script now sets the working path. In this example, our binary will be launched from the same directory as our PBS script. The results from the binary will also be placed here.

```
cd $PBS_O_WORKDIR
```

Confirms current working directory.

```
pwd
```

Finally, the last line of the PBS script calls MPI to run our `hello-world` binary. You can replace "hello-world" with the file name of whatever binary you wish to execute on the compute nodes.

```
mpirun hello-world
```

## Procedure

Now that we've covered the basics of a PBS script in the context of an SHPC Condo, let's talk about actually creating and using the script on your allocation.

When creating and editing your PBS script, you have two basic options. **Option 1:** Create and edit your PBS script on your local machine and upload it to the Lustre path using secure copy (`scp`). **Option 2:** Create and edit your PBS script directly on the compute node (from Lustre storage) using Vi.

### Option 1: Create PBS Locally and Upload to Lustre Storage

1. Using your favorite text editor, create your PBS script on your local machine or [download the pre-made example script](#).
2. Use the `scp` command to copy your PBS script from the source machine to the SHPC file system (Lustre storage in the Open protection zone in this case).

```
$ scp /path/to/hello-world.pbs xcams@or-condo-login.ornl.gov:/lustre/or-hydra/cades-birthright/xcams/hello-world.pbs`
```

Replace "/path/to/hello-world.pbs" with the path of your script, and replace "xcams" with your XCAMS/UCAMS ID.

You may be prompted for your XCAMS/UCAMS password if you have not copied your public SSH key to the login node.

With the PBS script in place, you can now move on to [compiling your hello world C code](#).

### Option 2: Create Your PBS Script Directly on the Compute Node

1. From the login node, change your working directory to the desired file system. We're going to use our Lustre allocation for this example.

```
$ cd /lustre/or-hydra/cades-birthright/ucams
```

Replace "ucams" with your own UCAMS/XCAMS user ID.

2. Use Vi to create and edit your PBS script.

```
$ vi hello-world.pbs
```

3. Create your PBS script within Vi or paste the contents of your PBS script into Vi.
4. When finished, hit `Esc` on your keyboard to exit the input mode.
5. Enter `:x!` into Vi's command line, and press `Return` to save your file and return to the Bash shell.

With the PBS script in place, you can now move on to [compiling your hello world C code](#).

## Step 3: Compile the C Program from Source

Below is Wes Kendall's simple "hello world" C program that utilizes MPI to run the job in parallel [1]. We will need to compile this source code on one of the compute nodes (you can also download the compiled binary below if you prefer).

### MPI Hello World Source Code

```
#include <mpi.h>
#include <stdio.h>
int main(int argc, char** argv) {
    // Initialize the MPI environment.
    MPI_Init(NULL, NULL);
    // Get the number of processes.
    int world_size;
    MPI_Comm_size(MPI_COMM_WORLD, &world_size);
    // Get the rank of the process.
    int world_rank;
    MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);
    // Get the name of the processor.
    char processor_name[MPI_MAX_PROCESSOR_NAME];
    int name_len;
    MPI_Get_processor_name(processor_name, &name_len);
    // Print off a hello world message.
    printf("Hello world from processor %s, rank %d"
        " out of %d processors\n",
        processor_name, world_rank, world_size);
    // Finalize the MPI environment.
    MPI_Finalize();
}
```

Download this source [here](#).

Download the compiled binary [here](#).

## Procedure

When creating and editing your `hello-world.c` source code, you have two basic options. **Option 1:** Create and edit your source code on your local machine and upload it to the Lustre path using secure copy (`scp`). Compile the binary on the compute node. **Option 2:** Create and edit your source code directly on the compute node (from Lustre storage) using Vi. Compile the binary on the compute node.

### Option 1: Create Source Code Locally and Upload to Lustre Storage

1. Using your favorite text editor, create and edit your `hello-world.c` source code on your local machine or download the pre-made source [here](#).
2. Use the `scp` command to copy your `hello-world.c` source code from your local machine to the SHPC file system (Lustre storage in the Open protection zone in this case).

```
$ scp /path/to/hello-world.c xcams@or-condo-login.ornl.gov:/lustre/or-hydra/cades-birthright/xcams/hello-world.c`
```

Replace "/path/to/hello-world.c" with the path of your source code, and replace "xcams" with your XCAMS/UCAMS ID.

You may be prompted for your XCAMS/UCAMS password if you have not copied your public SSH key to the login node.

3. Load the MPI compiler using the PE-gnu module.

```
$ module load PE-gnu
```

4. Compile the C source into a binary.

```
$ mpicc -o hello-world hello-world.c
```

5. Use `ls -al` to verify the presence of the `hello-world` binary in your working directory.

With the C code compiled into a binary (`hello-world`), we can now schedule and run the job on our compute nodes.

## Option 2: Create and Edit Source Code Directly on Compute Node

1. Ensure that you are still in your working directory (`/lustre/or-hydra/cades-birthright/ucams`) using `pwd`.
2. Use Vi (`vi`) to create your C source file within your working directory.

```
$ vi hello-world.c
```

3. Paste the hello world C code into Vi.

- Hit `Esc` on your keyboard to exit the input mode.
- Enter `:set paste` into Vi's command line, and press `Return` to enter paste mode.
- Paste the C code into Vi.

4. When finished, hit `Esc` on your keyboard to exit the paste/input mode.

5. Enter `:x!` into Vi's command line, and press `Return` to save your file and return to the Bash shell.

You now have a C source file that you can compile.

6. Load the MPI compiler using the PE-gnu module.

```
$ module load PE-gnu
```

7. Compile the C source into a binary.

```
$ mpicc -o hello-world hello-world.c
```

8. Use `ls -al` to verify the presence of the `hello-world` binary in your working directory.

With the C code compiled into a binary (`hello-world`), we can now schedule and run the job on our compute nodes.

## Step 4: Run the Job

1. Before proceeding, ensure that you are still in your working directory (using `pwd`) and that you still have the PE-gnu module loaded (using `module list`).
  - We need to be in the same path/directory as our PBS script and our C binary. Use `ls -al` to confirm their presence.
  - PE-gnu also loads OpenMPI, GCC, and XALT. Use `module list` to confirm their presence. If necessary, use `module load PE-gnu` to reload the module(s).
2. Use `qsub` to schedule your batch job in the queue.

```
$ qsub hello-world.pbs
```

This command will automatically queue your job using Torque and produce a six-digit job number (shown below).

```
143295.or-condo-pbs01
```

You can check the status of your job at any time with the `checkjob` command.

```
$ checkjob 143295
```

You can also stop your job at any time with the `qdel` command.

```
$ qdel 143295
```

### 3. View your results.

Once your job completes, Torque will produce two output/data files. These output/data files, unless otherwise specified in the PBS script, are placed in the same path as your binary.

One file (`myscript.ojobnumber`) contains the results of the binary you just executed, and the other (`myscript.ejobnumber`) contains any errors that occurred during execution.

Replace "myscript" with the name of your script and "jobnumber" with your job number.

You can view the contents of these files using the `more` command followed by the file name.

```
$ more MyMPIhw.0143295
```

Your output should look something like this, with one line per processor core (16 in this case):

```
Hello world from processor or-condo-c136.ornl.gov, rank 3 out of 16 processors
Hello world from processor or-condo-c136.ornl.gov, rank 4 out of 16 processors
Hello world from processor or-condo-c136.ornl.gov, rank 6 out of 16 processors
Hello world from processor or-condo-c136.ornl.gov, rank 11 out of 16 processors
Hello world from processor or-condo-c136.ornl.gov, rank 7 out of 16 processors
Hello world from processor or-condo-c136.ornl.gov, rank 14 out of 16 processors
Hello world from processor or-condo-c136.ornl.gov, rank 2 out of 16 processors
Hello world from processor or-condo-c136.ornl.gov, rank 5 out of 16 processors
Hello world from processor or-condo-c136.ornl.gov, rank 8 out of 16 processors
Hello world from processor or-condo-c136.ornl.gov, rank 9 out of 16 processors
Hello world from processor or-condo-c136.ornl.gov, rank 10 out of 16 processors
Hello world from processor or-condo-c136.ornl.gov, rank 12 out of 16 processors
Hello world from processor or-condo-c136.ornl.gov, rank 13 out of 16 processors
Hello world from processor or-condo-c136.ornl.gov, rank 15 out of 16 processors
Hello world from processor or-condo-c136.ornl.gov, rank 0 out of 16 processors
Hello world from processor or-condo-c136.ornl.gov, rank 1 out of 16 processors
```

### 4. Download your results (using the `scp` command or an SFTP client) or move them to persistent storage.

## Works Cited

1. Wes Kendall, "MPI Hello World," *MPI Tutorial*, accessed June 14, 2017, <http://mpitutorial.com/tutorials/mpi-hello-world/>.

# Running Scientific Computational Workflows

## Overview

Workflows offer benefits of automation and efficient orchestration (eg. data parallel execution) of multi-stage computation. Furthermore, they are powerful reproducibility and portability tools for science and engineering applications.

Typically, a workflow is written in a high level language that is offered and understood by a workflow management software or simply a workflow tool.

## Workflow tools available on Condos

We currently offer support for the following workflow tools on SHPC:

1. Nextflow
2. Makeflow
3. Swift

A brief description about each of the aforementioned workflow tools is provided below:

### Nextflow

[Nextflow](#) is a favored workflow tool among Singularity container users. Similarly, it is popular among users from the Biosciences domain.

### Makeflow

The [Makeflow](#) workflow system uses a Makefile like language to define workflows that may be deployed and executed over clusters and clouds.

### Swift

[Swift](#) uses a C-like syntax to define workflows. Swift is capable of stitching computational steps defined in the workflow as a true HPC workflow that uses the Message Passing Paradigm of parallel computation using the MPI libraries and its own load balancer.

## Example Workflows

### Hello World

#### Nextflow

```
#!/usr/bin/env nextflow

params.str = 'Hello world!'

process splitLetters {
    output:
        file 'chunk_*' into letters mode flatten
```

```

"""
printf '${params.str}' | split -b 6 - chunk_
"""

}

process convertToUpper {

    input:
    file x from letters

    output:
    stdout result

    """
    cat $x | tr '[a-z]' '[A-Z]'
    """
}

result.subscribe {
    println it.trim()
}

```

Save the above code in a file, eg. `hello.nf`. To run the workflow on open condo login node, do the following:

```

$ module purge
$ module load PE-gnu
$ module load java/1.8.0_131
$ module load nextflow
$ nextflow run hello.nf

```

You should see output similar to the following:

```

N E X T F L O W ~ version 0.27.6
Launching `nextflow_example.nf` [insane_meucci] - revision: 5319db7b93
[warm up] executor > local
[f9/cb98ba] Submitted process > splitLetters
[94/6ed3f3] Submitted process > convertToUpper (1)
[cb/506a85] Submitted process > convertToUpper (2)
HELLO
WORLD!

```

## Makeflow

A "Hello World" in Makeflow would look something like so:

```

ECHO=/bin/echo

hello.txt:
    $ECHO 'Hello World!' > hello.txt

```

Save the above code in a file, say `hello.mkf` and run it on the open condo like so:

```

$ module load PE-gnu
$ module load cctools/6.2.7
$ makeflow hello.mkf

```

If all goes well, the output should look like so:

```

parsing hello.mkf...

```

```
local resources: 32 cores, 128833 MB memory, 6893119 MB disk
max running local jobs: 32
checking hello.mkf for consistency...
hello.mkf has 1 rules.
recovering from log file hello.mkf.makeflowlog...
makeflow: hello.txt is reported as existing, but does not exist.
starting workflow....
submitting job: /bin/echo ''Hello World!'' > hello.txt
submitted job 123822
job 123822 completed
nothing left to do.
```

And you should see a new file called `hello.txt` in your current working directory.

## Swift

A Swift Hello World workflow looks like so:

```
import io;
printf("Hello world");
```

Swift uses two steps to workflow execution: compile and run.

Load the swift module on condo like so:

```
$ module purge
$ module load PE-gnu
$ module load java/1.8.0_131 mpich/3.2
$ module load swift
```

Compile and run the workflow like so:

```
$ stc hello.swift
```

The above step will produce a TCL file called `hello.tic`. Run the TCL file like so:

```
turbine -n 2 hello.tic
```

If all goes well, you should see the following output:

```
Hello world
```

## General remarks

1. Note that the above workflows will run on login nodes. In order for them to run over compute nodes, more configuration is needed.
2. Note that `Nextflow` expects absolute paths for data and executables since it works in its own temp directory. Please adjust the paths to where you choose to run the workflow.

## Where to go from here?

- Use the [Crystal Workflow](#) with these workflow tools.
- Come talk to us at CADES if you think one or more of your applications will benefit with the help of the aforementioned workflow tools.



## Crystal Workflow

CrystalFlow is a hypothetical workflow with low-medium complexity that adequately illustrates the benefits and characteristics of scientific computational workflows.

### About the Crystal Workflow

The workflow is a crystal shaped graph as shown in the figure below.

```
graph TD;
P1-->P2;
P1-->P3;
P2-->P4;
P3-->P5;
P4-->P6;
P5-->P6;
```

In the above workflow, each of the boxes represent process and arrows represent the dependency between connected processes. For example, process `P1` produces a data file that is consumed by processes `P2` and `P3`.

Each of the 6 processes of this workflow are implemented in C and bash. Either may be used in the workflows shown below. The initial input file is pre-prepared. The code and data for these workflows are available on open SHPC condo at `/software/T/B/D`. A `Makefile` will build the C executables found in the directory named `c`. The following sections show how this workflow may be orchestrated using each of the three workflow management tools.

## Nextflow

The following code snippet shows how the above workflow would be expressed in Nextflow.

```
#!/usr/bin/env nextflow

in1 = file('/home/km0/crystalworkflow/shell/inputs/in1.txt')

process p1 {
    input:
        file in1
    output:
        file 'out1.txt' into out1
    """
    ~/crystalworkflow/shell/p1/p1.sh $in1 'out1.txt'
    """
}
process p2 {
    input:
        file out1
    output:
        file 'out2.txt' into out2
    """
    ~/crystalworkflow/shell/p2/p2.sh $out1 'out2.txt'
    """
}
process p3 {
    input:
        file out1
    output:
        file 'out3.txt' into out3
    """
}
```

```

~/crystalworkflow/shell/p3/p3.sh $out1 'out3.txt'
"""
}

process p4 {
    input:
    file out2
    output:
    file 'out4.txt' into out4
"""
~/crystalworkflow/shell/p4/p4.sh $out2 'out4.txt'
"""

}

process p5 {
    input:
    file out3
    output:
    file 'out5.txt' into out5
"""
~/crystalworkflow/shell/p5/p5.sh $out3 'out5.txt'
"""

}

process p6 {
    input:
    file out4
    file out5
    output:
    file 'out6.txt' into out6
"""
~/crystalworkflow/shell/p6/p6.sh $out4 $out5 'out6.txt'
"""

}

```

Assuming the above workflow is saved in a file named `crystal.nf`, it could be run as follows:

```

$ module purge
$ module load PE-gnu
$ module load java/1.8.0_131
$ module load nextflow
$
$ nextflow run crystal.nf

N E X T F L O W ~ version 0.27.6
Launching `crystal.nf` [thirsty_allen] - revision: e3b42d107d
[warm up] executor > local
[db/d513da] Submitted process > p1
[89/e16494] Submitted process > p2
[c3/9d4ddd] Submitted process > p3
[0d/5406b9] Submitted process > p4
[cf/4b94bb] Submitted process > p5
[c2/3bae00] Submitted process > p6

```

## Makeflow

The following code snippet shows how the crystal workflow would be implemented using Makeflow.

```

P1=../shell/p1/p1.sh
P2=../shell/p2/p2.sh
P3=../shell/p3/p3.sh
P4=../shell/p4/p4.sh
P5=../shell/p5/p5.sh
P6=../shell/p6/p6.sh

../shell/p1/out1.txt:
    $P1 ../shell/inputs/in1.txt ../shell/p1/out1.txt

```

```
..../shell/p2/out2.txt:  
    $P2 ..../shell/p1/out1.txt ..../shell/p2/out2.txt  
..../shell/p3/out3.txt:  
    $P3 ..../shell/p1/out1.txt ..../shell/p3/out3.txt  
..../shell/p4/out4.txt:  
    $P4 ..../shell/p2/out2.txt ..../shell/p4/out4.txt  
..../shell/p5/out5.txt:  
    $P5 ..../shell/p3/out3.txt ..../shell/p5/out5.txt  
..../shell/outputs/out6.txt:  
    $P6 ..../shell/p4/out4.txt ..../shell/p5/out5.txt ..../shell/outputs/out6.txt
```

Assuming the above workflow is saved in a file named `crystal.mkf`, it could be executed like so:

```
$ module purge  
$ module load PE-gnu  
$ module load cctools/6.2.7  
$ makeflow crystal.mkf  
parsing crystal.mkf...  
local resources: 32 cores, 128833 MB memory, 6593404 MB disk  
max running local jobs: 32  
checking crystal.mkf for consistency...  
crystal.mkf has 6 rules.  
starting workflow....  
submitting job: ..../shell/p6/p6.sh ..../shell/p4/out4.txt ..../shell/p5/out5.txt ..../shell/outputs/out6.txt  
submitted job 37132  
submitting job: ..../shell/p5/p5.sh ..../shell/p3/out3.txt ..../shell/p5/out5.txt  
submitted job 37133  
submitting job: ..../shell/p4/p4.sh ..../shell/p2/out2.txt ..../shell/p4/out4.txt  
submitted job 37134  
submitting job: ..../shell/p3/p3.sh ..../shell/p1/out1.txt ..../shell/p3/out3.txt  
submitted job 37135  
submitting job: ..../shell/p2/p2.sh ..../shell/p1/out1.txt ..../shell/p2/out2.txt  
submitted job 37136  
submitting job: ..../shell/p1/p1.sh ..../shell/inputs/in1.txt ..../shell/p1/out1.txt  
submitted job 37137  
cat: ..../shell/p3/out3.txt: No such file or directory  
p3 completed.  
p5 completed.  
p4 completed.  
job 37135 completed  
p1 completed.  
p2 completed.  
job 37134 completed  
job 37133 completed  
job 37136 completed  
job 37137 completed  
p6 completed.  
job 37132 completed  
nothing left to do.
```

## Swift

The following code snippet shows the Swift implementation of the crystal workflow. Note that the Swift implementation invokes the C version of executables but it can equally invoke the bash version.

```
import io;  
  
app (file out) p1 (file inp){ ".../c/p1/p1" inp out }  
  
app (file out) p2 (file inp){ ".../c/p2/p2" inp out }  
  
app (file out) p3 (file inp){ ".../c/p3/p3" inp out }  
  
app (file out) p4 (file inp){ ".../c/p4/p4" inp out }
```

```
app (file out) p5 (file inp){ "../c/p5/p5" inp out }

app (file out) p6 (file inp1, file inp2){ "../c/p6/p6" inp1 inp2 out }

file in1 = input("../c/inputs/in1.txt");

file out1 <"../c/p1/out1.txt">      = p1(in1);
file out2 <"../c/p2/out2.txt">      = p2(out1);
file out3 <"../c/p3/out3.txt">      = p3(out1);
file out4 <"../c/p4/out4.txt">      = p4(out2);
file out5 <"../c/p5/out5.txt">      = p5(out3);
file out6 <"../c/outputs/out6.txt"> = p6(out4,out5);
```

Assuming the above program is saved in a file called `crystal.swift`, it may be run on Open SHPC like so:

```
$ module purge
$ module load PE-gnu
$ module load java/1.8.0_131 mpich/3.2
$ module load swift

$ stc crystal.swift

$ turbine -n 2 crystal.tic
./c/p1/out1.txt
./c/p3/out3.txt
./c/p2/out2.txt
./c/p5/out5.txt
./c/p4/out4.txt
./c/outputs/out6.txt
```

## Birthright Cloud User Policy

Oak Ridge National Laboratory's (ORNL) Compute and Data Environment for Science (CADES) now provides eligible customers with an OpenStack cloud computing solution with customizable virtual machines (VM). This new resource, called "Birthright Cloud," enables customers in science and technology directorates to leverage self-service portals to rapidly request these VMs for production, testing, and development.

Cloud computing provides an efficient pooling of on-demand, self-managed virtual infrastructure, consumed as a service. The OpenStack platform used here is an open-source cloud computing software solution that allows the creation of individual "Project" allocations for each user. Users can then fill these Project allocations with their own VMs without further intervention from CADES administrators—a true self-service implementation.

The CADES OpenStack Birthright Cloud allocations provide:

- **Self Service** – Through the Horizon web interface, users can create, manage, and delete VMs.
- **Portability** – Operations can be performed using any local system that provides a Bash terminal and SSH functionality.
- **Elasticity** – Users can create VMs on demand and delete them when they are no longer needed.

## Disclaimers

If a concept or feature is not explicitly described within this policy, then it is not explicitly supported by the CADES team.

The only official copy of this document is this online electronic version found on <http://support.cades.ornl.gov>.

This policy is subject to change.

## Acceptable Use

Computers, software, and communications systems provided by CADES are to be used for work associated with, and within the scope of, an approved project. The use of CADES resources for personal or non-work-related activities is strictly prohibited. All computers, networks, email, and storage systems are property of the US Government. Any misuse or unauthorized access is prohibited and is subject to criminal and civil penalties. CADES systems are provided to users without any warranty. CADES will not be held liable in the event of any system failure or data loss or corruption for any reason, including, but not limited to: negligence, malicious action, accidental loss, software errors, hardware failures, network losses, or inadequate configuration of any computing resource or ancillary system.

## User Responsibilities

All Birthright Cloud users must comply with ORNL security rules and with the following:

- All operating system patches must be applied according to ORNL patching requirements.
- If user-sourced software images are uploaded, the user is responsible for keeping a copy of the image in case of accidental deletion or corruption.
- No moderate/confidential data should be mounted or copied to the VMs. Open science only.
- VMs should be removed from your OpenStack Project when they are no longer needed.
- VM operating systems must be updated or migrated before they reach an end-of-life development status.

## Application for Resources

Birthright Cloud allocations are available to ORNL research and technical staff, by request, through CADES. The request is made through the ORNL XCAMS portal and requires your UCAMS ID. An activation notice will be sent when your resources are ready for use. CADES reserves the right to throttle access to Birthright Cloud allocations as resource constraints require.

| How to: [Request Your CADES Birthright Cloud Allocation](#)

## Authentication and Authorization

Users can access their Birthright Cloud allocation using a web-based GUI called "Horizon." See the Birthright Cloud user guide for details.

| How to: [Manage Your OpenStack Project in Horizon](#)

Users are prohibited from changing or circumventing access controls to allow themselves or others to perform actions outside of their authorized privileges. In the event that an account is compromised, users must notify the CADES support team ([cades-help@ornl.gov](mailto:cades-help@ornl.gov)) immediately.

Users should also promptly inform the CADES support team ([cades-help@ornl.gov](mailto:cades-help@ornl.gov)) of any changes in their contact information (email, phone, affiliation, etc.).

The CADES team reserves the right to terminate accounts if any terms of this policy are violated.

**Note:** DO NOT share your credentials, passwords, private keys, or certificates, with anyone.

## Account Access Maintenance

ORNL staff who have been granted a Birthright Cloud allocation have indefinite access to these resources for the duration of their time at ORNL and/or for as long as these resources are made available through CADES.

As underlying technologies and platforms change, users may be required to perform account access maintenance as needed.

## Access at the End of a Project

When a user leaves ORNL, their Birthright Cloud allocation will close out, which results in the termination of account access and deletion of any remaining VMs running in their allocation. The user should move or save any data that he or she wishes to keep **before leaving ORNL**.

## Computing Policy

CADES provides public VM images for Birthright Cloud customers. These images have been customized for better integration into both the ORNL environment as well as the user's scientific workflow, and are the only images fully supported by CADES. CADES will not provide support for user-provided images. If you still want to run a custom cloud image, or if you would like to inquire about migrating an image from an existing VM resource, please contact the CADES support team ([cades-help@ornl.gov](mailto:cades-help@ornl.gov)) for your request.

Because of the highly heterogeneous hardware architecture of CADES resources (in terms of processors, network interconnects, and disk technologies), and the fact that some of these resources are shared or may be overcommitted, CADES provides no guaranteed minimum performance level. However, if your application requires a certain level of performance that you have not

been able to obtain using your Birthright Cloud allocation, please contact the CADES support team ([cades-help@ornl.gov](mailto:cades-help@ornl.gov)) for assistance.

## Storage Policy

The Birthright Cloud allocations, like other cloud solutions available on the market, provide a fungible resource that is subject to certain reliability constraints. Valuable data should be stored on a secondary storage solution, not exclusively on a user's Birthright Cloud VM Instance.

No moderate/confidential data should be mounted or copied to your Birthright Cloud VM Instance. Birthright Cloud VM Instances are for open science.

## Storage Allocation

From inception, each Birthright Cloud allocation has a set storage quota. CADES reserves the right to change this storage quota at any time. See the [Birthright Cloud User Documentation](#) for details.

If a user requires more storage for their allocation, he or she can submit a proposal to the CADES Resource Utilization Council to request a storage quota increase. This proposal should describe the amount of storage desired and the scientific goal and merit of the work being performed using the CADES Birthright Cloud allocation. These requests will be reviewed by the council on a case-by-case basis.

## Data Retention

When a project ends and a Birthright Cloud allocation is closed out, account access is terminated, and any remaining VMs and their associated data are deleted. Users are responsible for moving or saving any data that they would like to keep **before their project ends and their allocation is closed out**.

## Backups

The Birthright Cloud allocations, like other cloud solutions available on the market, provide a fungible resource that is subject to certain reliability constraints. Valuable data should be stored on a secondary storage solution, not exclusively on a user's Birthright Cloud VM Instance.

At this time, there are no CADES-supported provisions for automatic backups of the VMs or their data. The user is responsible for backing up data and instances to their desired secondary storage solution.

## Purge Policy or Quota

From inception, each Birthright Cloud allocation has a set quota, and CADES reserves the right to change this storage quota at any time. See the Birthright Cloud user documentation for details.

| How to: [View your OpenStack Project Quota](#)

If a user requires more resources for their allocation, he or she can submit a proposal to the CADES Resource Utilization Council to request a quota increase. This proposal should describe the resources desired (RAM, CPUs, storage) and the scientific goal and merit of the work being performed using the CADES Birthright Cloud allocation. These requests will be reviewed on a case-by-case basis.

## Special Requests and Policy Exemptions

Users can request policy exemptions by contacting the CADES support team ([cades-help@ornl.gov](mailto:cades-help@ornl.gov)). Requests are subject to review by the CADES Resource Utilization Council.

## Acknowledging CADES

The following acknowledgment should be included in publications and presentations that contain work performed using CADES resources.

This research used resources of the Compute and Data Environment for Science (CADES) at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.