Research and High Performance Computing with Alces Flight Compute on AWS

How to deploy, configure and start computing

Vaughan Jones - Alces Flight Ltd
May 2016

CONTENTS

Overview

The following paper is aimed at those who either already run their workloads in the Cloud, or are wishing to run their compute jobs in the Cloud. Alces Flight Compute environments are designed to provide the tools and utilities required for users of all skill levels; making research and High Performance Computing in the Cloud as simple as possible.

Alces Flight

Alces Flight Cloud software aims to reduce the complexity and skills required to get started with Cloud computing - providing ready-to-go compute environments using popular Cloud providers including AWS.

The Alces Flight range of Cloud software appliances have been designed with usability in mind - taking the time consuming configuration away and instead providing vital features required to get started with your research.

Leveraging popular and traditional techniques - the Alces Flight software appliances deliver an unparalleled computing experience to users and researchers of all backgrounds and skill levels.

Computing on AWS

AWS is the fastest growing, and largest public Cloud provider - making AWS the perfect setting for Cloud computing. With multiple services available, and a large pool of resource including multiple types of specialised instance types - compute clusters tailored to your compute workload requirements are easily obtainable for the time you require.

Alces Flight on AWS

The Alces Flight software appliances manage the deployment and configuration of your AWS instances - automatically configuring each deployed host in the environment. By pre-configuring your compute environments - core hours can be better spent doing research rather than configuration.

The Alces Flight Compute appliance for Linux is a scalable, self-configuring cloud-ready High Performance Computing environment - built on traditional, well-known Linux tools and utilities. The Alces Flight Compute appliance is a multi-functional AMI; acting as either a cluster master node - hosting cluster scheduler services and providing shared user data and applications, or as a cluster compute host to run your batch workloads on.

An Alces Flight compute environment provides many useful and popular features, including;

- **Batch scheduler installation and configuration;** Open Grid Scheduler is currently available, which is automatically configured including queue setup with known-working settings. Other

popular cluster schedulers will be made available with later releases of the Alces Flight Compute appliance including SLURM, Torque and OpenLava.
- **Automatic compute host scaling**; Alces Flight compute environments dynamically grow and shrink in size based on
- **Shared user data**; each cluster is configured with a shared home directory - exported to each of the nodes via NFS. Additional storage spaces can also be configured from external sources including parallel filesystems and additional block storage volumes.
- **Alces Gridware**; the Alces Gridware utility is a tool that makes the installation of Linux applications, accelerated libraries and compilers simple and effortless. Gridware packages are served through the Linux *modules* environment. The Alces Gridware repository contains over 850 packages spread over a wide range of research use types including bioinformatics tools, chemistry applications, engineering applications and much more.
- **Storage management utilities**; also included is the `alces storage` utility - providing seamless use and access of multiple storage tiers including local POSIX storage and remote object storage targets, allowing you to easily manage your data across multiple tiers of storage without having to learn different tools and APIs.

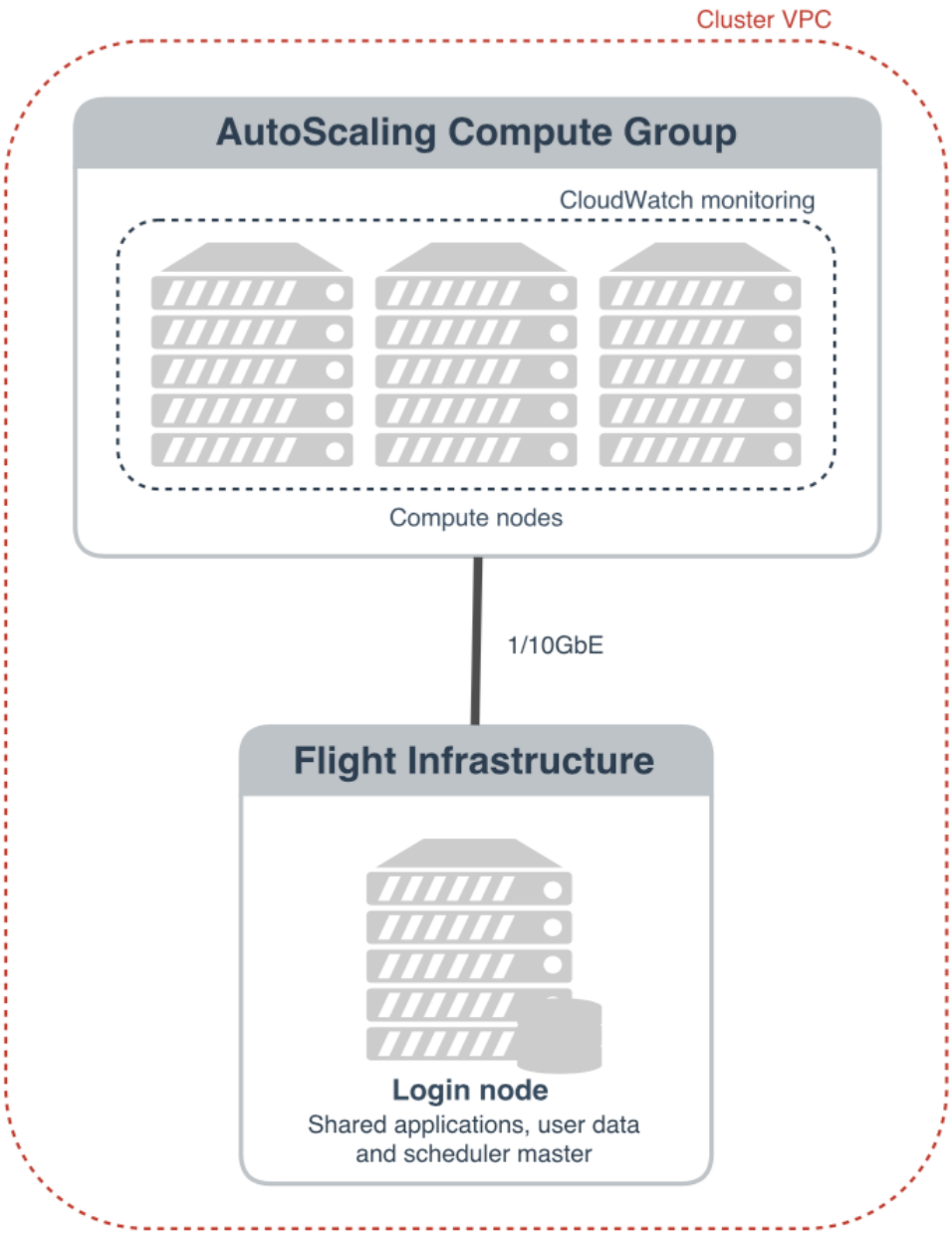
Many other tools and utilities are included with an Alces Flight compute environment to help you with your research and environment deployment.

Useful links

Alces Flight environment documentation
http://alces-flight-appliance-docs.readthedocs.org/en/latest/

AWS Spot Instances
https://aws.amazon.com/ec2/spot/

Deployed Architecture



Deployment Details

AWS services used

AWS EC2 - Compute, Image and block storage services

AWS EC2 is the Amazon compute service - providing an interface to interact with compute instances, images (AMIs) and block storage volumes (EBS) as well as provide useful features such as AutoScaling instance groups.

Some of the features of EC2 used by an Alces Flight Compute environment on AWS include;

- **AMI service**; each release of the Alces Flight Compute appliance is stored as an AMI - used to launch instances from
- **Compute service**; login, compute and storage nodes are launched using the AWS compute service
- **AutoScaling compute groups**; dedicated compute hosts are launched into AutoScaling Compute groups, allowing compute clusters to grow and shrink in sized based on load and demand
- **Block storage service**; the block storage service (EBS) is used both to host instance root volumes, as well as provide additional data storage volumes

AWS VPC - Networking

AWS VPC is the networking service provided by Amazon. The VPC service provides several features including private networks, security groups and much more. An Alces Flight Compute environment is typically deployed into a dedicated VPC, keeping it separate from other running machines on your existing networks.

Some of the features of VPC used by an Alces Flight Compute environment on AWS include;

- **VPC network**; VPC allows you to create a virtual network, completely segregated from other networks both within your account and on AWS. Alces Flight Compute environments are typically deployed into a single cluster network.
- **Security groups**; security groups are used to limit traffic in and out of instances, acting as a secondary layer of defence to the Linux firewall included with an Alces Flight Compute environment. Each environment is configured with well-known working settings to keep your environment secure.

AWS CloudWatch - Monitoring

AWS CloudWatch is the AWS monitoring service - allowing a wide range of both pre-set and custom metrics to be collected and used to scale your environment with.
An Alces Flight Compute environment automatically sends scheduler metrics to CloudWatch, which allows the dedicated compute hosts to automatically scale up and down based on cluster scheduler load.

Many other metrics can be collected, including; storage utilisation, CPU load, memory usage etc.

AWS IAM - Instance role management

AWS IAM is the user access and control service - primarily used to grant additional users access to an AWS account. IAM can also be used to provide certain permissions to AWS services to EC2 compute instances.

In an Alces Flight Compute environment - the cluster login node is provided several permissions to talk to different AWS services to perform various actions, including;

- **AutoScaling group permissions**; the cluster login node is able to adjust the size of the AutoScaling group in order to correctly scale up and down based on cluster scheduler usage. As well as adjusting the size of groups - the login node is also able to protect and unprotect compute instances from termination
- **CloudWatch write permissions**; the cluster login node is able to write metrics to the CloudWatch service

CloudFormation template

AWS CloudFormation provides a method to easily launch and manage multiple services by creating a 'template'. The template instructs the CloudFormation service which resources to create with the required settings and customisations for an Alces Flight Compute environment.

CloudFormation also allows you to create the same base infrastructure each time; as well as managing the deletion of all resources from your AWS account - ensuring no services are left running.

Template options

The following templates are available to use, each with slightly different configurations - please see the below table for details on each template:

Template	Compute instance cores	Compute instance memory	Usage
8 node template	2 per instance	4GB per instance	Evaluation or testing
16 node template	16 per instance	30GB per instance	Research and High Performance Computing
32 node template	36 per instance	60GB per instance	

Deploying an Alces Flight Compute Environment

Prerequisites

The following section lists requirements for launching an Alces Flight Compute environment on AWS using CloudFormation. Please ensure each of these requirements have been met before launching an environment.

Amazon AWS account

You should have access to an AWS account - whether it is shared or your own personal account. If logging in as an IAM user - please ensure you have the appropriate permissions to use each of the services listed above; and have the account holders permission.

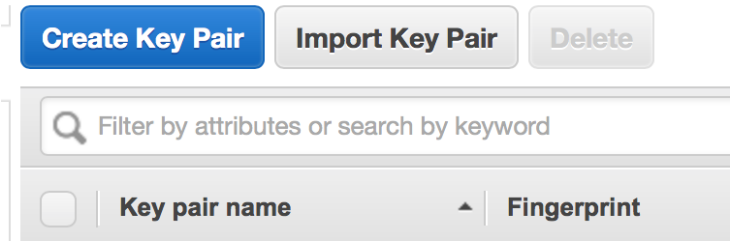
The `eu-west (Ireland)` region should also be selected before continuing with launching an environment.

AWS Keypair

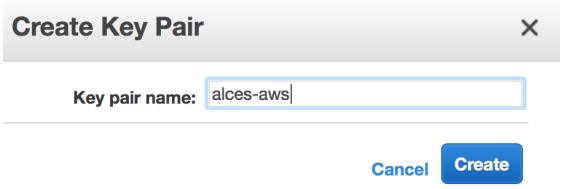
An SSH key pair should be added to your AWS account, either by creating a new one or importing your existing SSH keypair. This can be done by navigating to the EC2 console, then selecting the *Key Pairs* page in the *Network & Security* section.

Creating a new key pair

To create a new key pair and download to your workstation - click the *Create Key Pair* option:



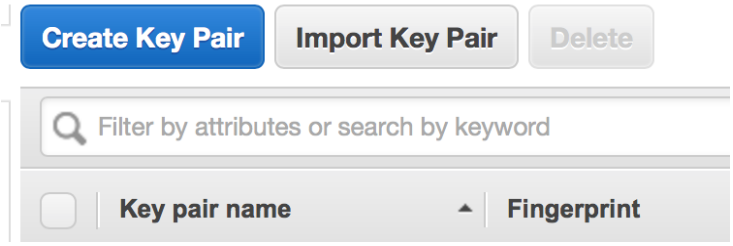
Next - enter a *Key Name* then press *Create*:



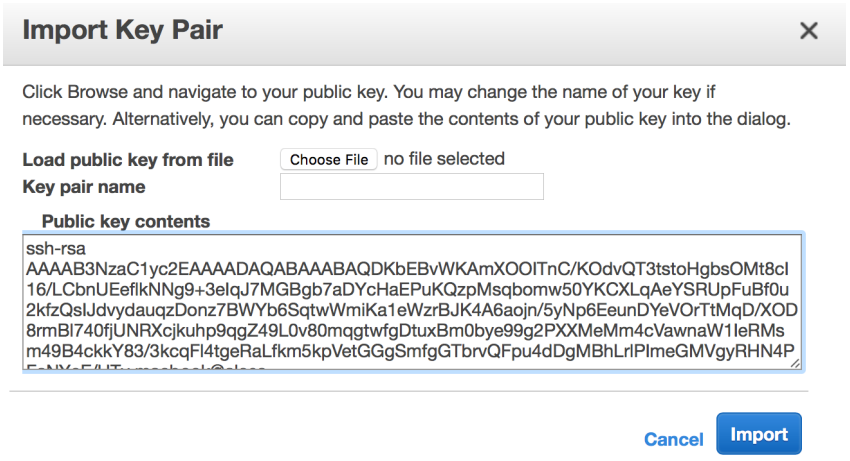
The key pair will now be automatically downloaded to your workstation. Store it in an appropriate location on your machine - with correct permissions. For a UNIX machine - `0640` typically suffices.

Importing an existing key pair

To use your existing SSH key pair to access any created AWS instances - click the *Import Key Pair* option:



Next - either browse to the location of your SSH public key, or manually enter the contents of your SSH public key. A name for the key pair should also be provided.



Finally - select the *Import* button to finish adding your key pair to your AWS account.

SSH Client

Access to an Alces Flight Compute environment is typically gained via SSH; as such - an SSH client is required on your local workstation.

- **Windows clients** - for Windows clients, we typically recommend the PUTTY client (http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html)
- **Mac OS X/Linux clients** - most Mac and Linux clients have the SSH package installed by default

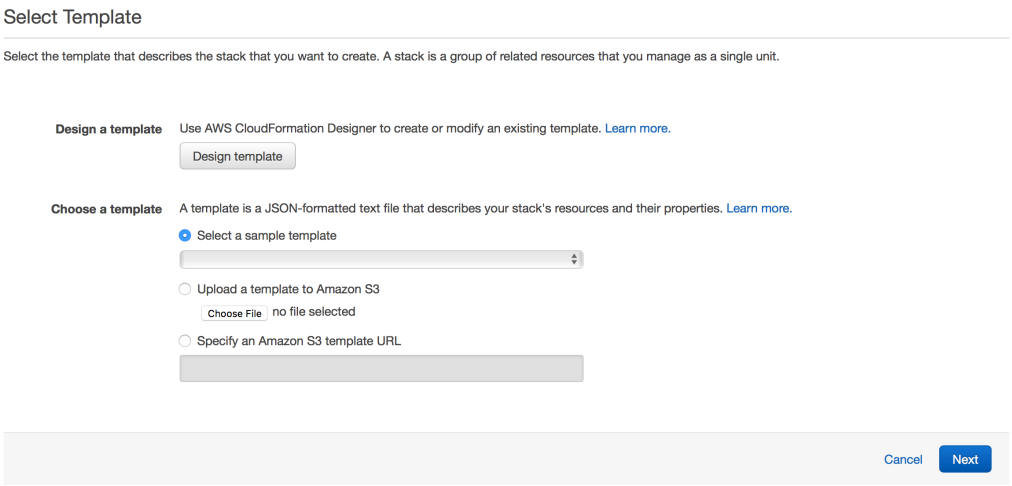
Launch the stack

With the above prerequisites met - a CloudFormation stack can now be deployed; using one of the following CloudFormation templates:

Small cluster: https://s3-eu-west-1.amazonaws.com/flight-aws-marketplace/2016.1/8-node.json
Medium cluster: https://s3-eu-west-1.amazonaws.com/flight-aws-marketplace/2016.1/16-node.json
Large cluster: https://s3-eu-west-1.amazonaws.com/flight-aws-marketplace/2016.1/32-node.json

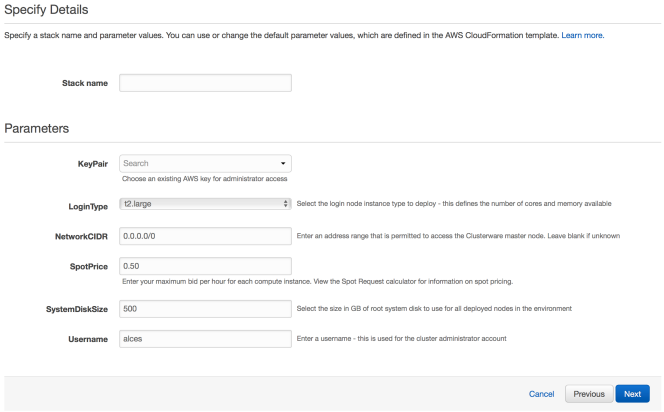
Each of the templates use the same basic services, with compute host instance types and numbers slightly modified for each template size.

Navigate to the *CloudFormation* console within your AWS account, then click the *Create Stack* button to bring up the template launch wizard:



Select the *Specify an Amazon S3 template URL* option - then enter one of the above URLs and select *Next*.

You will then be taken to the configuration page, which offers a selection of options used to modify your environment:



Enter the options displayed, with your own preferences:

- **Stack name**; the stack name defines both the unique identifier for the CloudFormation stack within your AWS account, as well as setting the cluster name. Enter your desired cluster name e.g. `aws-cluster`
- **Key Pair**; choose from the list your previously created or imported AWS keypair. This is added to each instance and used for administrator access to the instances
- **Login Type**; this defines the login node instance type - defining the number of CPU cores and memory available to the login node.
- **Network CIDR**; the network CIDR option allows you to restrict access to your cluster to a specific IP range. Use this option for increased security to your environment.
- **Spot Price**; enter the price (in USD) per hour that you are prepared to pay for each compute instance
- **SystemDiskSize**; this defines the amount of shared storage available to each of the nodes within the environment, providing additional storage for both Gridware applications and shared user data
- **Username**; the username field sets the default administrator username - used to log in with the provided AWS keypair

Click *Next* to take you to the *Tags* page. Here you can set a tag for your AWS stack - identifying it within your AWS account. Once complete, click the *Next* button to head to the *Review* page.

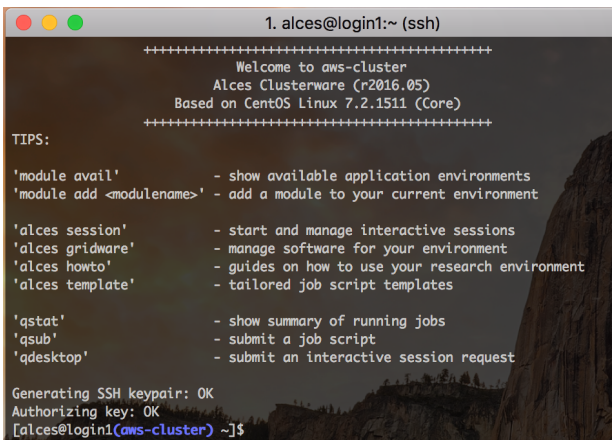
Once you have verified all of your chosen settings - click the *Launch* button to begin launching your cluster.

You can view the progress of the stack creation from the *Events* tab within the CloudFormation console for your created stack.

Accessing the environment

Once the stack is marked `CREATE_COMPLETE` - you can begin accessing your environment. From the *Outputs* tab in the CloudFormation console, select the `AccessIP` output. This is the public IP address of your cluster login node - used to access your environment.

Using your SSH client - use your AWS key pair, administrator username and public access IP to SSH to your cluster login node:



Computing with an Alces Flight Compute environment

The following section details some of the basic workflows possible with an Alces Flight Compute environment on AWS - covering application installation, running graphical jobs, running interactive compute jobs and running MPI jobs.

Installing applications with Alces Gridware

The Alces Gridware utility allows you to easily install your favourite Linux applications, libraries and compilers without the complication of having to manually compile code and figure out dependencies.

Each Alces Flight Compute environment includes the Gridware utility - which can be invoked using the `alces gridware` command. Many Gridware commands are available - but the following section will focus on the installation of a basic Gridware package.

The Gridware utility handles software compilation for you, often optimising for the architecture you are compiling on - providing the best performing applications for your environment rather than pre-compiled software.

To view the list of available Gridware packages, use the following command:

```
$ alces gridware list
```

To install one of the listed applications, use the `alces gridware install <package>` command as follows, for example - to install the disk benchmarking application IOZone:

```
$ alces gridware install apps/iozone/3.420
Installing base/apps/iozone/3.420

 > Preparing package sources
        Download --> iozone3_420.tar ... OK
          Verify --> iozone3_420.tar ... OK

 > Preparing for installation
           Mkdir ... OK (/var/cache/gridware/src/apps/iozone/3.420/
gcc-4.8.5)
         Extract ... OK
           Patch --> alces-iozone-ssh.patch ... OK

 > Proceeding with installation
         Compile ... OK
                Mkdir ... OK (/opt/gridware/depots/2c0ed688/el7/pkg/
apps/iozone/3.420/gcc-4.8.5)
         Install ... OK
          Module ... OK

Installation complete.
```


The IOZone version 3.420 application will now be available as an application module:

```
$ module avail
---  /opt/gridware/local/el7/etc/modules  ---
  apps/iozone/3.420/gcc-4.8.5
  compilers/gcc/system
  libs/gcc/system
---  /opt/clusterware/etc/modules  ---
  services/gridscheduler
```

Gridware will also warn of any potential dependencies on other Gridware packages, for example if trying to install the High Performance Linpack (HPL) application - an error will occur if the appropriate prerequisite packages are not met:

```
$ alces gridware install apps/hpl
Installing base/apps/hpl/2.1
ERROR: Unable to satisfy compilation requirements: mpi/openmpi, libs/atlas
```

Alces Gridware is unparalleled in its function and usage, and will empower your research productivity - by allowing you to repeatedly install applications, libraries and compilers on each of your cloud clusters with no knowledge of software compilation required. Its as simple as `alces gridware install`.

Running a graphical job

> *Note - the* `demo` *Gridware depot is required in order to perform the tasks described in the following section*

Included with an Alces Flight Compute environment is the `alces session` tool - which significantly simplifies launching and managing graphical desktop sessions.
From your cluster login node - create a GNOME desktop session using the `alces session` utility, for example:

```
$ alces session start gnome
VNC server started:
    Identity: d95e5808-070f-11e6-a476-0603cab3dfff
        Type: gnome
        Host: 52.50.197.102
        Port: 5901
     Display: 1
    Password: lhghoW0E
   Websocket: 41361

Depending on your client, you can connect to the session using:

  vnc://alces:lhghoW0E@52.50.197.102:5901
```



```
52.50.197.102:5901
52.50.197.102:1

If prompted, you should supply the following password: lhghoW0E
```

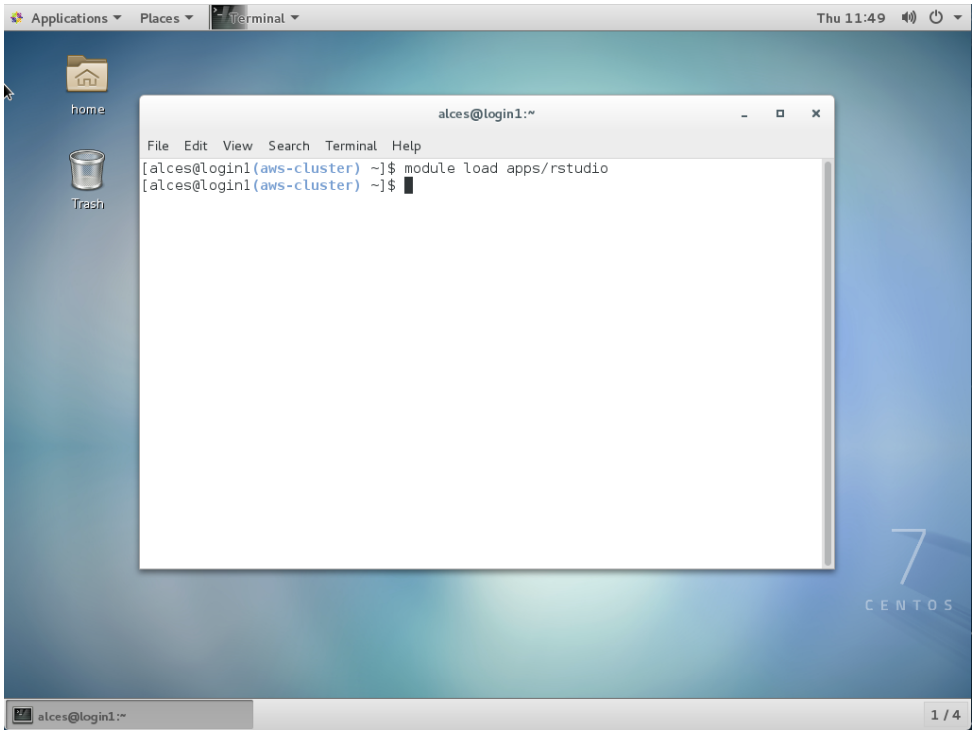
The session start output will provide you useful information used to connect to your session with your favourite VNC client. Mac OS X users can simply navigate to the `vnc://<user>:<password>@<public IP>` address to open the built-in VNC viewer.

You can view the currently enabled desktop session types using the `alces session avail` command, for example:

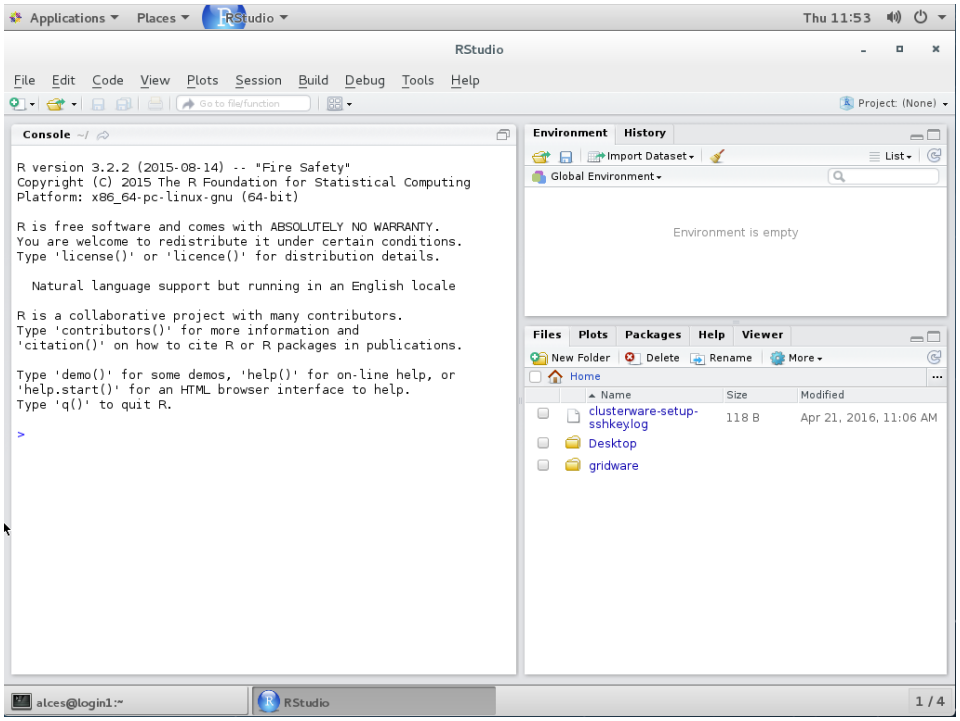
```
$ alces session avail
[ ] base/chrome
[ ] base/cinnamon
[*] base/default
[ ] base/fvwm
[*] base/gnome
[ ] base/icewm
[ ] base/terminal
[ ] base/trinity
[ ] base/xfce
```

Additional session types can be enabled using the `alces session enable <name>` command.

Once you have connected to your VNC session - open the *Terminal* application then load the R*Studio* application module:

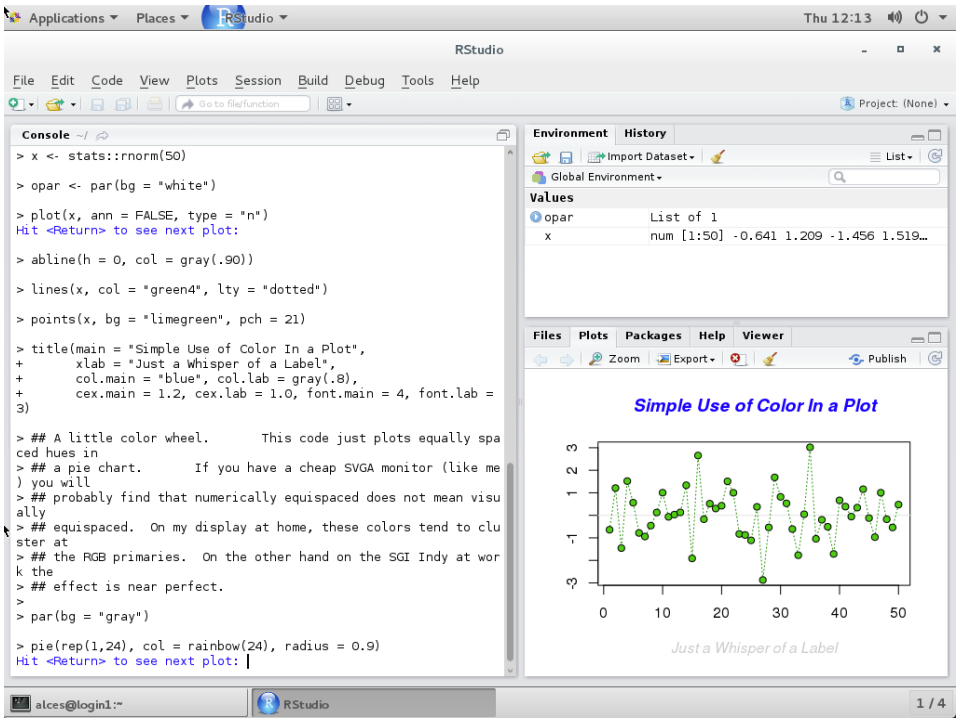


Next - load the *RStudio* application by typing `rstudio` into the command line prompt - which will load the *RStudio* window:



Next - using the RStudio command prompt, enter the following to begin the RStudio graphical demonstration:

```
$ demo("graphics")
```



Running an interactive job

> *Note - the* `apps/fah` *Gridware application should be installed in order to follow the next section*

An Alces Flight Compute environment allows you to easily run interactive jobs or applications. From the login node - start a session on one of the dedicated compute hosts, this can be done using the cluster scheduler. To use an entire compute node, run the following command from your Flight Compute login node:

```
$ qrsh -l excl=true
```

Once you have gained your interactive session - load the Folding at Home Gridware application using the `module load` command:

```
$ module load apps/fah
apps/fah/6.34/bin
 |
 OK
```

You can then invoke the Folding at Home application with appropriate options:

```
$ fah6 -smp -advmethods
```

The default options can be used when performing the initial setup. Once setup is complete, the Folding at Home application will begin fetching work - you can watch the live output as you contribute computing cycles towards a cure for Alzheimer's, Parkinson's and many cancers.

Running an MPI job over multiple nodes

> *Note - the* `demo` *Gridware depot is required in order to perform the tasks described in the following section*

Many jobs can benefit from running over multiple cores or even compute hosts. This is achieved using various MPI libraries. The following example uses the `OpenMPI` Gridware application - although many other MPI options are available.

The following section will detail running the Intel Messaging Benchmark (IMB), which measures latency and bandwidth between multiple compute hosts.

To begin - write a simple job script which loads the IMB Gridware application, and runs it over two compute nodes using the MPI library. Save the job script in your home directory:

```
#!/bin/bash -l
#$ -pe mpinodes-verbose 2
```



```
#$ -N imb-2nodes -o $HOME/imb.$JOB_ID.out
module load apps/imb
module load mpi/openmpi
mpirun -np 2 -npernode 1 IMB-MPI1
```

Next - submit the job script to the cluster scheduler. The job script contains the appropriate scheduler directives including;

- Parallel environment to use
- What to name the job, which appears in querying running jobs
- Where to write output data to
- Which Gridware applications to load

```
$ qsub imb.sh
Your job 2 ("imb") has been submitted
```

You can watch the live output of the job by viewing the `$HOME/imb.$JOB_ID.out` file, for example:

```
$ tail -f $HOME/imb.2.out
```

Start computing with Alces Flight today

This paper details a few examples of possible activities within an Alces Flight Compute environment. Many other workloads are possible - allowing you to get started with your research in the Cloud today.

