Customising Alces Flight Compute on AWS

How to configure and customise your environment

Alces Flight - 2016.2

Vaughan Jones - Alces Flight Ltd

June 2016

Revision number	Release date	Release notes	Author(s)
1.0	June 2016	Initial release	Vaughan Jones

CONTENTS

Overview

The Alces Flight Compute product on Amazon Web Services (AWS) Marketplace provides a set of well-known working default options, aimed at getting users and researchers up and running in the fastest possible time. The default options are there for general purpose usage, with no particular optimisations or customisations for specific workloads in mind. This whitepaper covers using the provided tools and utilities to customise your environments including resource and instance level customisation.

The level of customisation ranges from modifying the CloudFormation templates to adjust the parameters and resource options set to use different types of storage or instance types - to advanced OS level customisation such as configuring storage mounts for external storage clusters. Basic customisations such as Gridware package installations are also included in this white paper for repeatability and ease of deployment when dealing with multiple ephemeral Alces Flight Compute environments.

Also covered in this whitepaper is the Alces Flight Compute customisation tool, which is included with each deployed environment, allowing you to easily run install scripts on each of your deployed clusters

This whitepaper also includes advanced customisation including the modification of the base Alces Flight Compute Amazon Machine Image (AMI) to include customisations and updates. The provided Alces Flight Compute AMI is shipped with the latest updates at the time of publication; you may wish to use your own custom Alces Flight Compute AMI with the latest updates including security patches to meet organisational requirements. This whitepaper will detail the method of deploying your own Alces Flight Compute AMI.

Intended audience

This whitepaper is intended for use by AWS account administrators as well as users with a requirement to customise their own or other users environments. There is a basic level of both Amazon Web Services (AWS) and OS level knowledge required in order to perform some of the example customisations and configurations described in this whitepaper.

Alces Flight Customiser

Included with each Alces Flight Compute environment is the *Alces Customiser* tool - which enables simple, repeatable configuration of your Alces Flight Compute environments by utilising your available Simple Storage Service (S3) buckets. The *Alces Customiser* tool allows you to place configuration scripts in your S3 bucket which are picked up at cluster launch time and run. Example customisation scripts may include;

- Configuring external storage mounts
- Installing packages via `yum`
- Installing commercial applications

The *Alces Customiser* tool requires a correctly configured bucket, for information on correctly configuring an S3 bucket for use with the *Alces Customiser* tool - please visit the documentation:

http://docs.alces-flight.com/en/stable/customisation/customisation.html

The *Alces Customiser* tool must be correctly set up and configured prior to deploying an environment in order to correctly perform customisation. When deploying your Alces Flight Compute environment from the AWS Marketplace - if a custom S3 bucket is used, it is important to note the custom bucket name when entering your CloudFormation parameters. For more information on answering the CloudFormation questions, please visit the documentation:

http://docs.alces-flight.com/en/stable/launch-aws/launching_on_aws.html#how-to-answer-cloudformation-questions

Example usage

The following section details an example usage of the Alces customiser tool to install the `emacs` command-line text editor using a simple shell script. From your local workstation, create a new shell script, such as `emacs.sh` - and enter the following contents:

```
#!/bin/bash
yum -y install emacs
```

Using either the S3 web console or `s3cmd` from your local workstation, place the emacs install script into your S3 bucket, in the `configure.d` folder. This can be done with the s3cmd utility using the following example command:

```
$ s3cmd put emacs.sh s3://vlj/customizer/default/configure.d/
emacs.sh
upload: 'emacs.sh' -> 's3://vlj/customizer/default/configure.d/
emacs.sh'  [1 of 1]
 33 of 33   100% in    0s    35.23 B/s  done
```

When deploying your next Alces Flight Compute environment - assuming the appropriate Customiser bucket is selected, the emacs install script will be run. To confirm success, log in to your environment and check that emacs has been installed:

```
[alces@login1(vaughan) ~]$ which emacs
/usr/bin/emacs
```

Using the *Alces Customiser* tool - many available configuration and customisation options are possible. Examples of some of the available customisations are available later on in this whitepaper.

Resource customisation

The Alces Flight Compute AWS Marketplace CloudFormation template creates a general purpose HPC cluster - with many default, generic options for any created resources such as compute instances and attached storage. Whilst these general purpose options will work well for many workloads (as well as being suitable for development and testing purposes) - you may wish to modify the resource creation using your own custom CloudFormation template(s) based on the latest available Alces Flight Compute AWS Marketplace CloudFormation template. The following section will detail how to modify some of the default options to use some of the available AWS resource options, including;

- Adding, removing or modifying instance types for login and compute instances
- Modifying login and compute EBS disk type to use alternative types
- Modifying compute host EBS disk to increase or decrease the size per compute host

This section of the whitepaper assumes a reasonable knowledge of AWS CloudFormation in order to perform some of the customisations. If you have not used CloudFormation before, we suggest reading the available documentation:

https://aws.amazon.com/cloudformation/

How to obtain the latest CloudFormation template

The Alces Flight Compute CloudFormation template is easily obtained from the Amazon Marketplace, which will be used to perform customisations. The following steps will help you locate the CloudFormation template:

1. Locate the Alces Flight Compute AWS Marketplace page
2. Follow the steps to begin launching the CloudFormation stack
3. When you reach the initial CloudFormation page ("**Select template**"), copy the S3 URL located in the **Specify an Amazon S3 template URL** and download to your local workstation

Customising instance types

The Alces Flight Compute CloudFormation template includes a default set of popular instance types for different types of workload including memory optimised and compute optimised instance types. You may wish to use additionally available instance types that are not listed in the Alces Flight Compute CloudFormation template - this section will detail how to customise the available instances for both login and compute instances.

Customising login node instance types

The CloudFormation templates includes a selection of mappings, which help create useful identifiers for selecting instance types when choosing the CloudFormation options. The instance type mappings can be found in the **Mappings** section of the CloudFormation template. A section of the instance type mapping block looks like the following:

```
"FlightTypeToInstanceType": {
  "compute-2C-3GB.small-c4.large": { "InstanceType": "c4.large" },
  "compute-8C-15GB.medium-c4.2xlarge": { "InstanceType": "c4.2xlarge" },
  "compute-16C-30GB.large-c4.4xlarge": { "InstanceType": "c4.4xlarge" },
```

In order to add custom instance types, you must know the instance type name, for example to add a storage-dense instance type to the available instance type selections, you could add the following line to the `FlightTypeToInstanceType` mappings block:

```
"storage-2C-6TB.small-d2.xlarge": { "InstanceType": "d2.xlarge" }
```

Once the custom instance type has been added, you can then add it to the login node instance type selection by modifying the `LoginType` values in the **Parameters** section. This can be done by adding another line to the AllowedValues section, resulting in the following example parameter block:

```
"LoginType": {
      "Description": "Select the login node instance type to deploy - this defines
the number of cores and memory available.",
      "Type": "String",
      "Default": "medium-r3.2xlarge",
      "AllowedValues": [
          "small-t2.large",
          "medium-r3.2xlarge",
          "large-c4.8xlarge",
          "disk.small-d2.xlarge",
       ]
  },
```

Customising compute node instance types

Much like adding additional login node instance types, compute node instance types can be added using the `FlightTypeToInstanceType` mappings block together with the `ComputeType` parameter block.

Customising instance disk types

Each instance deployed into an Alces Flight Compute environment uses Elastic Block Storage (EBS) backing for its root volume. The EBS type used is a general purpose, slow but cost-effective disk type used for general purpose workloads. The EBS **magnetic** disk type is used for both the login node and compute node disk type. You may wish to change to another EBS disk type, such as one of the available SSD volume types. The following section details how to change the disk type for both login and compute hosts in the Alces Flight Compute AWS Marketplace CloudFormation template. For information on the available EBS volume types, and the characteristics of each volume type - please visit the documentation:

https://aws.amazon.com/ebs/

Login node disk type

You can modify the login node EBS root disk by locating the `FlightLogin` resource block within the CloudFormation template. Within the `FlightLogin` resource block, you can modify the `BlockDeviceMappings` option which contains the root disk setting:

```
"BlockDeviceMappings": [
    {
        "DeviceName": "/dev/sda1",
        "Ebs": {
            "VolumeSize": { "Ref": "LoginSystemDiskSize" }
        }
    },
```

No EBS volume type is specified, so the volume type defaults to **magnetic**. To choose one of the available EBS types, add another option to the list of parameters to set the volume type such as the following:

```
"VolumeType": "io1"
```

Your CloudFormation template block should now look like the following:

```
"BlockDeviceMappings": [
    {
        "DeviceName": "/dev/sda1",
        "Ebs": {
            "VolumeSize": { "Ref": "LoginSystemDiskSize" },
            "VolumeType": "io1"
        }
    },
```

Compute node disk type

Much like modifying the login node disk type, the compute node EBS disk type can be modified by editing the `BlockDeviceMappings` block within the `ComputeConfig` resource.

Instance customisation

Each Alces Flight Compute environment is deployed with a default configuration - you may wish to customise the deployed configuration for each cluster you deploy. Using the previously mentioned Alces customiser tool - it is simple to make configuration changes to each of the future environments you deploy. The following section assumes that the required set up for using the Alces customiser tool has been performed.

Setting up yum packages and repositories

The Alces customiser tool is particularly useful for interacting with the distributions package manager, for example `yum`. Using the Alces customiser tool - it is simple to add yum repositories and install packages from the available repositories when an environment launches.

Setting up yum repositories

Using the Alces customiser tool - you can easily add yum repositories to each of the nodes in your compute environment through the following example script - which adds an example yum repository which you or another organisation may look after. Yum repository configuration files must be correctly formatted in order to be picked up by the yum tool. You can also use the yum repository set up script to install packages available through the newly configured repository.

```
#!/bin/bash
cat << EOF > /etc/yum.repos.d/alces.repo
[alces-example]
name=Example repository
baseurl=http://repo.domain.com/CentOS/7/os/x86_64/
enabled=1
gpgcheck=1
gpgkey=http://repo.domain.com/CentOS/7/os/x86_64/RPM-GPG-KEY-CentOS-7
EOF
```

Your yum repository installation script should be placed in the appropriate Alces customiser folder.

Once your environment has deployed - you can check the availability and successful installation of your new yum repositories:

```
[alces@login1(vlj) ~]$ yum repolist
Loaded plugins: fastestmirror
alces  alces/primary
Determining fastest mirrors
 * base: ftp.heanet.ie
 * extras: ftp.heanet.ie
 * updates: ftp.heanet.ie
base/7/x86_64              CentOS-7 - Base                                 9,007
extras/7/x86_64            CentOS-7 - Extras                                 310
updates/7/x86_64           CentOS-7 - Updates                              1,687
repolist: 11,005
```

Setting up storage mounts

You may also have storage servers that you wish to mount on the nodes in your Alces Flight Compute environments for access to your data. This can be done using the Alces customiser tool - assuming appropriate set up has been made to allow access from your Alces Flight Compute environment.

The following example storage configuration script can be used in order to set up an external storage mount and optionally mount it:

```
#!/bin/bash
mkdir /mnt/data
chmod 777 /mnt/data
cat << EOF >> /etc/fstab
52.81.92.31:/data /mnt/data nfs defaults
EOF
mount -a
```

Installing Gridware applications

The Alces customiser tool can also be used to pre-install Alces Gridware packages and depots, allowing you to start your science even faster. The following example configuration script can be used to install the benchmark Gridware depot as well as an additional single application:

```
#!/bin/bash -l
NODEROLE=$2
if [ "$NODEROLE" == "master" ];
then
    ALCESBIN="/opt/clusterware/bin/alces"
    ${ALCESBIN} gridware depot install benchmark
    ${ALCESBIN} gridware install apps/python/2.7.8
fi
```

Your chosen applications will then be installed each time an Alces Flight Compute environment is deployed - you can see the output of the installations in the `/var/log/clusterware/instance.log` file, for example:

```
Jun 15 13:52:10 [cluster-customizer:configure] Running configure hook: /opt/clusterware/
var/lib/customizer/defa
ult/configure.d/gridware.sh
Jun 15 13:52:10 [cluster-customizer:configure] Installing depot: benchmark
Jun 15 13:52:10 [cluster-customizer:configure]
Jun 15 13:52:10 [cluster-customizer:configure]  > Initializing depot: benchmark
Jun 15 13:52:10 [cluster-customizer:configure]       Initialize ... OK
Jun 15 13:52:10 [cluster-customizer:configure] module use /opt/gridware/benchmark/
$cw_DIST/etc/modules
Jun 15 13:52:10 [cluster-customizer:configure]
Jun 15 13:52:10 [cluster-customizer:configure] Importing mpi-openmpi-1.8.5-el7.tar.gz
```

Image customisation

The Alces Flight Compute Amazon Machine Image (AMI) shipped through the AWS Marketplace is a general purpose image aimed at getting users and researchers started with their compute clusters. Whilst there are customisation tools available, you may wish to create a custom Alces Flight Compute image to meet needs such as security updates and patches as well as the pre-configuration and customisation of certain things for use by your fellow researchers.

Creating your own Alces Flight Compute AMI's enables you to have multiple copies for different purposes - for example you may wish to create AMI's with popular applications for a specific Science field pre-installed for a particular group of researchers such as bioinformatics or chemistry.

At the time of shipping - the latest available OS updates are applied to the Alces Flight Compute AMI, however as releases are not regular - you may wish to have your own Alces Flight Compute AMI's that contain the latest available security updates and patches.

In order to create your own custom Alces Flight Compute AMI - you must first have the appropriate read permission for one of the available AMI's. Once you have permission, you can use the AWS command-line tools to copy the AMI from its source destination into your own AWS account. The copy can be performed with the following example command:

```
aws ec2 copy-image \
      --source-region eu-west-1 \
      --source-image-id ami-123456 \
      --name flight-compute-custom
```

Once the AMI copy process has completed - you can begin performing your required customisations. The basic process for customising is as follows;

1. Launch an instance using the previously copied AMI with on-demand reservation type
2. Log into the newly launched instance and perform the customisations you wish to make such as distribution package installations or Alces Gridware package installations etc.
3. Optionally clean the instance ready for AMI creation - such as clearing bash history, temporary directories and log files
4. Either from the AWS EC2 console or using the AWS command-line tools, create an image from the running instance. Note - the process can take several minutes to complete.
5. Once the AMI creation process is complete - you can use your new customised Alces Flight Compute AMI to deploy Alces Flight Compute HPC clusters.

Customising your environment

This whitepaper has detailed some of the ways in which you can customise your Alces Flight Compute environment including package installation and customisation as well as cluster resource creation modification when deploying via AWS CloudFormation. Using the basic examples provided, you should now be able to perform basic customisation of your environments as well as custom Alces Flight Compute AMIs for use with your own CloudFormation templates and HPC clusters.

You may also wish to configure your own Alces Flight Compute images to use configuration management utilities such as Chef or Puppet in order to manage and configure your Alces Flight Compute environments - granting you even greater control.

For more information on some of the topics covered in this whitepaper, please refer to the available documentation:

http://docs.alces-flight.com/en/stable/customisation/customisation.html
https://aws.amazon.com/cloudformation/