AWS Spot Market with Alces Flight Compute

How to interact and take advantage of the Spot market

Alces Flight Compute - 2016.2

Vaughan Jones - Alces Flight Ltd
May 2016

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

CONTENTS

THIS MATERIAL IS DESIGNED TO ASSIST CAPABLE USERS TO CUSTOMISE OUR SOFTWARE PRODUCTS, EXTENDING THEM TO PERFORM OPERATIONS BEYOND ORIGINAL DESIGN PARAMETERS. EXAMPLES OF POSSIBLE EXTENSIONS ARE PROVIDED TO HELP CUSTOMERS REALISE FUTURE POTENTIAL OF THE SOFTWARE AND ARE PROVIDED WITHOUT WARRANTY AND ARE NOT SUPPORTED OR GUARANTEED BY ALCES FLIGHT LTD OR OUR LOCAL SOFTWARE RESELLERS. THIS DOCUMENTATION HAS BEEN CREATED TO INFORM AND SHARE KNOWLEDGE WITH CUSTOMERS FOR REFERENCE PURPOSES ONLY; SOFTWARE FEATURES AND PERFORMANCE ARE NOT GUARANTEED. THIS DOCUMENTATION IS NOT DESIGNED AS A STAND-ALONE TRAINING TOOL – EXAMPLE COMMANDS AND SYNTAX ARE INTENDED TO DEMONSTRATE FUNCTIONALITY IN A TRAINING ENVIRONMENT AND MAY CAUSE DATA LOSS IF EXECUTED ON LIVE SYSTEMS. REMEMBER: ALWAYS TAKE BACKUPS OF ANY VALUABLE DATA. THIS DOCUMENTATION IS PROVIDED “AS IS” AND WITHOUT ANY WARRANTY; WITHOUT EVEN THE IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. NO COMPUTER SYSTEM CAN BE COMPLETELY SECURE - FOLLOW GOOD SECURITY ADVICE AND ALWAYS HAVE YOUR COMPUTER SYSTEMS CHECKED BY COMPETENT SECURITY PROFESSIONALS BEFORE BEING USED WITH LIVE DATA. PLEASE SEE THE EULA INCLUDED WITH THE PROVIDED SOFTWARE PACKAGES FOR FULL USAGE TERMS AND CONDITIONS. WE WELCOME CUSTOMER FEEDBACK AND SUGGESTIONS FOR FUTURE ENHANCEMENTS - PLEASE VISIT THE COMMUNITY SUPPORT SITE AT WWW.ALCES-FLIGHT.COM.

FOR MORE INFORMATION ON ALCES FLIGHT, PLEASE VISIT: HTTP://WWW.ALCES-FLIGHT.COM/. PLEASE SUPPORT SOFTWARE DEVELOPERS WHEREVER THEY WORK – IF YOU USE OPEN-SOURCE SOFTWARE, PLEASE CONSIDER CONTRIBUTING TO THE MAINTAINING ORGANISATION OR PROJECT, AND CREDITING THEIR WORK AS PART OF YOUR RESEARCH AND PUBLICATIONS.

Overview

The following paper is aimed at those wishing to learn more about the AWS Spot Market - and how an Alces Flight Compute environment is able to interact with the Spot market to reduce your total cost of ownership.

Alces Flight

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

The Alces Flight range of 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.

AWS Spot Market

The AWS Spot Market provides access to the AWS Cloud spare capacity, often at a significant discount over on-demand instance costs. AWS Spot instances behave in the same manner as any other AWS instance type, allowing you to create them using your own AMIs and configuration details.

Using Spot instances allows you to set the price you are prepared to pay per hour per instance - allowing you to easily forecast your spend.

Spot instances are ideal for research and high performance computing (HPC), particularly where cluster load fluctuates and compute jobs are able to checkpoint and re-run.

Alces Flight Compute on AWS Spot

Alces Flight Compute environments are able to utilise the AWS Spot Market, using Spot instances as dedicated compute hosts - automatically adding and removing new and killed Spot instances to the environment in a dynamic nature. Spot instances should typically be used for compute jobs that run for less than 24 hours - however in some cases Spot instances may run for many days before being killed.

The Spot market allows you to significantly reduce your total operating and research costs when using the AWS Cloud for non mission-critical tasks. For mission-critical tasks - on-demand compute instances should be utilised to avoid potential failure.

Spot can be utilised in a number of ways that greatly benefit researchers and organisations, gaining access to a large pool of resource that may not be affordable using on-premise resource. Access to tens of thousands of compute cores is instantly available through the Spot market - allowing you to dynamically scale your Alces Flight Compute environment as and when the compute capacity is required using the spare capacity AWS has to offer on the Spot market.

All users of the AWS Spot market pay the same price for each billing hour in a certain period - even if the maximum bid entered by an individual user is far higher than the current bid price. For example, you can launch 1 Spot instance using the `c4.8xlarge` instance type with a current bid price of $0.34 USD per hour, and enter a maximum bid of $0.50 USD per hour - for the first hour you will only pay $0.34 USD. Within the first hour, AWS may reduce the bid price of a `c4.8xlarge` instance type to $0.29 USD, at which point you will only pay $0.29 USD for the next billing hour. In the second hour, AWS may increase the current bid price to $0.51 USD per hour - at which point your Spot instance will be cancelled as the current bid price exceeds your maximum bid price. The current bid price is controlled by AWS in response to the demand for on-demand, reserved and dedicated instance types.

When to use Spot?

Spot instances should be used for particular types of workloads and applications, as well as considering the importance of job completion within a timeframe. Spot instances should be considered ephemeral and volatile - and should not be used for time and mission critical workloads or applications. Many organisations and researchers can benefit from using Spot instances for their workloads, typical use-cases include;

- Scientific research computing; non-critical research compute workloads can be run on Spot instances - if a Spot instance is killed whilst running, the job can simply be re-run on a different Alces Flight compute host
- Embarrassingly parallel compute jobs; a compute job split into multiple parallel tasks is well suited to the Spot market
- Testing; Spot instances can be used to test job scripts and applications before running a production workload without incurring the cost of running a cluster using on-demand instances

Mission-critical workloads can also be run using the Spot instance market when combined with additional application and job script features such as checkpointing, allowing your jobs to restart from a certain point if killed whilst running on a Spot instance. Utilising these features can significantly reduce your total operating cost of compute jobs whilst ensuring compute jobs reach completion.

AWS Spot instances also issue termination notices, which notifies the instance two minutes prior to termination - this enables you to add features to your job scripts and workloads which identify potential compute node termination, giving the application or job time to checkpoint or write out result data before quitting and the node terminating.

How Alces Flight works with Spot

When creating Spot instance(s), the process below is followed;

1. AWS EC2 sets the bid price for each instance type
2. Your instance(s) are launched, with your maximum bid price set. You may enter a much higher value than the current Spot bid price, but you will never pay over the current bid price set by AWS
    A. If the Flight Compute AMI is manually launched through the EC2 console, the maximum bid price is manually entered for each set of instances launched
    B. If launched through the AWS AutoScaling interface - the bid is pre-configured as part of the AutoScaling Launch Configuration
3. If both available capacity and bid values are met - your Spot instances will be launched
4. Spot instances run until either;
    A. Capacity limits are reached and instances are killed
    B. Maximum bid price is exceeded by the current global Spot bid price for your region

Alces Flight Compute environments automatically work with the AWS Spot market - dynamically and automatically creating and destroying compute instances as the cluster requires capacity. Alces Flight Compute environments operate with the Spot market in the following way;

1. If no jobs are currently waiting in the scheduler queue, compute nodes will be checked for shutdown eligibility, when the following criteria is met;
    A. Node is within 5 minutes of its next billing hour
    B. No compute jobs are currently active on the compute node
2. Once jobs begin queueing in the cluster scheduler queue, the Alces Flight Compute environment will determine the number of compute hosts required to fulfil the currently queued workload
3. The above processes continue to dynamically scale up and down the number of compute hosts in your Alces Flight Compute environment to meet the scheduler demand

AWS CloudWatch

An Alces Flight Compute environment also interacts with the AWS CloudWatch service in order to appropriately scale up the compute hosts within your cluster. AWS CloudWatch provides a method of accessing some of the built-in metric monitoring included with each AWS instance including CPU usage, disk usage, network usage and much more. Custom metrics can also be used - an Alces Flight Compute environment pushes a custom metric to record the cluster scheduler queue status.

Using a combination of default and custom metrics, alarms can be created for different conditions, along with thresholds for each alarm. Using these alarms allow you to interact with the AWS AutoScaling group feature and associated AutoScaling Scaling Policies, thus controlling your Spot instances.

A typical AutoScaling/CloudWatch setup would perform the following tasks;

1. Alarms monitor metric for a single instance or average of each instance in an AutoScaling group, such as CPU utilisation
2. Alarms check the threshold every 5 minutes, or an otherwise defined interval
    C. If the metric figure is above the threshold, the defined AutoScaling policy will be triggered
      - The AutoScaling policy attached will increase the number of Spot instances in your AutoScaling group by the defined amount

    D. If the metric figure is below the threshold, the monitoring will continue with no action taken

An Alces Flight Compute environment includes pre-configured AWS CloudWatch alarms and AutoScaling policies which are used to automatically scale your Spot instances based on scheduler demand.

AWS Marketplace CloudFormation templates

The Alces Flight Compute AWS Marketplace CloudFormation templates include several important configurations in order to work with the AWS Spot market. The following snippets are taken from the AWS Marketplace templates for Alces Flight Compute release 2016.2, and detail what each resource and configuration provides - these snippets can be reused for your own CloudFormation templates to work with the Spot market.

The template shown in the appendix of this paper is the complete version of the Alces Flight Compute AWS MarketPlace CloudFormation version 2016.2.

Parameters section

The following snippets are taken from the Parameters section of the CloudFormation template, and used in order to appropriately work with the Spot market:

```
        "ComputeSpotPrice": {
            "Description": "Enter your maximum bid per hour for each compute instance.
View the Spot Request calculator for information on spot pricing. (Enter 0 for on-
demand)",
            "Type": "Number",
            "Default": "0"
        },
```

The above snippet offers the user a choice of deploying Spot instances or using on-demand instance types. By specifying a number above 0 - Spot bids are placed, if the current Spot bid is below your input max bid, Spot instances will start to launch.

Conditions section

The following snippets are taken from the Conditions section of the CloudFormation template, and used in order to appropriately work with the Spot market:

```
        "UseSpotCompute": {
            "Fn::Not": [
                {
                    "Fn::Equals": [
                        {
                            "Ref": "ComputeSpotPrice"
                        },
                        "0"
                    ]
                }
            ]
        },
```

The above snippet works with the previously listed ComputeSpotPrice parameter - if a Spot bid price greater than 0 is entered, the CloudFormation template will proceed to launch any requested compute instances as Spot instances.

Resources section

The following snippets are taken from the Resources section of the CloudFormation template, and used in order to appropriately work with the Sot market:

```
        "ComputeConfig": {
            "Type": "AWS::AutoScaling::LaunchConfiguration",
            "Properties": {
                "AssociatePublicIpAddress": "True",
                "KeyName": {
                    "Ref": "KeyPair"
                },
                "ImageId": {
                    "Fn::FindInMap" : [ "AWSRegionArch2AMI", { "Ref" : "AWS::Region" },
{ "Fn::FindInMap" : [ "AWSInstanceType2Arch", { "Fn::FindInMap":
[ "FlightTypeToInstanceType", { "Ref": "ComputeType" }, "InstanceType" ] }, "Arch" ] } ]
                },
             "IamInstanceProfile": {
                    "Fn::If": [
                        "UseAlarms",
                        {
                            "Ref": "ComputeIAMProfile"
                        },
                        {
                            "Ref": "AWS::NoValue"
                        }
                    ]
                },
                "InstanceType": {
                     "Fn::FindInMap": [ "FlightTypeToInstanceType", { "Ref":
"ComputeType" }, "InstanceType" ]
                },
                "BlockDeviceMappings": [
                    {
                        "DeviceName": "/dev/sda1",
                        "Ebs": {
                            "VolumeSize": "20"
                        }
                    }
                ],
                "SpotPrice": {
                    "Fn::If": [
                        "UseSpotCompute",
                        {
                            "Ref": "ComputeSpotPrice"
                        },
                        {
                            "Ref": "AWS::NoValue"
                        }
                    ]
                },
                "SecurityGroups": [
                    {
                        "Ref": "FlightSG"
                    }
                ],
                "UserData": {
                    "Fn::Base64": {
                        "Fn::Join": [
                            "",
                            [
                                "#cloud-config\n",
                                "system_info:\n",
                                "  default_user:\n",
                                "    name: ", { "Ref": "Username" }, "\n",
                                "write_files:\n",
                                "- content: |\n",
                                "    cluster:\n",
                                "      uuid: '11111111-2222-3333-444444444444'\n",
                                "      token: '1A0a1aaAA1aAAA/aaa1aAA=='\n",
                                "      name: ", { "Ref": "AWS::StackName" }, "\n",
                                "      role: 'slave'\n",
                                "      master: ", { "Fn::GetAtt": [ "FlightLogin",
"PrivateIp" ] }, "\n",
                                "      tags:\n",
                                "        scheduler_roles: ':compute:'\n",
                                "  owner: root:root\n",
                                "  path: /opt/clusterware/etc/config.yml\n",
                                "  permissions: '0640'\n"
                            ]
                        ]
                    }
                }
            }
        },
```

The above snippet shows the AWS AutoScaling Group Launch Configuration - this is a pre-defined set of requirements for instances launched into the AWS AutoScaling group which includes the configuration required of each launched compute host.

Included within the Launch Configuration is the AWS Spot price bid - this marks the Launch Configuration to use either on-demand or spot instances based on the choice previously made in the Parameters section:

```
                "SpotPrice": {
                    "Fn::If": [
                        "UseSpotCompute",
                        {
                            "Ref": "ComputeSpotPrice"
                        },
                        {
                            "Ref": "AWS::NoValue"
                        }
                    ]
                },
```

The following snippet details the AWS AutoScaling Group:

```
        "FlightCompute": {
              "Type": "AWS::AutoScaling::AutoScalingGroup",
              "DependsOn": "FlightLogin",
              "Properties": {
                  "DesiredCapacity": {
                    "Fn::If": [
                        "UseAlarms",
                        {
                            "Ref": "InitialNodes"
                        },
                        "8"
                    ]
                  },
                  "LaunchConfigurationName": {
                      "Ref": "ComputeConfig"
                  },
                  "PlacementGroup": {
                      "Ref": "PlacementGroup"
                  },
                  "Tags": [
                      {
                          "Key": "Name",
                          "Value": {
                            "Fn::Join": [
                                "",
                                [
                                    {
                                        "Ref": "AWS::StackName"
                                    },
                                    "-compute"
                                ]
                            ]
                        },
                        "PropagateAtLaunch": "true"
                      }
                  ],
                  "MinSize": "1",
                  "MaxSize": "8",
                  "VPCZoneIdentifier": [
                    {
                        "Ref": "FlightSubnet"
                    }
                  ]
              }
        },
```

The above snippet details the AWS AutoScaling Group - this works together with the previously shown AWS AutoScaling Launch Configuration - creating as few or as many Spot compute instances as your workload requires.

Using AWS Spot instances

Using Spot with Alces Flight Compute

When deploying your Alces Flight Compute environment from AWS Marketplace (version 2016.2), an option is provided to use Spot instances or On-demand instances. When deploying your environment - enter your desired Spot bid price in the `ComputeSpotPrice` parameter box to enable Spot compute instances. Entering a value of `0` will disable Spot instances and deploy Alces Flight compute nodes as on-demand instance types.

Once your Alces Flight Compute environment has launched, it will automatically interact with the Spot instance market - dynamically launching and destroying instances as your workload increases and decreases using a number of variables to determine whether additional or a decreased compute node capacity is required.

Using Spot instances outside of Alces Flight Compute

There are several methods of deploying Spot instances for your own purposes and needs outside of an Alces Flight Compute environment - the most common methods of launching Spot instances are;

- **Launching through the EC2 Management Console**; when requesting an instance through the EC2 Management Console, you can request Spot instances as part of the creation process
- **Launching through the AWS CLI tool;** Spot instances can be requested through the AWS CLI tool using the `aws ec2 request-spot-instances <options>` command
- **Launching with an AutoScaling group;** together with an AutoScaling Launch Configuration and AutoScaling Group - Spot compute instances can be easily created, controlled and managed through multiple interfaces including EC2 Management Console and command-line tools
- **Alces Flight Compute AWS MarketPlace CloudFormation templates;** Through the Alces Flight Compute AWS MarketPlace CloudFormation templates; the Alces Flight Compute AWS MarketPlace CloudFormation templates include the configuration shown above to automatically create Spot compute instances

Planning for failure

Spot instances come with the risk of failure, to prevent manual intervention there are actions you may wish to perform to ensure a Spot instance failure does not affect your workload.

Many cluster schedulers support job restarting or re-running on failure - the Alces Flight Compute 2016.2 release includes the SGE scheduler which supports job restarting.

The following example job script includes the job restart flag, informing the scheduler to re-run the job if it is marked as failed:

```
#!/bin/bash -l
#$ -N testjob
#$ -o $HOME/testjob.out.$JOB_ID
#$ -r yes
sleep 10000
```

The `-r yes flag` informs the scheduler to re-run the job on failure. Note - if a Spot instance is killed whilst a job is running, unless saved to persistent storage - all data will be lost.

Once the compute host running the job is killed - the scheduler will automatically find a suitable replacement host and re-run the job from the beginning:

```
[alces@login1(vaughans-cluster) ~]$ qsub spot-restart.sh
Your job 1 ("testjob") has been submitted
[alces@login1(vaughans-cluster) ~]$ qstat
job-ID  prior   name       user         state submit/start at     queue
----------------------------------------------------------------------------------------------------
1 11.02734 testjob    alces        r     05/10/2016 15:22:00 byslot.q@ip-10-75-0-99.eu-wes
[alces@login1(vaughans-cluster) ~]$ pdsh -w ip-10-75-0-99 'sudo shutdown -h now'
ip-10-75-0-99: Warning: Permanently added 'ip-10-75-0-99,10.75.0.99' (ECDSA) to the list of known
hosts.
ip-10-75-0-99: Connection to ip-10-75-0-99 closed by remote host.
pdsh@login1: ip-10-75-0-99: ssh exited with exit code 255
[alces@login1(vaughans-cluster) ~]$ qstat
job-ID  prior   name       user         state submit/start at     queue
-----------------------------------------------------------------------------------------------------
1 11.02734 testjob    alces        Rq    05/10/2016 15:22:00
[alces@login1(vaughans-cluster) ~]$ qstat
job-ID  prior   name       user         state submit/start at     queue
---------------------------------------------------------------------------------------------------
1 11.02734 testjob    alces        Rr    05/10/2016 15:28:00 byslot.q@ip-10-75-0-98.eu-west
```

The above output shows:

- Submitting the example job script to the cluster scheduler
- Manually killing the compute host to simulate a Spot instance being removed
- Cluster scheduler re-running the compute job on a different compute host (marked `Rr`)

Cost-savings

The primary purpose of utilising the AWS Spot market is often the cost savings over on-demand and reserved instance types for low-priority and re-runnable compute jobs. The following tables show the potential cost savings of different sizes of compute clusters for different lengths of time when utilising Spot instances. Note - the prices shown below are accurate as of the time of publication (May 2016).

72 compute core cluster - c4.8xlarge instance type

Number of Spot instances	Duration in wall-time run for	Spot instance estimated cost (USD)	On-demand instance estimated cost (USD)	Savings over on-demand
2	1 hour	$0.68	$3.81	82%
	1 day	$16.32	$91.49	
	7 days	$114.24	$640.42	
	1 month	$496.40	$2,782.76	

324 compute core cluster - c4.8xlarge instance type

Number of Spot instances	Duration in wall-time run for	Spot instance estimated cost (USD)	On-demand instance estimated cost (USD)	Savings over on-demand
9	1 hour	$3.06	$17.15	82%
	1 day	$73.44	$411.70	
	7 days	$514.08	$2,881.87	
	1 month	$2,233.80	$12,522.42	

1,008 compute core cluster - c4.8xlarge instance type

Number of Spot instances	Duration in wall-time run for	Spot instance estimated cost (USD)	On-demand instance estimated cost (USD)	Savings over on-demand
28	1 hour	$9.52	$53.37	82%
	1 day	$228.48	$1,280.83	
	7 days	$1,599.36	$8,965.82	
	1 month	$6,949.60	$38,958.64	

Many other AWS instance types are available through the AWS Spot Market - allowing you to choose instance types best suited to your workload whilst benefiting from the reduced operating costs of the Spot market. Each of the Alces Flight Compute AWS Marketplace templates include several Spot compute instance type options.

The cost savings over on-demand instance types are significant in all cases and scenarios - and further cost-saving methods are available using additional features of the AWS Spot market.

Block duration Spot instances

An additional feature of the AWS Spot Market is the ability to launch Spot instances for a fixed duration. For research and high performance computing workloads this is often achievable, as many compute jobs will run for approximately the same time - each time.

The block duration feature allows you to specify a fixed length of time up to a maximum of 6 hours for a single or multiple Spot instance(s). Block duration instances benefit from only being billed for a fixed hourly price that remains whilst the Spot instance runs. This can help to avoid Spot market price fluctuations and accurately forecast operating costs of compute tasks.

Block duration Spot instances also benefit from running without interruption or termination for the length of their block request - unlike regular Spot instances which are subject to interruption or termination at any point in their lifecycle.

Block duration Spot instances do not receive as greater discount over on-demand prices as regular Spot instances - but still often receive a 30-45% discount, plus an additional 5% during quiet hours.

Most popular cluster schedulers/task managers will detail the wall clock time a compute job took to complete, you should use the worst-case scenario when choosing your block duration time - and include an extra 5 minutes each side to allow for node configuration and shutdown.

Block duration Spot vs On-demand

The following tables provide comparisons of purchasing different sizes of compute clusters using Block Duration Spot and On-demand instance prices - showing the savings available for each compute cluster size.

72 core cluster - c4.8xlarge instance type

Number of Spot Instances	Duration in wall-time run for (hours)	Block duration Spot instance estimated cost (USD)	On-demand instance estimated cost (USD)	Block duration Spot instance savings over on-demand
2	1	$1.94	$3.81	49%
	6	$14.81	$22.87	35%

324 core cluster - c4.8xlarge instance type

Number of Spot Instances	Duration in wall-time run for (hours)	Block duration Spot instance estimated cost (USD)	On-demand instance estimated cost (USD)	Block duration Spot instance savings over on-demand
9	1	$8.73	$17.15	49%
	6	$66.64	$102.92	35%

1,008 core cluster - c4.8xlarge instance type

Number of Spot instances	Duration in wall-time run for (hours)	Block duration Spot instance estimated cost (USD)	On-demand instance estimated cost (USD)	Block duration Spot instance savings over on-demand
28	1	$27.16	$53.37	49%
	6	$207.31	$320.21	35%

Whilst Block Duration Spot instances receive a lower discount than regular Spot instances - the benefits of instance protection may greatly out-weigh the increased cost per instance. By knowing a compute job will take 2 hours to run, you can plan and schedule your Spot instances accordingly - knowing a fixed price is set for the duration of the Spot instances lifetime.

Is Spot right for me?

Many workloads and applications can benefit from the advantages of Spot - which in turn greatly reduce your overall operating cost. Due to the flexibility of the AWS Cloud and the Spot market - compute nodes are no longer required to sit around idle waiting for jobs to arrive. Alces Flight Compute together with the Spot market allow for a dynamically scaling, efficient environment based on your workload and operating cost requirements, by entering maximum bid prices per instance it is easy to cap your spend for your compute cluster.

For more information on the AWS Spot Market - please visit the AWS documentation and Spot pricing history pages:

https://aws.amazon.com/ec2/spot/
https://eu-west-1.console.aws.amazon.com/ec2sp/v1/spot/launch-wizard?region=eu-west-1
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-spot-instances-history.html
http://docs.alces-flight.com/en/latest/

Appendix

Alces Flight Compute AWS MarketPlace CloudFormation template

```
{
    "Description": "Launch an Alces HPC environment with a single SGE master node
together with 8 initial compute nodes using EC2 spot.",
    "Parameters": {
        "KeyPair": {
            "Description": "Choose an existing AWS key for administrator access",
            "Type": "AWS::EC2::KeyPair::KeyName"
        },
        "Username": {
            "Description": "Enter a username - this is used for the cluster administrator
account",
            "Type": "String",
            "Default": "alces",
            "MaxLength": "16",
            "MinLength": "1",
            "AllowedPattern": "[-a-z0-9]*",
            "ConstraintDescription": "Enter a username between 1 and 16 characters using
a-z, 1 to 9 and -"
        },
        "ComputeSpotPrice": {
            "Description": "Enter your maximum bid per hour for each compute instance.
View the Spot Request calculator for information on spot pricing. (Enter 0 for on-
demand)",
            "Type": "Number",
            "Default": "0"
        },
        "ComputeType": {
            "Description": "Select the compute node instance type to deploy - this
defines the number of cores and memory available",
            "Type": "String",
            "Default": "compute.small-c4.large",
            "AllowedValues": [
                "compute.small-c4.large",
                "compute.medium-c4.2xlarge",
                "compute.large-c4.4xlarge",
                "compute.dedicated-c4.8xlarge",
                "balanced.small-m3.xlarge",
                "balanced.medium-m3.2xlarge",
                "balanced.large-m4.4xlarge",
                "balanced.dedicated-m4.10xlarge",
                "memory.small-r3.xlarge",
                "memory.medium-r3.2xlarge",
                "memory.large-r3.4xlarge",
                "memory.dedicated-r3.8xlarge",
                "gpu.medium-g2.2xlarge",
                "gpu.dedicated-g2.8xlarge"
            ]
        },
        "InitialNodes": {
            "Description": "Enter how many nodes to start initially. For efficiency, we
recommend starting with a small number and allow autoscaling to add nodes when you have
jobs waiting in the queue. (Enter 0 to disable autoscaling and start all nodes at init)",
            "Type": "Number",
            "Default": "0",
            "MaxValue": "8"
        },
        "LoginSystemDiskSize": {
            "Description": "Enter the size in GB of shared system disk to deploy. This
defines the amount of shared user storage and application storage available",
            "Type": "String",
            "Default": "500"
        },
        "LoginType": {
            "Description": "Select the login node instance type to deploy - this defines
the number of cores and memory available",
            "Type": "String",
            "Default": "small-t2.large",
            "AllowedValues": [
                "small-t2.large",
                "medium-r3.2xlarge",
                "large-c4.8xlarge"
            ]
        },
        "NetworkCIDR": {
            "Description": "Enter a valid CIDR that is permitted to access the
Clusterware master node. Eg 0.0.0.0/0 for access from all locations (insecure)",
            "Type": "String",
            "AllowedPattern": "[0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3}/[0-9]{1,2}",
            "ConstraintDescription": "Please specify a valid CIDR, e.g. 0.0.0.0/0"
        }
    },
    "Conditions": {
        "UseSpotCompute": {
            "Fn::Not": [
                {
                    "Fn::Equals": [
                        {
                            "Ref": "ComputeSpotPrice"
                        },
                        "0"
                    ]
                }
            ]
        },
       "UseAlarms": {
            "Fn::Not": [
                {
                    "Fn::Equals": [
                        {
                            "Ref": "InitialNodes"
                        },
                        "0"
                    ]
                }
            ]
        },
       "LoginUsePlacement": {
           "Fn::Not": [
             {
                 "Fn::Equals": [
                    {
                        "Ref": "LoginType"
                    },
                    "small-t2.large"
                 ]
             }
           ]
       }
    },
    "Mappings": {
        "AWSRegionArch2AMI" : {
            "us-east-1"      : { "HVM64" : "NOT_SUPPORTED" },
            "us-west-2"      : { "HVM64" : "NOT_SUPPORTED" },
            "us-west-1"      : { "HVM64" : "NOT_SUPPORTED" },
            "eu-west-1"      : { "HVM64" : "ami-ca50dbb9" },
            "ap-southeast-1" : { "HVM64" : "NOT_SUPPORTED" },
            "ap-northeast-1" : { "HVM64" : "NOT_SUPPORTED" },
            "ap-southeast-2" : { "HVM64" : "NOT_SUPPORTED" },
            "sa-east-1"      : { "HVM64" : "NOT_SUPPORTED" },
            "cn-north-1"     : { "HVM64" : "NOT_SUPPORTED" },
            "eu-central-1"   : { "HVM64" : "NOT_SUPPORTED" }
        },
        "AWSInstanceType2Arch" : {
            "t2.large"    : { "Arch" : "HVM64" },
            "m2.xlarge"   : { "Arch" : "HVM64" },
            "m3.2xlarge"  : { "Arch" : "HVM64" },
            "m4.4xlarge"  : { "Arch" : "HVM64" },
            "m4.10xlarge" : { "Arch" : "HVM64" },
            "r3.xlarge"   : { "Arch" : "HVM64" },
            "r3.2xlarge"  : { "Arch" : "HVM64" },
            "r3.4xlarge"  : { "Arch" : "HVM64" },
            "r3.8xlarge"  : { "Arch" : "HVM64" },
            "g2.2xlarge"  : { "Arch" : "HVM64" },
            "g2.8xlarge"  : { "Arch" : "HVM64" },
            "c4.8xlarge"  : { "Arch" : "HVM64" },
            "c4.large"    : { "Arch" : "HVM64" },
            "c4.2xlarge"  : { "Arch" : "HVM64" },
            "c4.4xlarge"  : { "Arch" : "HVM64" }
        },
        "FlightTypeToInstanceType": {
          "compute.small-c4.large": { "InstanceType": "c4.large" },
          "compute.medium-c4.2xlarge": { "InstanceType": "c4.2xlarge" },
          "compute.large-c4-4xlarge": { "InstanceType": "c4.4xlarge" },
          "compute.dedicated-c4.8xlarge": { "InstanceType": "c4.8xlarge" },
          "balanced.small-m3.xlarge": { "InstanceType": "m2.xlarge" },
          "balanced.medium-m3.2xlarge": { "InstanceType": "m3.2xlarge" },
          "balanced.large-m4.4xlarge": { "InstanceType": "m4.4xlarge" },
          "balanced.dedicated-m4.10xlarge": { "InstanceType": "m4.10xlarge" },
          "memory.small-r3.xlarge": { "InstanceType": "r3.xlarge" },
          "memory.medium-r3.2xlarge": { "InstanceType": "r3.2xlarge" },
          "memory.large-r3.4xlarge": { "InstanceType": "r3.4xlarge" },
          "memory.dedicated-r3.8xlarge": { "InstanceType": "r3.8xlarge" },
          "gpu.medium-g2.2xlarge": { "InstanceType": "g2.2xlarge" },
          "gpu.dedicated-g2.8xlarge": { "InstanceType": "g2.8xlarge" },
          "small-t2.large": { "InstanceType": "t2.large" },
          "medium-r3.2xlarge": { "InstanceType": "r3.2xlarge" },
          "large-c4.8xlarge": { "InstanceType": "c4.8xlarge" }
        }
    },
    "Resources": {
        "FlightVPC": {
            "Type": "AWS::EC2::VPC",
            "Properties": {
                "Tags": [
                    {
                        "Key": "Name",
                        "Value": {
                            "Ref": "AWS::StackName"
                        }
                    }
                ],
                "CidrBlock": "10.75.0.0/16",
                "EnableDnsSupport": "true",
                "EnableDnsHostnames": "true"
            }
        },
        "FlightRouteTable": {
            "Type": "AWS::EC2::RouteTable",
            "DependsOn": "FlightVPC",
            "Properties": {
                "VpcId": {
                    "Ref": "FlightVPC"
                },
                "Tags": [
                    {
                        "Key": "Application",
                        "Value": {
                            "Ref": "AWS::StackId"
                        }
                    }
                ]
            }
        },
        "FlightSubnet": {
            "Type": "AWS::EC2::Subnet",
            "Properties": {
                "VpcId": {
                    "Ref": "FlightVPC"
                },
                "CidrBlock": "10.75.0.0/24",
                "Tags": [
                    {
                        "Key": "Name",
                        "Value": {
                            "Fn::Join": [
                                "",
                                [
                                    {
                                        "Ref": "AWS::StackName"
                                    },
                                    "-subnet"
                                ]
                            ]
                        }
                    }
                ]
            }
        },
        "FlightGW": {
            "Type": "AWS::EC2::InternetGateway",
            "DependsOn": "FlightVPC"
        },
        "FlightGWAttach": {
            "Type": "AWS::EC2::VPCGatewayAttachment",
            "DependsOn": "FlightVPC",
            "Properties": {
                "VpcId": {
                    "Ref": "FlightVPC"
                },
                "InternetGatewayId": {
                    "Ref": "FlightGW"
                }
            }
        },
        "FlightRoute": {
            "Type": "AWS::EC2::Route",
            "DependsOn": "FlightGWAttach",
            "Properties": {
                "RouteTableId": {
                    "Ref": "FlightRouteTable"
                },
                "DestinationCidrBlock": "0.0.0.0/0",
                "GatewayId": {
                    "Ref": "FlightGW"
                }
            }
        },
        "SubnetToRouteTable": {
            "Type": "AWS::EC2::SubnetRouteTableAssociation",
            "DependsOn": "FlightRouteTable",
            "Properties": {
                "SubnetId": {
                    "Ref": "FlightSubnet"
                },
                "RouteTableId": {
                    "Ref": "FlightRouteTable"
                }
            }
        },
        "FlightSG": {
            "Type": "AWS::EC2::SecurityGroup",
            "Properties": {
                "Tags": [
                    {
                        "Key": "Name",
                        "Value": {
                            "Ref": "AWS::StackName"
                        }
                    }
                ],
                "GroupDescription": "Enable SSH access to the Alces Clusterware master 
node",
                "VpcId": {
                    "Ref": "FlightVPC"
                },
                "SecurityGroupIngress": [
                    {
                        "IpProtocol": "-1",
                        "FromPort": "0",
                        "ToPort": "65535",
                        "CidrIp": "10.75.0.0/24"
                    },
                    {
                        "IpProtocol": "6",
                        "FromPort": "5900",
                        "ToPort": "5920",
                        "CidrIp": {
                            "Ref": "NetworkCIDR"
                        }
                    },
                    {
                        "IpProtocol": "6",
                        "FromPort": "22",
                        "ToPort": "22",
                        "CidrIp": {
                            "Ref": "NetworkCIDR"
                        }
                    }
                ],
                "SecurityGroupEgress": [
                    {
                        "IpProtocol": "-1",
                        "FromPort": "0",
                        "ToPort": "65535",
                        "CidrIp": "0.0.0.0/0"
                    }
                ]
            }
        },
        "MasterIAM": {
            "Condition":  "UseAlarms",
            "Type": "AWS::IAM::Role",
            "Properties": {
                "AssumeRolePolicyDocument": {
                    "Statement": [
                        {
                            "Effect": "Allow",
                            "Principal": {
                                "Service": [
                                    "ec2.amazonaws.com"
                                ]
                            },
                            "Action": [
                                "sts:AssumeRole"
                            ]
                        }
                    ]
                }
            }
        },
        "MasterIAMProfile": {
            "Condition":  "UseAlarms",
            "Type": "AWS::IAM::InstanceProfile",
            "Properties": {
                "Path": "/",
                "Roles": [
                    {
                        "Ref": "MasterIAM"
                    }
                ]
            }
        },
        "MasterPermissions": {
            "Condition":  "UseAlarms",
            "Type": "AWS::IAM::Policy",
            "Properties": {
                "PolicyName": "MasterPermissions",
                "PolicyDocument": {
                    "Version": "2012-10-17",
                    "Statement": [
                        {
                            "Effect": "Allow",
                            "Action": [
                                "cloudwatch:PutMetricData",
                                "autoscaling:DescribeAutoScalingGroups",
                                "autoscaling:SetDesiredCapacity",
                                "autoscaling:UpdateAutoScalingGroup",
                                "autoscaling:SetInstanceProtection",
                           "autoscaling:TerminateInstanceInAutoScalingGroup"
                            ],
                            "Resource": [
                                "*"
                            ]
                        }
                    ]
                },
                "Roles": [
                    {
                        "Ref": "MasterIAM"
                    }
                ]
            }
        },
        "ComputeIAM": {
            "Condition":  "UseAlarms",
            "Type": "AWS::IAM::Role",
            "Properties": {
                "AssumeRolePolicyDocument": {
                    "Statement": [
                        {
                            "Effect": "Allow",
                            "Principal": {
                                "Service": [
                                    "ec2.amazonaws.com"
                                ]
                            },
                            "Action": [
                                "sts:AssumeRole"
                            ]
                        }
                    ]
                }
            }
        },
        "ComputeIAMProfile": {
            "Condition":  "UseAlarms",
            "Type": "AWS::IAM::InstanceProfile",
            "Properties": {
                "Path": "/",
                "Roles": [
                    {
                        "Ref": "ComputeIAM"
                    }
                ]
            }
        },
        "ComputePermissions": {
            "Condition":  "UseAlarms",
            "Type": "AWS::IAM::Policy",
            "Properties": {
                "PolicyName": "ComputePermissions",
                "PolicyDocument": {
                    "Version": "2012-10-17",
                    "Statement": [
                        {
                            "Effect": "Allow",
                            "Action": [
                                "autoscaling:DescribeAutoScalingGroups"
                            ],
                            "Resource": [
                                "*"
                            ]
                        }
                    ]
                },
                "Roles": [
                    {
                        "Ref": "ComputeIAM"
                    }
                ]
            }
        },
        "FlightLogin": {
            "Type": "AWS::EC2::Instance",
            "DependsOn": "FlightGWAttach",
            "Properties": {
                "Tags": [
                    {
                        "Key": "Name",
                        "Value": {
                            "Fn::Join": [
                                "",
                                [
                                    {
                                        "Ref": "AWS::StackName"
                                    },
                                    "-login1"
                                ]
                            ]
                        }
                    }
                ],
                "IamInstanceProfile": {
                    "Fn::If": [
                        "UseAlarms",
                        {
                            "Ref": "MasterIAMProfile"
                        },
                        {
                            "Ref": "AWS::NoValue"
                        }
                    ]
                },
             "PlacementGroupName": {
                    "Fn::If": [
                    "LoginUsePlacement",
                    {
                        "Ref": "PlacementGroup"
                    },
                    {
                        "Ref": "AWS::NoValue"
                    }
                 ]
             },
                "ImageId": {
                    "Fn::FindInMap" : [ "AWSRegionArch2AMI", { "Ref" : "AWS::Region" }, {
"Fn::FindInMap" : [ "AWSInstanceType2Arch", { "Fn::FindInMap":
[ "FlightTypeToInstanceType", { "Ref": "LoginType" }, "InstanceType" ] }, "Arch" ] } ]
                },
                "NetworkInterfaces": [
                    {
                        "AssociatePublicIpAddress": "True",
                        "DeviceIndex": "0",
                        "GroupSet": [
                            {
                                "Ref": "FlightSG"
                            }
                        ],
                        "SubnetId": {
                            "Ref": "FlightSubnet"
                        }
                    }
                ],
                "BlockDeviceMappings": [
                    {
                        "DeviceName": "/dev/sda1",
                        "Ebs": {
                            "VolumeSize": {
                            "Ref": "LoginSystemDiskSize"
                            }
                        }
                    },
                    {
                        "DeviceName": "/dev/xvdb",
                        "VirtualName": "ephemeral0"
                    },
                    {
                        "DeviceName": "/dev/xvdc",
                        "VirtualName": "ephemeral1"
                    }
                ],
                "InstanceType": {
                    "Fn::FindInMap": [ "FlightTypeToInstanceType", { "Ref":
"LoginType" }, "InstanceType" ]
                },
                "KeyName": {
                    "Ref": "KeyPair"
                },
                "UserData": {
                    "Fn::Base64": {
                        "Fn::Join": [
                            "",
                            [
                                "#cloud-config\n",
                                "system_info:\n",
                                "  default_user:\n",
                                "    name: ", { "Ref": "Username" }, "\n",
                                "hostname: login1\n",
                                "write_files:\n",
                                "- content: |\n",
                                "    cluster:\n",
                                "      uuid: '11111111-2222-3333-444444444444'\n",
                                "      token: '1A0a1aaAA1aAAA/aaa1aAA=='\n",
                                "      name: ", { "Ref": "AWS::StackName" }, "\n",
                                "      role: 'master'\n",
                                "      tags:\n",
                                "        scheduler_roles: ':master:'\n",
                                "        storage_roles: ':master:'\n",
                                "        access_roles: ':master:'\n",
                           "        ",{ "Fn::If": ["UseAlarms", "autoscaling: enabled\n",
"autoscaling: disabled\n" ] },
                                "  owner: root:root\n",
                                "  path: /opt/clusterware/etc/config.yml\n",
                                "  permissions: '0640'\n"
                            ]
                        ]
                    }
                }
            }
        },
        "ComputeConfig": {
            "Type": "AWS::AutoScaling::LaunchConfiguration",
            "Properties": {
                "AssociatePublicIpAddress": "True",
                "KeyName": {
                    "Ref": "KeyPair"
                },
                "ImageId": {
                    "Fn::FindInMap" : [ "AWSRegionArch2AMI", { "Ref" : "AWS::Region" }, {
"Fn::FindInMap" : [ "AWSInstanceType2Arch", { "Fn::FindInMap":
[ "FlightTypeToInstanceType", { "Ref": "ComputeType" }, "InstanceType" ] }, "Arch" ] } ]
                },
             "IamInstanceProfile": {
                    "Fn::If": [
                        "UseAlarms",
                        {
                            "Ref": "ComputeIAMProfile"
                        },
                        {
                            "Ref": "AWS::NoValue"
                        }
                    ]
                },
                "InstanceType": {
                     "Fn::FindInMap": [ "FlightTypeToInstanceType", { "Ref":
"ComputeType" }, "InstanceType" ]
                },
                "BlockDeviceMappings": [
                    {
                        "DeviceName": "/dev/sda1",
                        "Ebs": {
                            "VolumeSize": "20"
                        }
                    }
                ],
                "SpotPrice": {
                    "Fn::If": [
                        "UseSpotCompute",
                        {
                            "Ref": "ComputeSpotPrice"
                        },
                        {
                            "Ref": "AWS::NoValue"
                        }
                    ]
                },
                "SecurityGroups": [
                    {
                        "Ref": "FlightSG"
                    }
                ],
                "UserData": {
                    "Fn::Base64": {
                        "Fn::Join": [
                            "",
                            [
                                "#cloud-config\n",
                                "system_info:\n",
                                "  default_user:\n",
                                "    name: ", { "Ref": "Username" }, "\n",
                                "write_files:\n",
                                "- content: |\n",
                                "    cluster:\n",
                                "      uuid: '11111111-2222-3333-444444444444'\n",
                                "      token: '1A0a1aaAA1aAAA/aaa1aAA=='\n",
                                "      name: ", { "Ref": "AWS::StackName" }, "\n",
                                "      role: 'slave'\n",
                                "      master: ", { "Fn::GetAtt": [ "FlightLogin",
"PrivateIp" ] }, "\n",
                                "      tags:\n",
                                "        scheduler_roles: ':compute:'\n",
                                "  owner: root:root\n",
                                "  path: /opt/clusterware/etc/config.yml\n",
                                "  permissions: '0640'\n"
                            ]
                        ]
                    }
                }
            }
        },
        "PlacementGroup": {
            "Type": "AWS::EC2::PlacementGroup",
            "Properties": {
                "Strategy": "cluster"
            }
        },
        "FlightCompute": {
              "Type": "AWS::AutoScaling::AutoScalingGroup",
              "DependsOn": "FlightLogin",
              "Properties": {
                  "DesiredCapacity": {
                    "Fn::If": [
                        "UseAlarms",
                        {
                            "Ref": "InitialNodes"
                        },
                        "8"
                    ]
                  },
                  "LaunchConfigurationName": {
                      "Ref": "ComputeConfig"
                  },
                  "PlacementGroup": {
                      "Ref": "PlacementGroup"
                  },
                  "Tags": [
                      {
                          "Key": "Name",
                          "Value": {
                            "Fn::Join": [
                                "",
                                [
                                    {
                                        "Ref": "AWS::StackName"
                                    },
                                    "-compute"
                                ]
                            ]
                        },
                        "PropagateAtLaunch": "true"
                      }
                  ],
                  "MinSize": "1",
                  "MaxSize": "8",
                  "VPCZoneIdentifier": [
                    {
                        "Ref": "FlightSubnet"
                    }
                  ]
              }
        },
        "ScaleUp": {
              "Condition":  "UseAlarms",
              "Type": "AWS::AutoScaling::ScalingPolicy",
              "Properties": {
                "AdjustmentType": "ChangeInCapacity",
                "AutoScalingGroupName": {
                    "Ref": "FlightCompute"
                },
                "Cooldown": "300",
                "ScalingAdjustment": "1"
              }
        },
        "CPUAlarmHigh": {
              "Condition":  "UseAlarms",
              "Type": "AWS::CloudWatch::Alarm",
              "Properties": {
                "AlarmDescription": "Scale-up if currently queued jobs exceeds limit",
                "MetricName": "JobStatus_qw",
                "Namespace": "ALCES-SGE",
                "Statistic": "Average",
                "Period": "60",
                "EvaluationPeriods": "3",
                "Threshold": "0",
                "AlarmActions": [
                    {
                        "Ref": "ScaleUp"
                    }
                ],
                "Dimensions": [
                    {
                        "Name": "AutoScalingGroupName",
                        "Value": {
                            "Ref": "FlightCompute"
                        }
                    }
                ],
                "ComparisonOperator": "GreaterThanThreshold"
              }
        }
    },
    "Outputs": {
          "Username": {
            "Description": "Administrator username used to log in to your environment
with. This should be used in conjunction with your selected AWS keypair - together with
the provided access IP",
            "Value": {
                "Ref": "Username"
            }
          },
          "AccessIP": {
            "Description": "Public access IP for your Flight Compute environment. Use
together with your chosen username to gain SSH access",
            "Value": {
                "Fn::GetAtt": [
                    "FlightLogin",
                    "PublicIp"
                ]
            }
          }
    }
}
```