



# Memory Machine™ Cloud Edition

## User Guide

First Published: 2023-01-27

Last Revised: 2023-02-06

Copyright 2022 - MemVerge. All rights reserved.

# Contents

<b>Introduction.....</b>	<b>3</b>
<b>Getting Started.....</b>	<b>4</b>
<b>Background.....</b>	<b>5</b>
<b>Memory Machine CE Architecture .....</b>	<b>6</b>
<b>Cloud Service Providers.....</b>	<b>10</b>
<b>Containers.....</b>	<b>12</b>
<b>Job Schedulers and Workflow Management Systems.....</b>	<b>13</b>
<b>Memory Machine CE Operation.....</b>	<b>14</b>
<b>Obtaining an OpCenter License for AWS.....</b>	<b>15</b>
<b>Deploying the OpCenter in AWS.....</b>	<b>16</b>
<b>Using the Memory Machine CE CLI.....</b>	<b>20</b>
<b>OpCenter in Everyday Use.....</b>	<b>22</b>
Running a Batch Job.....	22
WaveRider.....	24
Manual Job Migration.....	25
Policy-driven Job Migration.....	26
Programmatic Job Migration.....	28
Upgrading OpCenter.....	30
Uninstalling OpCenter.....	31
<b>Support and Troubleshooting.....</b>	<b>32</b>
<b>Memory Machine CE CLI.....</b>	<b>34</b>
<b>CLI Command Reference.....</b>	<b>35</b>

---

# Introduction

## What is Memory Machine™ Cloud Edition?

Memory Machine Cloud Edition (CE) is a software platform that streamlines the way containerized applications are deployed in the cloud.

Based on user-supplied policy, Memory Machine CE selects and instantiates cloud resources on behalf of the user. Memory Machine CE has a built-in job scheduler so users can deploy Docker containers (and other containers that comply with the Open Container Initiative *image-spec*) across a group of virtual machines.

Memory Machine CE includes AppCapsule, MemVerge's checkpoint/restore (C/R) capability. The AppCapsule is a moment-in-time snapshot of the application instance, including in-memory state and relevant files. AppCapsule is used to support workload mobility and workload continuity. Workload mobility means that a job can be moved from one virtual machine to another, for example, to a more powerful virtual machine that is a better fit for the next stage of execution. Workload continuity uses workload mobility to provide high availability — if the underlying virtual machine is reclaimed, the workload automatically moves to a new virtual machine and resumes running.

Most of the time, hyperscale Cloud Service Providers (CSPs) have excess virtual machine capacity, which they offer as Spot Instances at varying discounts, sometimes as much as 90%. The trade-off is that any Spot Instance can be reclaimed with only nominal warning (typically, two minutes or less). Memory Machine CE's AppCapsule feature is triggered automatically when the CSP signals that it is reclaiming the Spot Instance. Job execution pauses and then resumes on a new Spot Instance, allowing users to take advantage of the reduced cost of Spot Instances without the risk of losing intermediate results before the job has run to completion.

## Getting Started

### About This Guide

Memory Machine CE relies on the Memory Machine CE OpCenter, a server that you deploy on a virtual machine in a Cloud Service Provider (CSP) network. This guide provides instructions on how to install the OpCenter and how to use it to run containerized applications in the CSP's network.

### Prerequisites

To install and run Memory Machine CE, you need an account with a CSP (such as AWS) and a working knowledge of the basic services the CSP offers.

Memory Machine CE's built-in job scheduler requires a job script — usually in the form of a shell script — that you provide.

### Supported Clouds

Memory Machine CE is designed to work on any cloud infrastructure. The Bondi 1.2 Release supports [AWS](#) and [Alibaba Cloud](#). Support for Microsoft's [Azure](#) and [Google Cloud](#) is planned.

### How This Guide is Organized

The organization is as follows:

- [Background](#)  
Provides an overview of the Memory Machine CE architecture, cloud computing, containers, and job schedulers.
- [Operation](#)  
Describes how to install the OpCenter and how to run a workload. Also provides a quick reference guide for everyday use of Memory Machine CE.
- [CLI Command Reference](#)  
Presents detailed information on commands available through the Memory Machine CE CLI.

---

# Background

## Introduction

Operating Memory Machine CE entails interacting with cloud services and third party software, for example, software related to containers and workflow management systems. This part of the Memory Machine CE User Guide provides background information on these topics. It is not an in-depth tutorial or reference; rather, this section highlights topics that you must be familiar with to use Memory Machine CE.

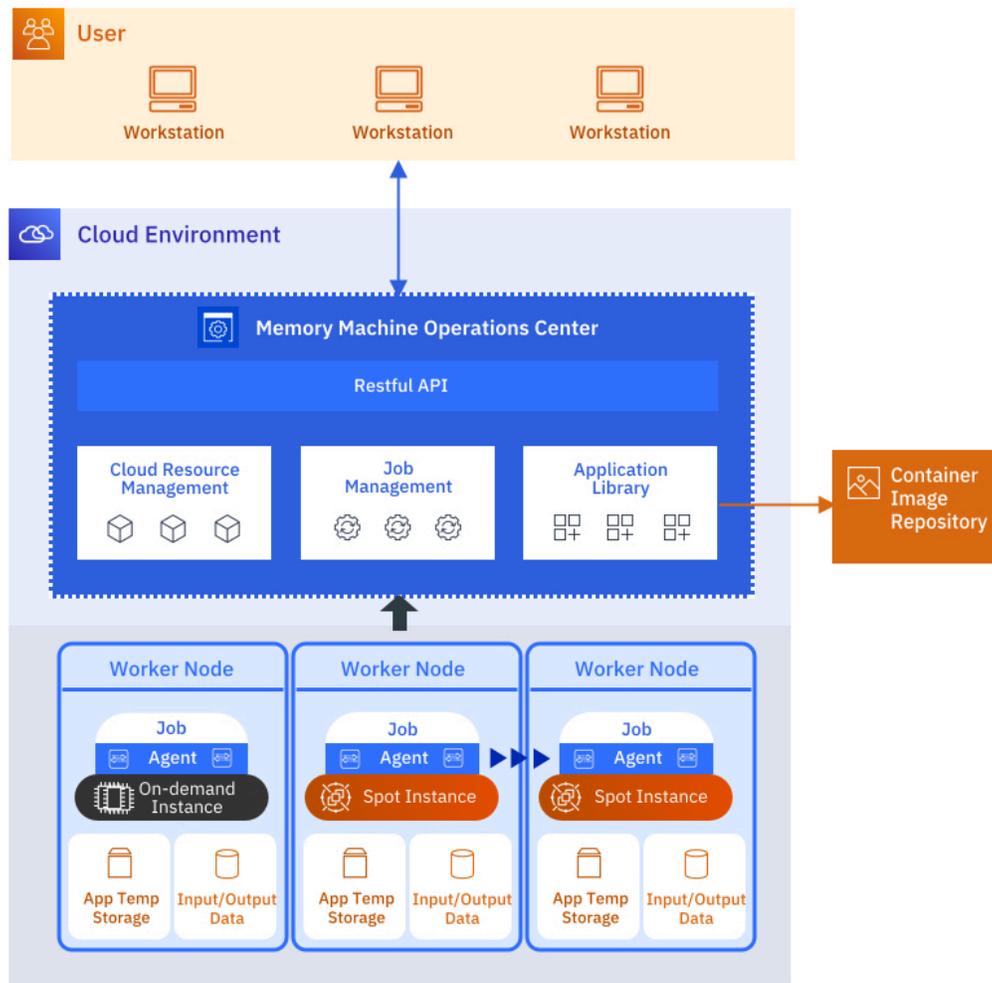
Users who have a working knowledge of these topics can skip this part and proceed to [Memory Machine CE Operation](#).

## Memory Machine CE Architecture

Memory Machine CE has several components that interact with cloud services to ensure mobility for submitted jobs so that, for example, jobs run to completion on Spot Instances.

### Design

Memory Machine CE enables users to run containerized applications on virtual machines leased from Cloud Service Providers. The virtual machines are usually Spot Instances but they can also be On-demand Instances (this is a per-job configuration option). All functions are controlled by the Memory Machine CE Operations Center, which receives CLI commands from users (clients) and manages resources in the cloud, as shown in [Figure 1: Memory Machine CE Architecture](#).



**Figure 1. Memory Machine CE Architecture**

### Components

The Memory Machine CE architecture includes the following components.

- Users (Clients)

Using Linux (or macOS) terminal shells or Windows command lines, clients use the Memory Machine CE CLI to interact with the Memory Machine CE OpCenter

- Memory Machine CE OpCenter

This provides the core functionality that allows Memory Machine CE to marshal resources for starting workloads and to migrate workloads if needed. If the OpCenter is not running, currently executing jobs continue (but are not migrated) and new jobs are not scheduled.

- Application Library

A container image registry is a service for hosting and distributing images. The default registry for Docker images is [Docker Hub](#). A repository is a collection of images within a registry (one registry hosts many repositories). A private repository requires a username and access token to post or retrieve images; a public repository does not. The Application Library contains a database of information for accessing container images in various repositories (public and private).

- Worker Nodes

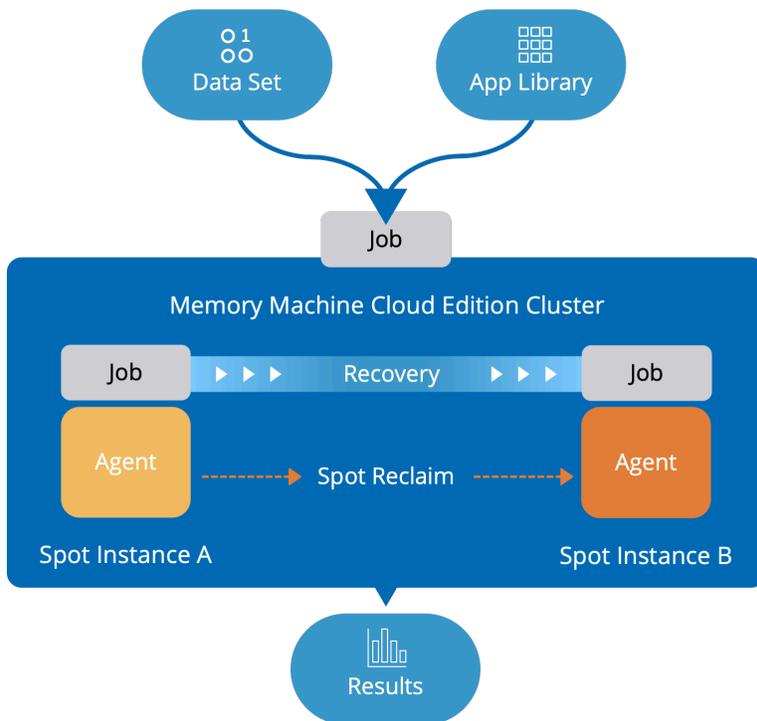
These are the compute engines provided by virtual machines running in the Cloud Service Provider's network. Worker nodes may have locally-mounted file systems and attached storage. On-demand Instances or Spot Instances can be used as worker nodes.

## Workflow

The operation of Memory Machine CE proceeds as follows. A client submits a job to the OpCenter, using CLI command options to select a container image from the Application Library and to specify the compute resources needed. The OpCenter uses this information to orchestrate the necessary resources in the Cloud Service Provider's network and schedules the job for execution. The cloud resources always include a compute node and may include block storage and file systems as well. One or more data sets usually accompany a job. The user-provided job script describes how these data sets are accessed, for example, by copying data from an AWS S3 bucket. The job script also describes where the output is placed — results are usually written to a persistent file system. When the job has run to completion, the user retrieves the results.

## Job Continuity

Using the AppCapsule feature, OpCenter automatically moves a workload running on a Spot Instance to a new Spot Instance if the first Spot Instance is reclaimed, as illustrated in [Figure 2: Memory Machine CE Operation](#).



**Figure 2. Memory Machine CE Operation**

In the example shown, the job starts executing on Spot Instance A. If the Cloud Service Provider signals that it intends to reclaim Spot Instance A, the OpCenter triggers the AppCapsule feature to capture the state of the running job and export the checkpoint image to persistent storage. A new Spot Instance is started (Spot Instance B in this case), the checkpoint image is imported from persistent storage, and the job resumes execution.

The job continues to run in this manner until completion, at which point the user retrieves the final results.

## Job Migration

For jobs running on Spot Instances, job migration occurs automatically if the Spot Instance is reclaimed. The new Spot Instance is of the same type as the one that was reclaimed.

Using the CLI, you can manually migrate a job from one virtual machine to another, for example, from a Spot Instance to an On-demand Instance of a different type. Using CLI command options, you can specify the new On-demand Instance by the instance type (for example, `c6xi.large` in AWS) or by specifying ranges of memory size and number of virtual CPUs.

Job migration is useful for "right-sizing" compute platforms: jobs can pass through several execution stages where the compute requirements are different; for example, one stage needs memory optimization whereas another stage needs compute optimization. Job migration moves the workload from one compute platform to another as the performance demands change.

Job migration can be initiated in three ways:

- Manually, using CLI commands
- Automatically, using a rules-based policy driven by resource utilization, for example CPU or memory utilization.
- Programatically, by inserting `float migrate` commands at breakpoints specified in the job script, for example, after loading data.

## Licensing

MemVerge maintains a [license server](#) where you request licenses for MemVerge products. Each Cloud Service Provider has its own mechanism for supporting the BYOL (Bring Your Own License) model. For example, AWS maintain its own License Manager so that you can apply licenses granted by software vendors to applications you run in AWS's network. After the OpCenter license is granted, the MemVerge license server uses an API to load the license into the AWS License Manager, where you must accept and activate the license.

For some Cloud Service Providers, you must apply a license key or upload a license file directly, without using a license manager in the Cloud Service Provider's network.

## Cloud Service Providers

To use Memory Machine CE, you need an account with a Cloud Service Provider.

### Introduction

Each of the global Cloud Service Providers (CSPs) offers hundreds of cloud-based services, ranging from general purpose computing and database systems to leading-edge capabilities such as Function-as-a-Service (serverless computing). The basic services are similar across CSPs although the terminology varies. To run containerized applications in the cloud, Memory Machine CE uses several of these services. A working knowledge of the version of these services provided by your CSP is required to deploy Memory Machine CE.

### CSP Services Used by Memory Machine CE

The following CSP services are used by Memory Machine CE:

- Virtual Machines

CSPs group virtual machines into classes, generally based on the type of processor used. Within a class, virtual machines are distinguished by the number of virtual CPUs and the amount of memory. Each virtual machine type has a name, for example, c6a.4xlarge in AWS.

- Virtual Private Cloud (VPC)

The VPC is a network of cloud resources that act like a private data center belonging to one customer.

- Availability Zones and Regions

To maximize the availability of cloud resources, CSPs distribute their data centers into geographically diverse regions. Within a single region, resources are further separated into zones — bounded by latency — so that so that if a resource fails in one zone, it can fail over to another zone in the same region.

- Identity and Access Management (IAM)

IAM is a service that allows you to control access to resources and authorization to perform certain tasks.

- Persistent file system

Although not required, access to a file system is useful for loading data and storing results. The file system persists after the virtual machine is deleted.

- Firewall

Virtual firewalls control network access to resources by imposing inbound and outbound filtering rules. For some applications, such as RStudio, an inbound rule must be applied to allow access to the container running RStudio on a worker node.

- Management Console

To deploy and operate Memory Machine CE, users must configure and view resources in the CSP's network. The interface that acts as the entry point for users is the Management Console. From there, users can navigate to other service-specific consoles and dashboards.

## Terminology used by CSPs

**Table 1. Mapping of Services across CSPs**

Service	AWS	Azure	Google Cloud Platform	AliCloud
Virtual Machine	Elastic Compute Cloud (EC2)	Virtual Machine	Compute Engine	Elastic Compute Service (ECS)
Object Storage	Simple Storage Service (S3)	Azure Blob Storage	Cloud Storage	Object Storage Service (OSS)
Block Storage	Elastic Block Store (EBS)	Azure Disk Storage	Persistent Disk	Elastic Block Storage
Virtual Private Cloud	Amazon Virtual Private Cloud	Azure Virtual Network	Virtual Private Cloud	Virtual Private Cloud
Regions and Availability Zones	Regions and Availability Zones	Regions and Zones	Azure Availability Zones and Regions	Regions and Zones
IAM	Amazon IAM	Azure Identity and Access Management	IAM	IDaaS
Firewall	Security Groups	Azure Firewall	Google Cloud Firewalls	Cloud Firewall
CLI	AWS CLI	Azure CLI	Cloud SDK	Alibaba Cloud CLI
Marketplace	AWS Marketplace	Azure Marketplace	Marketplace	Marketplace
Cloud Deployment	AWS CloudFormation	Azure Deployment Manager	Cloud Deployment Manager	Compute Nest
Management Portal	AWS Management Console	Azure Portal	Google Cloud Console	Alibaba Cloud Management Console

## Using CSP Services with Memory Machine CE

OpCenter runs on a virtual machine inside the CSP's network. From there, OpCenter selects, configures, and deploys the resources required to run the jobs submitted to OpCenter. In some cases, the user must perform additional configuration tasks manually. Although the procedures are different for each CSP, the procedures always begin by requiring the user to log in to the CSP's Management Console.

Most OpCenter maintenance procedures can be performed by using `float` commands, but the procedures for instantiating or deleting OpCenter must be carried out using the CSP's Management Console.

## Containers

Containers provide a standardized and repeatable way to package, deploy, and manage applications in the cloud.

### Introduction

A container image is a template for creating a container, i.e., a container is the runnable instance of a container image. Applications submitted to Memory Machine Cloud Edition must be packaged in images compliant with the [Open Container Initiative](#) (OCI) image format. An example of an OCI-compliant container image is Docker.

### Docker

The company [Docker](#) was an early developer of Linux container technology and the Docker platform is often considered the de-facto standard. Docker uses a client-server architecture in which the client uses `docker` commands to interact with the docker daemon (`dockerd`). Docker is a complete container environment — images are built, and containers are deployed and managed using the `docker` CLI.

### Container Image Registry

A container repository is a collection of container images. A registry is a collection of repositories and also a service that allows users to store, access, and share images in specific repositories. A repository may be public in which case all images in the repository are available to all users. A private repository restricts access so that only users who have a valid username and access token may access the repository.

MemVerge maintains a private repository at [Docker Hub](#) to store images that are automatically loaded by Memory Machine CE.

### Use with Memory Machine CE

If you use existing images (accessible from a public or private repository) to submit workloads to Memory Machine CE, you can complete all the necessary tasks using the `float` CLI. To build your own custom image and upload it to a repository, you can use `docker` or another container management tool such as `podman`.

If you build a container image on your local server, using `docker` or `buildah` for example, you can bypass the image registry and upload the image directly to the OpCenter using `float` commands.

## Job Schedulers and Workflow Management Systems

Memory Machine CE includes a job scheduler for running batch jobs in the cloud.

### Introduction

*Workload* managers, also known as job schedulers, are used to schedule, execute, and monitor batch jobs on clusters of compute nodes. Contention for resources is resolved by maintaining a queue of pending work. *Workflow* management systems are used to manage data-analytic pipelines, i.e., sequences of computational tasks such as found in genomic analyses.

Examples of job schedulers are:

- Slurm
- IBM Spectrum LSF ("load sharing facility")
- AWS Batch

Examples of workflow management systems are:

- Cromwell
- Nextflow

### Job Scripts

A job script is a text file that has job setup information for the workload manager followed by commands to execute. A job script must start with a shebang (`#!/bin/bash` is commonly used but you can use others). A job script may be as simple as a few lines of shell script but is often more complicated.

Most jobs submitted to OpCenter include a job script, but it is possible to submit a job without a job script.

---

# Memory Machine CE Operation

## Summary

To deploy and use Memory Machine CE, complete the following two tasks (the order is not significant). There are steps in each task that are different for each Cloud Service Provider (CSP). Consult the relevant sections in this guide for the steps that apply to your CSP.

- Obtain and activate the license that enables you to use OpCenter.

To request a license, go to the [MemVerge license server](#) and follow the on-line instructions. Each CSP has its own method of supporting BYOL (Bring Your Own License) subscriptions. For example, in AWS, use the [AWS License Manager](#) to accept and activate the license.

- Create and start an OpCenter instance.

Each CSP has a procedure for provisioning a collection of resources as a single group. MemVerge provides a template that describes the resources you need. Use this template with the CSP-specific service that assembles and instantiates these resources. With AWS, for example, use the [AWS CloudFormation](#) service.

Once the OpCenter is running, the software can be upgraded without needing to re-provision CSP resources.

Anyone with login credentials on the OpCenter instance can submit jobs. All interactions with the OpCenter use the Memory Machine CE CLI. There are three ways to access the CLI:

- Use the CLI console provided by the OpCenter web interface.
- Download the CLI binary and run it in a terminal window on your local machine.
- Log in to the OpCenter server and run the CLI in a secure terminal session.

The following sections provide detailed instructions for these tasks. The tasks have steps that are CSP-specific. Look for the section that applies to your CSP.

# Obtaining an OpCenter License for AWS

## Before You Begin

Have your AWS account ID available.

**Step 1** Open a browser and go to the [MemVerge license server](#); enter your credentials at the sign-in screen.



**Note:** If you are not registered, click on the *Register* button and follow the instructions.

**Step 2** In the Memory Machine Cloud Edition tab, click on *Request License*.

**Step 3** Enter your AWS account ID. If you are requesting a license to use in the AWS China region, check the box marked *AWS China Region*.

**Step 4** Click on *Confirm*.

A *License information* screen displays.

**Step 5** When the *License State* shows *Pending accept*, log in to your *AWS Management console* and go to the *AWS License Manager* console.

**Step 6** On the panel on the left-hand side of the *License Manager* console, click on *Granted licenses*.

The *Granted licenses* panel shows the licenses that are available to you.

**Step 7** Select the license associated with the *Product name* Memory Machine Cloud Edition and *Issuer* MemVerge.

**Step 8** If you have not accepted this license previously, click the *Activate* button on the right-hand side, near the top.



**Note:** If you are in a different region from the home region of the OpCenter license, a pop-up message explains that all license-related actions (activate, deactivate, etc.) must take place in the license's home region. Click on the *Make edits in Home Region* button to change to the license's home region and click *Activate* again.

**Step 9** If needed, change to the region where you create the *CloudFormation stack*.



**Note:** Once activated in its home region, the license is valid in all regions.

# Deploying the OpCenter in AWS

## Before You Begin

- Contact your AWS administrator to obtain an AWS account with root privileges.
- You need to select a VPC subnet and enter information on CIDR blocks. If you don't have this information, contact your AWS administrator.

**Step 1** Log in to your *AWS Management console*

**Step 2** Create a key pair

**Step 2.1** From the navigation bar, click on *Services* and select *EC2*.

**Step 2.2** On the right-hand side of the navigation bar of the *EC2 Dashboard*, check what region you are in.

If you need to change regions, use the drop-down menu to select a new region.

**Step 2.3** From the left-hand panel of the *EC2 Dashboard*, go to *Network & Security*, and click on *Key Pairs*.

The *Key Pairs* panel shows the key pairs that are available. If you need to create a key pair, click on the *Create key pair* button on the right-hand side and follow the instructions. A copy of the key pair file is automatically downloaded to your local machine.



**Note:** The key pair is unique to a region. If you change regions, you must create a key pair for the new region.

**Step 3** Identify the Availability Zone in which each VPC subnet is located

**Step 3.1** Using the search tool in the navigation bar, find the *VPC dashboard*.

**Step 3.2** On the left-hand panel of the *VPC dashboard*, click on *Subnets*.

The *Subnets* console displays a table that shows each VPC subnet, the VPC the subnet is part of, and the Availability Zone in which the subnet resides.

**Step 3.3** Keep this browser tab open.

You need to retrieve information from this table later in the deployment procedure.

**Step 4** Locate Memory Machine Cloud Edition in the AWS Marketplace

*Choose one of the following:*

- Log in to the [MemVerge License Manager](#), then go to step 4.1.
- Log in to the *AWS Management console*, then go to step 4.4.

**Step 4.1** Select the Memory Machine Cloud Edition tab.

**Step 4.2** On the top, right-hand side, click on *Deploy Product > AWS*.

A new browser tab opens at the Memory Machine Cloud Edition entry in the AWS Marketplace. If you are not logged in to AWS, enter your credentials at the prompt.

**Step 4.3** Skip to step 5.

**Step 4.4** While logged in to AWS, go the AWS Marketplace and select *Discover products* from the left-hand panel.

**Step 4.5** Search for Memory Machine Cloud Edition.

**Step 4.6** Click on the result and then proceed to the next step.

**Step 5** Subscribe to Memory Machine Cloud Edition

**Step 5.1** Click on *Continue to Subscribe*.

**Step 5.2** If you agree to the terms and conditions, click on *Accept Terms*, then click on *Continue to Configuration*.

**Step 5.3** At the *Configure this software* screen, use the pull-down menu to select the region in which to deploy OpCenter.

**Step 5.4** Click on *Continue to Launch*.

**Step 5.5** Under *Choose Action*, use the pull-down menu to select *Launch CloudFormation*.

**Step 5.6** Click on *Launch*.

You are now at the *CloudFormation > Stacks > Create stack* screen. The left-hand panel shows the four-step procedure for creating a new stack. You are at the first step.

**Step 5.7** Click on *Next*.

**Step 6** Specify the *CloudFormation* stack details

**Step 6.1** Enter a unique stack name, using the allowed characters.

**Step 6.2** Fill in the required parameters as follows (seek guidance from your AWS administrator if needed).

- *OmvOpCenterType*: Accept the default or use the pull-down menu to change the size of the VM to run OpCenter.
- *ImvFloatKeyName*: Use the pull-down menu to select key pair name you created in step 2.

- *2mvFloatVPCID*: Select a VPC from the pull-down menu. (Usually, there is only one VPC; if there are multiple, consult your AWS administrator for guidance.)
- *3mvFloatSubnetID*: Select a VPC subnet from the pull-down menu. (Usually, VPC subnets are equivalent; consult your AWS administrator for guidance.)
- *4mvAvailabilityZone*: Go to the browser tab you opened in step 3 to determine the Availability Zone in which your VPC subnet resides. Then return here to use the pull-down menu to select the appropriate Availability Zone.
- *5mvFloatPublicCIDR*: Provide the range of IP addresses allowed to access server. To allow access from any address, enter `0.0.0.0/0`
- *6mvFloatSSHCIDR*: Provide the range of IP addresses allowed to access server using `ssh`. To allow `ssh` access from any address, enter `0.0.0.0/0`
- *7mvFloatInternalCIDR*: Provide the range of IP addresses for internal communication among OpCenter instances. To allow communication from any address, enter `0.0.0.0/0`
- *8mvDataVolumeSize*: Accept the default or enter a whole number.

**Step 6.3** Click on *Next*.

### Step 7 Configure stack options

Keep the default options and click on *Next*.

### Step 8 Review

**Step 8.1** Check the box at the bottom of the page to acknowledge that you are aware *CloudFormation* may create IAM resources.

**Step 8.2** Click on *Create stack* to proceed.

Wait until the process completes successfully.

### Step 9 Check your stack

**Step 9.1** Go to the *CloudFormation > Stacks* screen and select your stack.

**Step 9.2** From the *CloudFormation > Stacks > "your\_stack\_name"* screen, select the *Resources* tab.

**Step 9.3** Click on *Physical ID* associated with *mvOpCenter*.

This takes you to the *Instances* screen of the EC2 console.

**Step 9.4** Select the OpCenter instance.

This screen displays OpCenter status as well as private and public IP addresses associated with the OpCenter.

**Step 9.5** Depending on whether you are inside or outside your organization's virtual private cloud, open a browser and go to the OpCenter private or public IP address, respectively.

If the OpCenter is running, the OpCenter web interface displays the OpCenter status as well as the license status.

## Using the Memory Machine CE CLI

**Step 1** Select your method for invoking Memory Machine CE CLI commands.

Choose one of the following:

- Using the OpCenter web interface, click on **Connect to OpCenter**.

At the login screen, enter username *admin* and default password *memverge*. A new tab opens. Click on the terminal icon displayed on the top right-hand side of the *Jobs* screen to open the *web CLI*.

A subset of CLI commands is available via the OpCenter web interface. Type **-h** to see what commands are supported. Skip to step 3.

- Using the OpCenter web interface, click on **Connect to OpCenter** (it is not necessary to log in) and then click on **Download CLI tool** displayed at the bottom of the login screen.

Use **chmod** to make the downloaded **float** binary executable and add path to the **float** binary to your **PATH** variable.



**Note:** From macOS 10.15 (Catalina) onwards, zsh is the default shell. In zsh, *float* is a reserved word. To use the Memory Machine CE CLI with macOS, either change the shell to bash or alias the word *float* to the **float** binary, as follows:

```
alias float = /path/to/float_binary/float
```

where **/path/to/float\_binary/** is the path to where you placed the **float** binary.

- Establish a secure terminal session (using **ssh**) on the OpCenter server. The **float** CLI is automatically installed on this server.



**Note:** The *web CLI* is a web interface that provides a **float** prompt, that is, every entry you type automatically has **float** pre-pended. The other two methods provide a terminal with a shell prompt — you must type **float** followed by subcommand and options.

**Step 2** Log in to the OpCenter as the *admin* user by typing the following:

```
float login -u admin -p memverge -a <OpCenter_ip_address>
```

where **<OpCenter\_ip\_address>** is the OpCenter's private IP address if you are within your organization's virtual private cloud (VPC), or public IP address if you are outside the VPC.

**Step 3** Add a user by entering the following command:

```
float user add <user_name> --passwd <password> --group <group_name> --create
```

where **<user\_name>**, **<password>**, and **<group\_name>** are the user name, password, and group, respectively. Repeat the command with different parameters to add other users.

**Step 4** Change the default password for the *admin* user by typing the following command:

```
float user passwd admin --passwd <admin_password>
```

where `<admin_password>` is the new password for the *admin* user.

**Step 5** From a terminal session (not the *web CLI*), log in to the OpCenter as the *admin* user by typing the following:

```
float login -u admin -p <admin_password> -a <OpCenter_ip_address>
```

where `<OpCenter_ip_address>` is the OpCenter's private IP address if you are within your organization's virtual private cloud (VPC), or public IP address if you are outside the VPC.

**Step 6** Upgrade to the latest OpCenter version.

**Step 6.1** To see what releases are available, type the following command:

```
float release ls
```

**Step 6.2** If there is a later release, upgrade by typing the following command:

```
float release upgrade -r <release_name>
```

where `<release_name>` is the latest release shown in the previous step. After the upgrade completes, log back in.

**Step 6.3** Synchronize the `float` version with the OpCenter version by typing the following command:

```
float release sync
```

# OpCenter in Everyday Use

## Introduction

Most users only run a subset of the `float` CLI commands in their everyday use of OpCenter for running workloads. OpCenter also includes support for upgrading the software without affecting its virtual machine instance, migrating running jobs to other virtual machine types, and testing application compatibility.

A detailed `float` [command reference](#) is available; this section provides summary descriptions for common use cases.

## Running a Batch Job

The Memory Machine Cloud Edition CLI has a rich set of options that allow you to customize your container runtime environment. The command strings shown here are simplified.

**Step 1** Log in to the OpCenter.

*Choose one of the following:*

- If you are using the *web CLI*, you are already logged in. Skip to step 2.
- If you are using a remote terminal or a terminal session on the OpCenter server, enter the following command:

```
float login -u <username> -p <password> -a <OpCenter_ip_address>
```

where `<username>` and `<password>` are login credentials, and `<OpCenter_ip_address>` is the OpCenter's private IP address if you are within your organization's virtual private cloud (VPC), or public IP address if you are outside the VPC.



**Note:** The OpCenter IP address is cached, so you can omit it when you log back in. After the cache expires, the IP address defaults to local host. If you get a "connection refused" error, retry with `-a <OpCenter_ip_address>` option included.

**Step 2** Display the images that are available in the OpCenter by entering the following command:

```
float image list
```

**Step 3** If the required image is not available, upload the image.

*Choose one of the following:*

- If the required image is available from a public repository, enter the following command:

```
float image add <image_name> <image_URI>
```

For example:

```
float image add python docker.io/bitnami/python
```

- If the required image is available from a private repository, enter the following command:

```
float image add <image_name> <image_URI> --user <repo_access_user> --token
<repo_access_token>
```

where `<repo_access_user>` and `<repo_access_token>` are the credentials for accessing the private repository. Obtain these from the repository owner.

- If the required image is stored on your local machine, enter the following command:

```
float image upload <image_name> --path </path/to/image>
```

where `</path/to/image>` is the path to the directory where the image tar file is located on your local machine.

**Step 4** Prepare the job script for the workload and make it accessible from the server on which you issue `float` commands.



**Note:** The job script can be a local file, but there are other access options, such as from an S3 bucket or from a web server.

**Step 5** Submit a job to the OpCenter by entering the following command:

```
float sbatch -i <image_name> -j <job_script> --cpu <num_cpu> --mem <mem_size>
--dataVolume [size=<vol_size>]:/<mnt_point>
```

where:

- `image_name` is the docker image to run the job
- `job_script` is the job script to execute (include the complete path to the job script file)
- `num_cpu` is the minimum number of virtual CPUs to use (can also specify as a range in form `min:max`)
- `mem_size` is minimum memory capacity to use in GB (can also specify as a range in form `min:max`)
- `vol_size` is the capacity of the data directory (in GB)
- `mnt_point` is the mount point for the data directory.

Example:

```
float sbatch -i python -j ./python_job_script.sh --cpu 4 --mem 8 --dataVolume
[size=10]:/data
```

Alternatively, you can use a definition file in conjunction with the `float sbatch` command. For example, to submit the same job, enter:

```
float sbatch -d ./def1.yaml
```

where `def1.yaml` is a file with the following contents:

```
image: python
job: python_job_script.sh
cpu: 4
mem: 8
```

```
dataVolume:
- "[size=10]:/data"
```

**Step 6** Check job status by entering the following command:

```
float squeue
```

The first column shows the unique job identifier associated with each job.

**Step 7** Display detailed information on the job you submitted by entering the following command:

```
float show -j <job_id>
```

where `<job_id>` is the unique job identifier shown in the previous step.

**Step 8** To display the logs associated with a job, enter the following:

```
float log ls <job_id>
```

where `<job_id>` is the job identifier.

**Step 9** If your job writes to a log file while it runs, view the output by entering the following command:

```
float log tail --follow <log_file_name> -j <job_id>
```

where `<log_file_name>` is the name of the log file you specified in the job script and `<job_id>` is the job identifier.



**Note:** As the job runs, logs are written to a directory mounted from the OpCenter server. When the job ends, the logs, for example, **stderr** and **stdout**, are automatically saved by the OpCenter as **stderr.autosave** and **stdout.autosave**, respectively.

**Step 10** To cancel a running a job, enter the following command:

```
float scancel -j <job_id>
```

### What To Do Next

If the job runs on a Spot Instance that is reclaimed, the job will "float" to a new Spot Instance and continue running. After the job has run to completion, retrieve your results from the location you specified in the job script.

## WaveRider

When enabled, WaveRider allows OpCenter to manage cloud resources by moving jobs to optimally-sized virtual machines.

### Feature Description

Resource utilization (CPU, memory and storage) is not constant while a job runs. Users face a difficult choice: over-provision and overpay, or under-provision and risk out-of-memory conditions. With

WaveRider, users can start with a small virtual machine (VM) and dynamically migrate to bigger VMs (and back down to smaller VMs) as the application demands change during a complete job run.

Job migration is initiated in three ways:

- Ad hoc: migration initiated manually using CLI commands.
- Automatic: migration triggered by rules-based policy configured per job (policy based on CPU and memory utilization thresholds).
- Programmatic: migration executed at specific program breakpoints defined by job script.

## Manual Job Migration

Running jobs can be moved manually at any time using the CLI, for example, to change the compute platform or the pay type.

### Step 1 Log in to the OpCenter.

Choose one of the following:

- If you are using the *web CLI*, you are already logged in. Skip to step 2.
- If you are using a remote terminal or a terminal session on the OpCenter server, enter the following command:

```
float login -u <username> -p <password> -a <OpCenter_ip_address>
```

where `<username>` and `<password>` are login credentials, and `<OpCenter_ip_address>` is the OpCenter's private IP address if you are within your organization's virtual private cloud (VPC), or public IP address if you are outside the VPC.

### Step 2 Migrate a running job to a new virtual machine instance.

Choose one of the following:

- To migrate a job to a specific virtual machine type, enter the following command:

```
float migrate -t <instance_type> -j <job_id>
```

where `<instance_type>` is a virtual machine type (for example, `c4.xlarge` in AWS) and `<job_id>` is the job identifier.

- To migrate to an instance whose capacities you define, enter the following command:

```
float migrate --cpu <minCPU>:<maxCPU> --mem <minMem>:<maxMem> -j <job_id>
```

where `<minCPU>:<maxCPU>` is the allowable range for the number of virtual CPUs, `<minMem>:<maxMem>` is the allowable range for the memory size in GB, and `<job_id>` is the job identifier. The upper bounds for number of virtual CPUs and memory size can be omitted in which case only the lower bounds apply.

- To migrate to an identical instance type (same memory capacity and number of virtual CPUs), enter the following command:

```
float migrate -j <job_id>
```

where `<job_id>` is the job identifier.



**Note:** The `vmPolicy` option associated with a running job can be modified. If the new policy is incompatible with the old policy (not considering `priceLimit`), then job migration is automatically triggered, so, for example, a job running on a Spot Instance migrates to an On-demand Instance.

**Step 3** Keep on issuing the `float squeue` command to monitor whether the job migrates successfully.

**Step 4** To view a record of any migration events, use the `float log cat job.events -j <job_id>` command.

Example of manual migration:

```
float migrate -f -j Xx0r4CYE7X6MRmivjoITf
float log cat job.events -j Xx0r4CYE7X6MRmivjoITf
2023-01-23T18:15:40.810895559Z: Ready to migrate with instance type:
c5.xlarge, cpu: 4, memory: 8, zone: us-east-1f, last instance type:
c5.xlarge(Spot)
2023-01-23T18:15:40.811126391Z: Ready to checkpoint host
i-004105d65ca4d2f43
2023-01-23T18:15:42.423102447Z: Checkpointed host i-004105d65ca4d2f43,
result: &{map[41c76520d597:]}, duration 1.611915792s
2023-01-23T18:15:42.637317005Z: Ready to reclaim host i-004105d65ca4d2f43
2023-01-23T18:16:25.51612009Z: Ready to create new host to recover
2023-01-23T18:16:26.344950731Z: Reclaimed host i-004105d65ca4d2f43
2023-01-23T18:18:08.758995433Z: Created new host:
i-0031d3ef20accf5a0(Spot)
2023-01-23T18:18:08.865722721Z: Got 1 containers on host
i-0031d3ef20accf5a0
2023-01-23T18:18:08.865763251Z: Ready to recover &{41c76520d597 false
true} on host i-0031d3ef20accf5a0
2023-01-23T18:18:08.872718907Z: Job floated
2023-01-23T18:18:09.306566654Z: Migrated to new VM: i-0031d3ef20accf5a0
```

### Policy-driven Job Migration

When enabled, WaveRider uses a rules-based policy to determine when to migrate a job to a different virtual machine

The `--migratePolicy` option can be used with the `float sbatch` command to define the rules that determine when a job is migrated to a new virtual machine. The policy can be added to a running job or an existing policy associated with a running job can be changed by using the `float modify` command.

The policy works as follows. If the upper threshold for CPU or memory utilization is crossed and utilization remains elevated for a specified interval, the job is migrated to a virtual machine that has more virtual CPUs or more memory. The increase in size is measured as a percentage defined by the `step` parameter. Similar behavior occurs if the lower threshold is crossed and utilization remains low for a specified interval: the job is moved to a smaller virtual machine.

**Step 1** Turn policy-driven migration on or off.

*Choose one of the following:*

- Policy-driven migration is disabled by default. To keep this setting, omit the `--migratePolicy` option. Equivalently, use `--migratePolicy [disable=true]` or `--migratePolicy [enable=false]`. Skip to step 4.
- To enable policy-driven migration, use `--migratePolicy [enable=true]` or `--migratePolicy [disable=false]`.

**Step 2** If needed, override default values for policy rules.

Parameters are attached to `--migratePolicy` as a string. The following parameters can be included in the string (default values listed in parentheses). If a unit is not shown, the value is a percentage of the maximum possible.

- `cpu.upperBoundRatio` (90): upper threshold for utilization per virtual CPU (percentage)
- `cpu.lowerBoundRatio` (5): lower threshold for utilization per virtual CPU (percentage)
- `cpu.upperBoundDuration` (30s): time that utilization per virtual CPU must remain above the upper threshold before migration is triggered
- `cpu.lowerBoundDuration` (5m0s): time that utilization per virtual CPU must remain below the lower threshold before migration is triggered
- `cpu.step` (50): The percentage increase (or decrease) in the number of virtual CPUs in the new virtual machine versus the original virtual machine
- `mem.upperBoundRatio` (90): upper threshold for memory utilization (percentage)
- `mem.lowerBoundRatio` (5): lower threshold for memory utilization (percentage)
- `mem.upperBoundDuration` (30s): time that memory utilization must remain above the upper threshold before migration is triggered
- `mem.lowerBoundDuration` (5m0s): time that memory utilization must remain below the lower threshold before migration is triggered
- `mem.step` (50): The percentage increase (or decrease) in memory capacity of the new virtual machine versus the original virtual machine



**Note:** Only parameters that have values different from the default need to be included in the `--migratePolicy` string.

**Step 3** Submit job with auto-migration policy included.

Example:

```
float sbatch -i python -j ./python_job_script.sh --dataVolume [size=10]:/data -c 4 -m
 8 --migratePolicy [enable=true,mem.upperBoundRatio=60]
id: 8dF5j3yTnuzbHFKUIJTYt
name: python-
user: tester
imageID: docker.io/bitnami/python:latest
status: Submitted
submitTime: "2023-01-22T23:25:47Z"
duration: 7s
```

```
inputArgs: ' -j ./python_job_script.sh -i python --migratePolicy
[enable=true,mem.upperBoundRatio=60] -m 8 -c 4 --dataVolume
[size=10]:/data '
vmPolicy:
  policy: spotFirst
  retryLimit: 3
  retryInterval: 10m0s
migratePolicy:
  cpu:
    upperBoundRatio: 90
    lowerBoundRatio: 5
    upperBoundDuration: 30s
    lowerBoundDuration: 5m0s
    step: 50
  mem:
    upperBoundRatio: 60
    lowerBoundRatio: 5
    upperBoundDuration: 30s
    lowerBoundDuration: 5m0s
    step: 50
```

**Step 4** Modify auto-migration policy associated with running job.

*Choose one of the following:*

- Keep the existing auto-migration policy or keep auto-migration turned off. Skip to the next step.
- To modify the auto-migration policy associated with a running job or to turn auto-migration on, use `float modify --migratePolicy <policy-string> -j <job_id>`

Example (turn auto-migration on and change one parameter from its default value):

```
float modify --migratePolicy [enable=true,mem.upperBoundRatio=70] -j
eFehjVpxNFet426BzCFgz
Warning: Are you sure you want to modify eFehjVpxNFet426BzCFgz?
New migratePolicy may impact auto-migration behavior.(yes/No): yes
Successfully modified eFehjVpxNFet426BzCFgz: --migratePolicy
[enable=true,mem.upperBoundRatio=70]
```

**Step 5** To view a record of any migration events, use the `float log cat job.events -j <job_id>` command.

## Programmatic Job Migration

Initiate job migration by inserting the `float migrate` command into the job script.

## Overview

Although not strictly required, most jobs are submitted with a job script that sets up the execution environment and tells the job scheduler what to do. Some jobs proceed in a series of stages, for example, an initial stage to read in a large dataset followed by a computationally-intensive stage. These two stages may repeat several times until the computation is complete. Then the results are written out to disk.

The resource requirements (CPU, memory, and storage) for each of these stages are different. A `float migrate` command can be inserted at each point where the job moves to a new stage with different resource requirements. In this manner, the job runs on a virtual machine that matches the requirements for the given stage.

## Example

The example shown here uses a simple shell program to demonstrate how programmatic migration can be implemented. In the example job script, the job is migrated at an arbitrary point. In a real application, the job would migrate at the point where resources were expected to change.

The contents of the job script are as follows:

```
cat progmigrate.sh
#!/usr/bin/env bash
LOG_PATH=$1
LOG_FILE=$LOG_PATH/output
touch $LOG_FILE
exec >$LOG_FILE 2>&1
echo "Starting to execute shell script"
echo "Job migrates when count reaches 50"
for(( c=1; c<300; c++))
do
    if [[ $((($c % 3)) == 0 )]; then
        echo "$c is a multiple of three"
    else
        echo "$c is NOT a multiple of three" >&2
    fi
    if [[ $c == 50 ]]; then
        /opt/memverge/bin/float migrate -f --instType c5.2xlarge
        echo "Job migration initiated, wait for a while"
        sleep 20s
    fi
    sleep 1s
done
echo "Job is complete"
```

The job runs on a centos9 image.

```
float submit -i centos9 -j ./progmigrate.sh --instType c5.large --dataVolume [size=10]:/data
```

The output file shows where the job migration occurs.

```
float log tail --follow output -j 4BZXIXWu6L5ioCSqEMD63
Starting to execute shell script
Job migrates when count reaches 50
1 is NOT a multiple of three
----[edited]
50 is NOT a multiple of three
4BZXIXWu6L5ioCSqEMD63 on i-0027c7d62eaabafa6 is now migrating, please use queue/show
to monitor migration progress. Check logs for details
Job migration initiated, wait for a while
51 is a multiple of three
----[edited]
299 is NOT a multiple of three
Job is complete
```

The log file shows the events occurring when migrating the job from a c5.large instance to a c5.2xlarge instance.

```
float log cat job.events -j 4BZXIXWu6L5ioCSqEMD63
----[edited]
2023-01-24T20:33:24.997727759Z: Ready to migrate with instance type: c5.2xlarge, cpu:
8, memory: 16, zone: us-east-1b, last instance type: c5.large(Spot)
```

```

2023-01-24T20:33:24.997929752Z: Ready to checkpoint host i-0027c7d62eaabafa6
2023-01-24T20:33:26.29553621Z: Checkpointed host i-0027c7d62eaabafa6, result:
  &{map[5ee5df78f592:]}, duration 1.297557077s
2023-01-24T20:33:26.515069329Z: Ready to reclaim host i-0027c7d62eaabafa6
2023-01-24T20:34:15.524359376Z: Ready to create new host to recover
2023-01-24T20:34:22.200051613Z: Reclaimed host i-0027c7d62eaabafa6
2023-01-24T20:35:55.597906896Z: Created new host: i-03fb670fbb577639e(Spot)
2023-01-24T20:35:55.694463545Z: Got 1 containers on host i-03fb670fbb577639e
2023-01-24T20:35:55.694528662Z: Ready to recover &{5ee5df78f592 false true} on host
  i-03fb670fbb577639e
2023-01-24T20:35:55.700293372Z: Job floated
2023-01-24T20:35:56.362953748Z: Migrated to new VM: i-03fb670fbb577639e
----[edited]

```

## Upgrading OpCenter

If a later version of OpCenter software is available, an OpCenter instance can be upgraded as long as no jobs are currently running.

**Step 1** Log in to OpCenter by entering the following command from a terminal:

```
float login -u admin -p <admin_passwd> -a <OpCenter_ip_address>
```

where `<admin_passwd>` is the admin password and `<OpCenter_ip_address>` is the OpCenter's private IP address if you are within your organization's virtual private cloud (VPC), or public IP address if you are outside the VPC.



**Note:** Only the *admin* user can upgrade software. Upgrades cannot be executed from the *web CLI*.

**Step 2** To see what version of OpCenter software is currently running, enter the following command:

```
float version
```

**Step 3** To see what versions of OpCenter software are available, enter the following command:

```
float release ls
```

**Step 4** If a later version of the software is available, upgrade the OpCenter instance by entering the following command:

```
float release upgrade --release <version>
```



**Note:** If `--release` is omitted, the latest version of the software is selected. If a job is running, the upgrade does not proceed.

**Step 5** After executing a software upgrade, check that the upgrade succeeded by entering the following command:

```
float version
```

**Step 6** If the OpCenter upgrade is successful, upgrade `float` to the same version by entering the following command:

```
float release sync
downloading ...
the float binary is synced up with opcenter
```

## Uninstalling OpCenter

Uninstalling requires deleting the CloudFormation stack running OpCenter.

**Step 1** Log in to the OpCenter

**Step 2** Use `float queue` to determine if any jobs are running.

**Step 3** Use `float scancel -j <job_id>` to cancel any running jobs.

**Step 4** Wait until all virtual machine instances have been reclaimed.

**Step 5** Log in to your *AWS Management console*.

Check what region you are in. If you need to change regions, use the drop-down menu to select the correct region.

**Step 6** Use the search tool in the navigation bar to find the *CloudFormation* console.

**Step 7** On the left-hand panel of the *CloudFormation* console, click on *Stacks*.

A list of all your CloudFormation stacks is displayed.

**Step 8** Select the OpCenter instance to delete.

**Step 9** Click on the *Delete* button.

## Support and Troubleshooting

### Support

Support for Memory Machine CE software is available 24 x 7 x 365. Contact MemVerge support by sending email to [support@memverge.com](mailto:support@memverge.com).

### Logs

OpCenter compiles logs of events related to its operation as well as logs that are specific to a particular job or a particular host (worker node).

To see the logs that pertain to the operation of the OpCenter, enter the following:

```
float log ls
+-----+-----+-----+
| LOG NAME | SIZE | LAST UPDATE TIME |
+-----+-----+-----+
| etcd.log | 239504 | 2023-02-02T01:46:04Z |
| opcenter.access_log | 2811941 | 2023-02-02T02:31:43Z |
| opcenter.log | 756620 | 2023-02-02T02:30:25Z |
| upgrade.log | 1974 | 2023-01-26T15:40:39Z |
+-----+-----+-----+
```

The most useful log for troubleshooting is the `opcenter.log`.

To see the logs that pertain to a particular job, enter the following:

```
float log ls -j <job_id>
```

Example:

```
float log ls -j TQ9PUJhoY0XLtL58KcU1M
+-----+-----+-----+
| LOG NAME | SIZE | LAST UPDATE TIME |
+-----+-----+-----+
| environments | 330 | 2023-02-02T02:21:04Z |
| job.events | 3009 | 2023-02-02T02:27:02Z |
| metrics-ale1970a868a.txt | 12232 | 2023-02-02T02:46:35Z |
| output | 25963 | 2023-02-02T02:46:40Z |
| stderr.autosave | 0 | 2023-02-02T02:21:27Z |
| stdout.autosave | 0 | 2023-02-02T02:21:27Z |
+-----+-----+-----+
```

The job script used to submit this job redirects `stderr` and `stdout` to a file called `output` (that is why `stderr.autosave` and `stdout.autosave` have zero size). The log called `job.events` is useful for troubleshooting.

To see the logs that pertain to a particular host, enter the following:

```
float log ls -i <host_id>
```

Example:

```
float log ls -i i-0e40db7d105cb6793
+-----+-----+-----+
| LOG NAME | SIZE | LAST UPDATE TIME |
+-----+-----+-----+
```

fagent.access_log	8494	2023-02-02T17:10:10Z
fagent.log	11735	2023-02-02T17:08:00Z
fagent_init.log	639	2023-02-02T17:06:34Z
internal_output	1116	2023-02-02T17:07:59Z
messages	154889	2023-02-02T17:10:01Z

The log called `messages` is useful for troubleshooting. The log called `fagent.log` records interactions between the worker node and the OpCenter.

## Troubleshooting

The following table shows commonly encountered errors and how to fix them.

**Table 2. Troubleshooting OpCenter Issues**

Error	Cause	Remedy
float login returns 'Get "https://127.0.0.1/api/v1/login": dial tcp 127.0.0.1:443: connect: connection refused'	Incorrect IP address for OpCenter or the OpCenter IP address has aged out of the local float cache.	Check OpCenter IP address and try again with float login -a <ip_address>
float command returns "Error: Session timeout (code: 2001)"	Current OpCenter session timed out	Log in to OpCenter
float image add or float submit returns "Error: Authentication failed, incorrect username or password (code: 2003)"	Attempt to access private repository with incorrect or missing credentials	Rerun command with valid credentials to access repository
float submit returns "Error: Unsupported argument, No instance types meet combined --cpu and --mem constraints. (code: 1027)"	No VM instance found that meets all requirements including price limit for Spot Instance	Resubmit with higher price limit if it is Spot Instance. Else, resubmit with different memory and vCPU ranges.
float queue shows status as "WaitingForLicense"	OpCenter cannot retrieve valid license	Check license status on MemVerge License Server. Obtain new license or activate existing license.
float log cat <logfile> returns "Error: Invalid argument, No such log"	Delay in writing to log file after the container starts running	Wait and then retry command
float ps returns "Error: Job is not executing"	Job is either initializing or it has completed	Use float queue to determine status of job
float queue returns "No jobs" although jobs have been submitted and completed	float queue queries job history in a fixed time interval (default: last hour)	Use float queue -A to include all jobs

---

# Memory Machine CE CLI

## **Introduction**

Memory Machine CE includes a CLI for interacting with the OpCenter. The OpCenter web interface provides a clickable button to download the CLI binary for a Linux, macOS, or Windows client. The CLI can also be used from a web console launched from the OpCenter web interface.

This part of the Memory Machine CE User Guide provides a detailed command reference for the CLI.

## CLI Command Reference

Use the `float` CLI commands to interact with the OpCenter; for example, to submit and manage jobs.

### Version

The Memory Machine CE CLI commands described here are consistent with the OpCenter release shown in the table.

Memory Machine CE CLI	OpCenter Release	Date
v1.2.1-885d2b4-Bondi.bin	v1.2.1-885d2b4-Bondi.bin	2023-02-01T06:49:07Z

### Global Flags

Use global flags with any `float` command or subcommand.

Flag	Usage	Definition
<code>-a, --address <a href="#">ip_address</a></code>	Connect to OpCenter server	IP address of OpCenter (default: localhost or last IP address used)
<code>--conf <a href="#">/path/to/folder</a></code>	Specify location of configuration folder	Location of configuration folder (default: " <code>~/user_home/.float</code> ")
<code>-F, --format <a href="#">json yaml table</a></code>	Specify format for output	Output format (default: yaml)
<code>-h, --help</code>	Display help	Help for Memory Machine CE CLI
<code>--logLevel <a href="#">log_level</a></code>	Specify log level	Log level (default: info)
<code>-p, --password <a href="#">password</a></code>	Log in to OpCenter	Login password
<code>-u, --username <a href="#">user_name</a></code>	Log in to OpCenter	Login username
<code>-v, --verbose <a href="#">on off</a></code>	Turn verbose mode on or off	Verbose mode setting (default: off)

### float completion

Use the `float completion` command to generate auto-completion script for use in current shell or in every shell started subsequently. Using a subcommand with the `-h` flag displays help on how to use the auto-completion script.

Subcommands	Usage	Option	Option Definition
<b>bash</b>	Generate auto-completion script for bash	--no-descriptions	Disable completion descriptions
<b>fish</b>	Generate auto-completion script for fish	--no-descriptions	Disable completion descriptions
<b>powershell</b>	Generate auto-completion script for powershell	--no-descriptions	Disable completion descriptions
<b>zsh</b>	Generate auto-completion script for zsh	--no-descriptions	Disable completion descriptions

## float config

Use the `float config` command to view or change the OpCenter configuration. Using a subcommand with the `-h` flag lists the options.

Subcommands	Usage	Option	Option Definition
<b>get</b>	Show runtime value of a single configuration parameter	<code>config_parameter</code>	Configuration parameter to display, e.g., <code>sessionTimeout</code>
<b>list, ls</b>	List configuration parameters for OpCenter (also shows if a parameter is changeable and if a change requires a system restart)	<code>-s, --scope filter_string</code>	String to filter the list of configuration parameters (default: list all parameters)
<b>set</b>	Set runtime value of a single configuration parameter	<code>config_parameter parameter_value</code>	Configuration parameter and the value to set it to

## Examples

- `float config get log.level`  
info
- `float config set sessionTimeout 1h`  
sessionTimeout is set to 1h, please restart OpCenter to take effect

## float group

Use the `float group` command to manage OpCenter user groups. Using a subcommand with the `-h` flag lists the options.

Subcommands	Usage	Option	Option Definition
<b>add, create</b>	Create a new group	<code>new_group</code>	Name of new group
		<code>--admin user1, user2...</code>	Username(s) of admins of new group
		<code>--user user1, user2...</code>	Username(s) added to new group
<b>delete, re-move, rm</b>	Delete a group	<code>group_name</code>   <code>group_id</code>	Name or id of group to delete
<b>info, show</b>	Display information about group including members	<code>group_name</code>   <code>group_id</code>	Name or id of group to query
<b>list, ls</b>	List groups that current user belongs to		
<b>update</b>	Update attributes of a group	<code>group_name</code>   <code>group_id</code>	Name or id of group to update
		<code>--add user1,user2...</code>	Username(s) added to group
		<code>--admin user1,user2...</code>	Username(s) given admin role in group
		<code>--name group_name</code>	New name for group
		<code>--remove user1,user2...</code>	Username(s) removed from group

### Example

```
float group add team --user tester
name: team
gid: 3
admins: ""
users: tester
```

## float image

Use the `float image` command to manage container images on OpCenter. Using a subcommand with the `-h` flag lists the options.

Subcom- mands	Usage	Option	Option Definition
<b>add</b>	Add image pull information to OpCenter	<code>image_name image_uri</code>	Image name and URI to pull image from repository. Do not use <code>image_uri</code> if <code>--link</code> specified.
		<code>--link file_link</code>	Link to access S3 or OSS file, e.g., <code>s3://bucket_name&gt;/file_path</code>  Do not use if <code>image_uri</code> specified.
		<code>--token repo_access_token</code>	Token to pull image from private repository
		<code>--user repo_access_user</code>	Username to pull image from private repository
<b>cache</b>	Cache image on resource configured for OpCenter (NFS folder or S3 bucket)	<code>image_name</code>	Container image to cache
		<code>-f, --force</code>	Overwrite existing cached version of image
		<code>--tag tag_string</code>	Tag to select specific container image (default: "latest")
<b>certify</b>	Validate that workload runs using OpCenter	<code>image_name</code>	Container image to validate
		<code>--bandwidth bandwidth</code>	Bandwidth (in Mbps) required for workload (only used in AliCloud)
		<code>-c, --cpu min_cpu:max_cpu</code>	Range of number of virtual CPUs to select from to run job (can omit <code>max_cpu</code> ).

<code>--customTag tag- Name:tagValue</code>	Tag customized by each Memory Machine CE subscriber, for example, to generate customized usage reports. Multiple custom tags can be applied by using <code>--customTag</code> multiple times in a single command.
<code>--dataVolume[size=vol_- size,throughput=t- put_rate]:/data_dir or vol_id:/container_mn- t_pt/path/to/dir or nfs://ip_ad- dress/server_export_- dir:/container_mnt_- pt/path/to/dir</code>	Data volume mounted by container.  <code>vol_size</code> in GB, <code>tput_rate</code> in Mbps (can omit), <code>data_dir</code> is data directory created, or  <code>vol_id</code> specifies existing EBS volume, <code>/container_mnt_-pt/path/to/dir</code> is path to where EBS volume is mounted on container, or  <code>ip_address</code> is IP address of NFS server, <code>/server_export_dir</code> is directory exported by NFS server, and <code>/container_mnt/pt/path/to/dir</code> is path to where NFS directory is mounted on container.
<code>-d, --def definition_- file</code>	Path to file if using definition file (yaml or json format) to provide input parameters to <code>certify</code> command.
<code>-e, --env env_key=env_- value</code>	Environment variable setting for the job
<code>--extraOptions extra_opt</code>	Container-specific string (enclose in quotes) passed to checkpoint process (needed for some containers)
<code>-f, --force</code>	Automatically answer "yes" at confirmation prompt
<code>--imageVolSize image_- vol_size</code>	EBS volume size in GB to load image (default: 6)

<code>-t, --instType</code> <code>instance_type</code>	VM instance type (e.g., c4.xlarge for AWS or ecs.g7.6xlarge for AliCloud). Do not combine with <code>--cpu</code> or <code>--mem</code> options.
<code>--interval</code> <code>job_interrupt_interval</code>	Interval between job interruptions in format <code>hh:mm:ss</code> where <b>h</b> , <b>m</b> , and <b>s</b> are hours, minutes, and seconds, respectively (default: 10m)
<code>-j, --job job_script</code>	Job script to run workload. Use format: path to local file, s3 file, OSS file or https   http file.
<code>--mem min_mem:max_mem</code>	Range of memory size (in GB) to select from to run job (can omit <code>max_mem</code> ).
<code>--metricsInterval metric_s_int</code>	Time in seconds between queries to obtain container metrics (default: 10s)
<code>--migratePolicy [migrate_policy]</code>	<p>Policy to determine auto-migration behavior. Format is <code>[option1=value2, option2=value2...]</code>. The available options with their default values are (if no units are attached, the value is a percentage):</p> <ul style="list-style-type: none"> <li>• <code>enable=false</code></li> <li>• <code>disable=true</code> (use <code>enable</code> or <code>disable</code>, not both)</li> <li>• <code>cpu.upperBoundRatio=90</code></li> <li>• <code>cpu.lowerBoundRatio=5</code></li> <li>• <code>cpu.upperBoundDuration=30s</code></li> <li>• <code>cpu.lowerBoundDuration=5m0s</code></li> <li>• <code>cpu.step=50</code></li> <li>• <code>mem.upperBoundRatio=90</code></li> <li>• <code>mem.lowerBoundRatio=5</code></li> </ul>

	<ul style="list-style-type: none"> <li>• mem.upperBoundDuration=30s</li> <li>• mem.lowerBoundDuration=5m0s</li> <li>• mem.step=50</li> </ul>
-n, --name <code>job_name</code>	Name to associate with job
-n, --noPublicIP	No public IP address assigned to container host (host only reachable via a private IP network)
-P, --publish <code>host_port:container_port</code>	Rule for publishing host port to container port, for example, <code>8080:80</code>
--rootVolSize <code>root_vol_size</code>	Root volume size in GB to load base OS (default: 40)
--securityGroup <code>sec_group</code>	Security group added to VM instance for this job. Use format <code>sg-xxxxx</code>
--shmSize <code>shm_size</code>	Size of <code>/dev/shm</code> in format <code>nu</code> where <code>n</code> is a number and <code>u</code> is b, k, m, or g for bytes, KiB, MiB or GiB, respectively (default: 64m)
--tag <code>tag_name</code>	Tag to select image version
--vmPolicy [ <code>vm_policy</code> ]	<p>VM creation policy. Use this format for [<code>vm_policy</code>] (numbers can be changed):</p> <p>[spotOnly=true,retryLimit=10,retryInterval=30s]</p> <p>[spotFirst=true,retryLimit=3,optimize=true]</p> <p>[onDemand=true]</p> <p>[spotOnly=true,priceLimit=0.1]</p>
-Z, --zone <code>availability_zone</code>	Availability zone in which to execute job

<b>delete, rm, remove</b>	Delete container image(s) and one or more tags	<b>image1 image2...</b> or <b>image1:tag1 image2:tag2...</b>	Container image(s) to delete. If no tags are specified, then all tags are removed.
		<b>-f, --force</b>	Automatically answer "yes" at confirmation prompt
		<b>--tag tag_name</b>	Tag associated with <b>image1 image2...</b> to remove. If this is the only tag associated with images, images are removed.
<b>list, ls</b>	Show all images available on OpCenter		
<b>tags, tag</b>	Display tags associated with image and image status	<b>image_name</b>	Container image name
<b>update</b>	Update information associated with image	<b>image_name</b>	Container image name
		<b>--addtag image_tag</b>	Additional tag to associate with image
		<b>--name image_name</b>	New name to identify image
		<b>--token repo_access_token</b>	New token required to access private repository
		<b>--user repo_access_user</b>	New username associated with token required to access private repository
<b>upload</b>	Load image from local server	<b>image_name</b>	Container image name
		<b>--path /path/to/image</b>	Path to image tar file on local server

**Examples**

- `float image list`

```
+-----+-----+-----+-----+
|  NAME  |          URI          | TAGS | ACCESS USER |
+-----+-----+-----+-----+
| python | docker.io/bitnami/python | 1 |             |
| r-base | docker.io/rocker/r-base  | 1 |             |
+-----+-----+-----+-----+
```

- `float image delete r-base`

Deleting r-base

- `float image add r-base docker.io/rocker/r-base`

```
name: r-base
uri: docker.io/rocker/r-base
owner: ""
accessUser: ""
accessToken: ""
tags:
  latest:
    status: Available
    uri: ""
    error: null
    lastUpdated: 2022-11-02T15:40:55.270169042Z
refCount: 0
```

## float log

Use the `float log` command to view and manage log files. Using a subcommand with the `-h` flag lists the options.

Subcom- mands	Usage	Option	Option Definition
<code>cat</code>	Write log file contents to standard output	<code>log_file</code>	Log file whose contents are displayed
		<code>-i, --hid host_id</code>	Host whose logs are displayed (default: OpCenter server)
		<code>-j, --job job_id</code>	Job whose logs are displayed (default: OpCenter)
<code>list, ls</code>	List all logs associated with target	<code>-i, --hid host_id</code>	Host whose log files are listed (default: OpCenter server)
		<code>-j, --job job_id</code>	Job whose log files are listed (default: OpCenter)

<b>rm</b>	Remove all logs associated with target	<code>-i, --hid <i>host_id</i></code>	Host whose log files are removed (default: OpCenter server)
		<code>-j, --job <i>job_id</i></code>	Job whose log files are removed (default: OpCenter)
<b>tail</b>	Write last <i>n</i> lines of log file to standard output	<code><i>log_file</i></code>	Log file to display
		<code>--follow</code>	Display new lines as they are appended to log file
		<code>-i, --hid <i>host_id</i></code>	Host whose log file lines are displayed (default: OpCenter server)
		<code>-j, --job <i>job_id</i></code>	Job whose log file lines are displayed (default: OpCenter)
		<code>-n, --num <i>n</i></code>	Number of lines to display (default: 100)

### Examples

```
float log tail --follow output -j XGiUDRto7kwofWBNPkiW5
Ready to prepare source data
Ready to download pbmc_1k_v3_fastqs from s3
Ready to download refdata-gex-GRCh38-2020-A from s3
Ready to run test
...[output edited]
```

```
float log ls
```

```
+-----+-----+-----+
| LOG NAME | SIZE | LAST UPDATE TIME |
+-----+-----+-----+
| etcd.log | 647667 | 2022-11-02T15:39:08Z |
| opcenter.access_log | 912282 | 2022-11-02T15:40:55Z |
| opcenter.log | 5678863 | 2022-11-02T15:51:24Z |
| upgrade.log | 500 | 2022-10-03T16:03:24Z |
+-----+-----+-----+
```

### float login

Use the `float login` command, with valid username and password, to log in to OpCenter. The command has no subcommands. Using the command with the `-h` flag lists the options.

Subcommands	Usage	Option	Option Definition
	Log in to OpCenter	--info	Display login status

## Examples

```
float login -u admin -p memverge -a 192.168.0.1
```

```
float login --info
address: 192.168.0.1
username: admin
role: admin
```

```
float login
Username: admin
Password:
Login Succeeded!
```

## float logout

Use the `float logout` command to log the current user out of the OpCenter and invalidate the authorization token. The command has no subcommands. Using the command with the `-h` flag lists the options.

## float migrate

Use the `float migrate` command to move a job from one VM instance to another VM instance of the same type or of a different type. The command has no subcommands. Using the command with the `-h` flag displays the options.

Subcom- mands	Usage	Option	Option Definition
	Migrate job to a new VM instance. If no options used, migrate to identical instance in same availability zone.	--cpu <code>min_cpu:max_cpu</code>	Range of number of virtual CPUs to select from for new VM instance (can omit <code>max_cpu</code> ).
		-f, --force	Automatically answer "yes" at confirmation prompt
		-t, --instType <code>instance_type</code>	VM instance type to migrate to (e.g., c4.xlarge in AWS or ecs.g7.6xlarge in AliCloud). Do



Modify certain attributes associated with running job	<code>--addSecurityGroup <b>sec_group</b></code>	Security group added to VM instance for this job. Use format <code>sg-xxxxx</code>
	<code>--extraOptions <b>extra_opt</b></code>	Container-specific string (enclose in quotes) passed to checkpoint process (needed for some containers)
	<code>--force</code>	Automatically answer "yes" at confirmation prompt
	<code>-j, --job <b>job_id</b></code>	Job to apply changes to
	<code>-M, --migratePolicy <b>mi-grate_policy</b></code>	New migrate policy to apply. See <code>float sbatch</code> for format.
	<code>--rmSecurityGroup <b>sec_group</b></code>	Security group to remove from VM instance for this job.
	<code>-V, --vmPolicy <b>vm_policy</b></code>	New VM creation policy to apply. See <code>float sbatch</code> for format.

## Example

```
float modify --vmPolicy [SpotOnly=true] -j PF0bgCvlpdJkog0RCBZPg
Warning: Are you sure you want to modify PF0bgCvlpdJkog0RCBZPg?
New vmPolicy may trigger migration.(yes/No): yes
Successfully modified PF0bgCvlpdJkog0RCBZPg: --vmPolicy [SpotOnly=true]
```

## float ps

Use the `float ps` command to show the complete process tree of a running job (the linux command `ps` must be installed in the container). The command has no subcommands. Using the command with the `-h` flag lists the options.

Subcommands	Usage	Option	Option Definition
	Show the complete process tree of job	<code>--args <b>podman_args</b></code>	Arguments passed to <code>podman</code>
		<code>-j, --job <b>&lt;job_id&gt;</b></code>	Job to query



Subcommands	Usage	Option	Option Definition
<code>download</code>	Download OpCenter usage history report	<code>--path /path/to/dir</code>	Path to save report on local machine
<code>get</code>	Generate and download usage report with filter(s) applied	<code>-A, --all</code>	Include all jobs (normal plus archived) in the report. Default: normal record archived after one hour of inactivity.
		<code>--filter filter1 filter2...</code>	Filters to apply to usage report. Example: status=executing timeRange=2010-10-22~
<code>list, ls</code>	List all available usage reports		

## Examples:

```
float report get -A usage_report_by_job -f status=Completed
```

```

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
-----+-----+
|          JOB ID          |          JOB NAME          | JOB STATUS |          LAUNCH TIME          |
JOB DURATION | NUMBER OF INSTANCES | INSTANCE COST | ONDEMAND INSTANCE COST | STORAGE
COST | NET SAVINGS |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
-----+-----+
| F4bM7dfzhotKvaVySxfYk | python-t3a.large          | Completed | 2023-01-19T19:38:03Z |
1h43m22s          |          1 | 0.0450 USD          | 0.1293 USD          |          0.0061
USD          | 0.0781 USD          |
...

```

```
float report download --path ./temp.gzip
```

```
Downloaded to ./temp.gzip
```

```
file temp.gzip
```

```
temp.gzip: gzip compressed data, original size modulo 2^32 14336
```

## float requeue

Use the `float requeue` command to re-submit a completed job or a job that failed to complete. The command has no subcommands. Using the command with the `-h` flag lists the options.

Subcommands	Usage	Option	Option Definition
-------------	-------	--------	-------------------

Re-submit completed or failed job	-j, --job <i>job_id</i>	Job to re-submit
--------------------------------------	-------------------------	------------------

## float restart

Use the `float restart` command to restart OpCenter terminates all login sessions). The command has no subcommands. Using the command with the `-h` flag lists the options.

Subcommands	Usage	Option	Option Definition
	Restart OpCenter	-f, --force	Automatically answer "yes" at confirmation prompt

## float sbatch (submit)

Use the `float sbatch` (same as `float submit`) command to submit a job. The command has no subcommands. Using the command with the `-h` flag lists the options.

Subcom- mands	Usage	Option	Option Definition
	Submit job for execution	--bandwidth <i>bandwidth</i>	Bandwidth (in Mbps) required for workload (only used in AliCloud)
		-c, --cpu <i>min_cpu</i> : <i>max_cpu</i>	Range of number of virtual CPUs to select from to run job (can omit <i>max_cpu</i> ).
		--customTag <i>tagName</i> : <i>tag-Value</i>	Tag customized by each Memory Machine CE subscriber, for example, to generate customized usage reports. Multiple custom tags can be applied by using <code>--customTag</code> multiple times in a single command.
		--dataVolume [ <i>size=vol_size,throughput=tput_rate</i> ]:/ <i>data_dir</i> or <i>vol-id</i> :/ <i>container_mnt_pt/path/to/dir</i> or	Data volume mounted by container.  <i>vol_size</i> in GB, <i>tput_rate</i> in Mbps (can omit), <i>data_dir</i> is data directory, or  <i>vol-id</i> is existing EBS volume, <i>/container_mnt_pt/path/</i>

<code>nfs://ip_address/server_export_dir:/container_mnt_pt/path/to/dir</code>	<code>to/dir</code> is path to where EBS volume is mounted on container, or <code>ip_address</code> is IP address of NFS server, <code>/server_export_dir</code> is directory exported by NFS server, and <code>/container_mnt_pt/path/to/dir</code> is path to where NFS directory is mounted on container.
<code>-d, --def definition_file</code>	Path to file if using definition file (yaml or json format) to provide input parameters to <code>sbatch</code> command
<code>-e, --env env_key=env_value</code>	Environment variable setting for the job
<code>--extraOptions extra_opt</code>	Container-specific string (enclose in quotes) passed to checkpoint process (needed for some containers)
<code>-f, --force</code>	Automatically answer "yes" at confirmation prompt
<code>-i, --image image_name</code>	Image to create container for job
<code>--imageVolSize image_vol_size</code>	EBS volume size in GB to load image (default: 6)
<code>-t, --instType instance_type</code>	VM instance type (e.g., c4.xlarge for AWS or ecs.g7.6xlarge for Al-iCloud). Do not combine with <code>--cpu</code> or <code>--mem</code> options.
<code>-j, --job job_script</code>	Job script to run workload. Use format: path to local file, s3 file, OSS file or https   http file.
<code>--mem min_mem:max_mem</code>	Range of memory size (in GB) to select from to run job (can omit <code>max_mem</code> ).

<code>--metricsInterval <b>metrics_</b> <b>int</b></code>	Time in seconds between queries to obtain container metrics (default: 10s)
<code>--migratePolicy [<b>migrate_</b> <b>policy</b>]</code>	<p>Policy to determine auto-migration behavior. Format is [<b>option1=value2, option2=value2...</b>] The available options with their default values are (if no units are attached, the value is a percentage):</p> <ul style="list-style-type: none"> <li>• enable=false</li> <li>• disable=true (use enable or disable, not both)</li> <li>• cpu.upperBoundRatio=90</li> <li>• cpu.lowerBoundRatio=5</li> <li>• cpu.upperBoundDuration=30s</li> <li>• cpu.lowerBoundDuration=5m0s</li> <li>• cpu.step=50</li> <li>• mem.upperBoundRatio=90</li> <li>• mem.lowerBoundRatio=5</li> <li>• mem.upperBoundDuration=30s</li> <li>• mem.lowerBoundDuration=5m0s</li> <li>• mem.step=50</li> </ul>
<code>-n, --name <b>job_name</b></code>	Name to associate with job
<code>-n, --noPublicIP</code>	No public IP address assigned to container host (host only reachable via a private IP network)
<code>-P, --publish <b>host_</b> <b>port:container_port</b></code>	Rule for publishing container port to container host port, for example, <b>8080:80</b>

<code>--rootVolSize <b>root_vol_size</b></code>	Root volume size in GB to load base OS (default: 40)
<code>--securityGroup <b>sec_group</b></code>	Security group added to VM instance for this job. Use format sg-xxxxx
<code>--shmSize <b>shm_size</b></code>	Size of <code>/dev/shm</code> in format <code>nu</code> where <code>n</code> is a number and <code>u</code> is b, k, m, or g for bytes, KiB, MiB or GiB, respectively (default: 64m)
<code>--tag <b>tag_name</b></code>	Tag to select image version
<code>--vmPolicy [<b>vm_policy</b>]</code>	VM creation policy. Use this format for [ <b>vm_policy</b> ] (numbers can be changed):  [spotOnly=true,retryLimit=10,retryInterval=30s]  [spotFirst=true,retryLimit=3,optimize=true]  [onDemand=true]  [spotOnly=true,priceLimit=0.1]
<code>-Z, --zone <b>availability_zone</b></code>	Availability zone in which to execute job

## Example

```
float sbatch -i tidyverse -j run_genericr.sh --cpu 4 --mem 8 --dataVolume [size=10]:/data
id: NDt428IsJtJZsB9WNUhGH
name: tidyverse-(worker)
workingHost: ""
usr: worker
status: submitted
submitTime: "2022-10-13T00:01:23Z"
```

## float scancel (cancel)

Use the `float scancel` (same as `float cancel`) command to cancel a job. The command has no subcommands. Using the command with the `-h` flag lists the options.

Subcommands	Usage	Option	Option Definition
-------------	-------	--------	-------------------

Cancel job	-f, --force	Automatically answer "yes" at confirmation prompt
	-j, --job <code>job_id</code>	Job to cancel

**Example**

```
float scancel -j KaSS49mJPomffUNhXrrCH
KaSS49mJPomffUNhXrrCH has been cancelled
```

**float show**

Use the `float show` command to display the status of a job or of a VM instance. The command has no subcommands. Using the command with the `-h` flag lists the options.

Subcommands	Usage	Option	Option Definition
	Show current status of job or VM instance	-i, --hid <code>host_id</code>	VM instance to query
		-j, --job <code>job_id</code>	Job to query
		--runningOnly	Display active jobs only. Used for Cromwell integration.

**Example**

```
float show -j FOIW5Y16KZgJ6Tsd02QuS
id: FOIW5Y16KZgJ6Tsd02QuS
name: cellranger-c5.12xlarge(worker)
workingHost: ""
usr: worker
output: 'container state: exited(0)'
status: completed
submitTime: "2022-10-03T19:17:01Z"
endTime: "2022-10-03T19:40:29Z"
.....[output edited]
```

**float sinfo (hosts)**

Use the `float sinfo` (same as `float hosts`) command to show current status of the cluster of worker nodes. The command has no subcommands. Using the command with the `-h` flag lists the options.

Subcommands	Usage	Option	Option Definition
	Show current cluster status	-A, --all	Show all jobs (normal plus archived). Default: normal

	record archived after one hour of inactivity.
<code>-f, --filter filter1 filter2...</code>	Filter(s) to apply to job status Example: status=executing timeRange=2010-10-22~

## float queue (list)

Use the `float queue` (same as `float list`) command to show a filtered list of queued jobs. The command has no subcommands. Using the command with the `-h` flag lists the options.

Subcommands	Usage	Option	Option Definition
	Show filtered list of queued jobs	<code>-A, --all</code>	Show all jobs (normal plus archived). Default: normal record archived after one hour of inactivity.
		<code>-f, --filter filter1 filter2...</code>	Filter(s) to apply to list of queued jobs. Example: status=executing timeRange=2010-10-22~

### Example

```
float queue -A --filter status=Completed
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+
| ID   | NAME | WORKING HOST | USER  | STATUS | SUBMIT TIME | DURATION | COST |
+-----+-----+-----+-----+-----+-----+-----+-----+
| FO...| cer...|               | worker| Completed| 2022-...   | 2022...   | 0.. |
...[edited for clarity]
```

## float status

Use the `float status` command to show current status of the OpCenter. The command has no subcommands. Using the command with the `-h` flag lists the options.

### Example

```
float status
ServerServer Status: normal
API Request Status: normal
License Status: valid
```

Init Time: "2022-10-05T07:28:11Z"  
 Up Time: "2022-11-11T01:25:57Z"

## float top

Use the `float top` command to show a sorted list of information (including CPU and memory utilization) about current jobs. The command has no subcommands. Using the command with the `-h` flag lists the options.

Subcommands	Usage	Option	Option Definition
	Show utilization and other information about running jobs	<code>--interval <i>top_int</i></code>	Time in seconds (3 or greater) between queries to obtain container metrics (default: 3s). Format is <i>ns</i> where <i>n</i> is integer.
		<code>-j, --job_id &lt;job_id&gt;</code>	Job to query

## float user

Use the `float user` command to manage OpCenter users (not available in web CLI console). Using a subcommand with the `-h` flag lists the options.

Subcommands	Usage	Option	Option Definition
<code>add, create</code>	Add a new user	<code><i>new_username</i></code>	Username for new user
		<code>--create</code>	Create group if the group does not exist (default: false)
		<code>--email <i>email_address</i></code>	Email address for new user
		<code>-g, --group <i>group_name</i></code>	Group associated with new user (default: "default")
		<code>--passwd <i>password</i></code>	Password for new user (default: "memverge")
<code>delete, remove, rm</code>	Delete a user	<code><i>user_name</i>   <i>user_id</i></code>	Username or id of user to delete
		<code>-f, --force</code>	Remove all active tokens belonging to user and delete user immediately
<code>info, show</code>	Display information about user	<code><i>user_name</i>   <i>user_id</i></code>	Username or id of user to query

<b>list, ls</b>	List groups that current user belongs to		
<b>passwd</b>	Reset user's password	<b>user_name</b>   <b>user_id</b>	Username or id of user to update
		<b>--passwd new_password</b>	New password for user
<b>update</b>	Update information associated with a user	<b>user_name</b>   <b>user_id</b>	Username or id of user to update
		<b>--email email_address</b>	New email address for user
		<b>-g, --group group_name</b>	New group associated with user
		<b>--name user_name</b>	New username for user
		<b>--passwd password</b>	New password for user

## Example

```
float user add tester --passwd secret123
username: tester
uid: 5
role: normal
group: default
email: ""
ownGroup: ""
```

```
float user info tester
username: tester
uid: 5
role: normal
group: default
email: ""
ownGroup: ""
```

## float version

Use the **float version** command to display the version of the **float** CLI client and the version of the OpCenter it is connected to. The command has no subcommands. Using the command with the **-h** flag lists the options.

## Example

```
float version
float version: v1.2.0-cb159df-Bondi
OpCenter version: v1.2.0-cb159df-Bondi
```