

User Agreements

Scientific Compute Platform(s)

Compute1 User Agreement

The RIS Compute Service is fundamentally built around *Docker Containers*. Docker is a *container platform* (see [Why Docker?](#) and [What is a Container?](#)) designed to make it easier to build and deploy *software runtime environments*. Users of the RIS Compute Service will be called upon to learn about Docker and its related concepts and technologies.

The Computing service is about *building and executing Docker container images*. While the RIS computing environment does offer ways to *build containers* (see Docker and the RIS Compute Service), it should be understood that users will oftentimes want their own computing environment, be that a Linux, MacOS, or Windows computing environment with which to build and work with Docker containers and Dockerfiles.

Furthermore, containers require a *container registry* to store the container images one builds. RIS does plan on offering a container registry service, but it is assumed that users will interact with public registries like Docker Hub.

The Compute service requires an understanding of these container technologies as well as a significant understanding of the Linux command line and related open source technologies, as well as *high performance computing job schedulers*.

Users of the service:

Agree to install Docker on their own computing workstation or laptop.

Agree to obtain accounts on Docker Hub or other public container registries.

Acknowledge that they will be learning Docker container technologies.

Acknowledge that they will be learning the Linux command line.

Acknowledge that they will be learning the IBM Spectrum LSF job scheduler.

Acknowledge that they will be using a *shared computing environment* and that their workloads may impact others.

Agree to be mindful of their workloads and strive to work with RIS if and when workloads negatively impact the cluster.

The use of Docker containers affords users the ability to run any software that can be built into a container. This is not fundamentally different than running arbitrary code downloaded from the Internet, which has been possible in any shared computing environment.

Users:

Acknowledge the risks of running code obtained from unverified sources.

Data Storage Platform(s)

Storage1 User Agreement

Use of Research Storage is primarily for research data which may include information that is classified as confidential and protected. Users of the storage who are unsure of the sensitivity of the data they intend to store should refer to the University's information classification policies (See the information classification policy.) or contact the information security office at infosec@wustl.edu for guidance.

Information in the protected class is required by agency regulation and university policy to be encrypted in transit and at rest. Sensitive information in the Research Storage is not to be removed to unprotected networks and computing resources. It is required to be encrypted if it is not in an approved university data center, on a mobile device or other computing system. See the encryption policy.

It is the responsibility of the storage user to ensure adequate protection of the information at all times when using this service.

Users of this service:

Agree to store only data that pertains to official business and is authorized to be stored within the service.

Agree to ensure that sensitive information stored within the service is restricted to authorized team members on a need-to-know basis.

Agree to ensure that access to sensitive data is based on your role or research.

Agree to not retrieve information for someone who does not have authorization to access that information.

Agree to ensure that Confidential and Protected information is protected against unauthorized access using encryption, according to Washington University Information Security Policy, when sending it via electronic means (telecommunications networks, e-mail, and/or facsimile) or storing it outside of protected networks (Note1) and devices (Note2). (See the encryption policy.)

Agree to coordinate your user access requirements, and user access parameters, with the Research Infrastructure Services (RIS) WashU IT group.

Agree to notify the service provider (RIS) if access to the storage resources is beyond that which you or they have authorized.

Agree to report all security incidents or suspected incidents to the RIS (ris@wustl.edu) and/or INFOSEC. (infosec@wustl.edu)

Agree to discontinue use of the service from any resources that show signs of being infected by a virus or other malware and report the suspected incident.

Agree to safeguard storage resources against waste, loss, abuse, unauthorized users, and misappropriation.

Agree to ensure that hard or electronic copies of Confidential and Protected information are destroyed after it is no longer needed. (See See the media reuse and disposal policy.)

Agree to not store U.S. classified national security information or Controlled Unclassified Information (CUI) on the service.

Agree to the monitoring of your use of this service for any violations of the above.

An unprotected network or networks with insufficient protection include any network other than WUCON or a High Trust Domain. Consult with the RIS or INFOSEC groups, if you do not know what network you are on or where the data will reside.

Any device that stores protected information and does not encrypt the information and does not have a password/passcode is considered unsafe and in violation of policy.

Policies

Compute1 Queues

General Queue Policies

Details

The general queue runs batch jobs much like the traditional HPC setting. They run in the background in the queue system.

The general queue also makes use of cache system, which you can learn more about [here](#).

Jobs in the general queue can run for up to 28 days.

Policies

The general queue falls under the fair use policy found in the [Compute1 User Agreement](#).

The general queue is for running jobs with large amounts of resource requirements.

The general queue is for running large numbers of jobs, especially the same analysis on multiple samples.

The general queue is NOT for GUI related software or interactive sessions.

General Interactive Queue

Details

The general-interactive queue runs jobs interactively so that you can interact directly with them or watch a job.

The general-interactive queue does not use the cache system and instead interfaces with the Storage Platform directly.

Jobs in the general-interactive queue can run for up to 24 hours.

Policies

The general-interactive queue falls under the fair use policy found in the [Compute1 User Agreement](#).

The general-interactive queue is for running interactive jobs.

The general-interactive queue is for GUI related software.

The general-interactive queue is for software and script development.

The general-interactive queue is NOT for jobs that require large amounts of resource requirements.

The general-interactive queue is NOT for multiple jobs running the same analysis on multiple samples.

Subscription Queue

Details

A subscription tier is associated with a number of resources that are guaranteed for use based on the tier.

There are currently three subscription tiers.

Tier 1 Resources

25 vCPUs

1 GPU

Tier 2 Resources

50 vCPUs

2 GPU

Tier 3 Resources

100 vCPUs

3 GPU

If you go over on the number of guaranteed vCPUs for a job submitted in this queue type, your job will not be guaranteed to run.

If you go over on the number of guaranteed GPUs for a job submitted in this queue type, the job will stay in pending and never run.

The -sla option is required for jobs submitted in this queue type.

Policies

The subscription queue falls under the [Compute1 User Agreement](#).

Usage policies are relegated by the owner of the subscription and is monitored by them.

Condo Queue

Details

A condo queue is a queue associated with a purchased condo.

More information about condos can be found here: <https://ris.wustl.edu/services/compute/compute-condo/>

The resources available in this queue are dependent on physical resources purchased as part of the condo.

Policies

The condo queue falls under the [Compute1 User Agreement](#).

Usage policies are relegated by the owner of the condo and is monitored by them.

Data

Compute Platforms

Compute1 \$HOME

\$HOME directories are limited to 10GB.

User \$HOME directories are intended to allow space for users to make use of the compute platforms, with the knowledge that the Storage Platforms is where data and software will be stored.

The \$HOME directory is required for the Compute Platform(s) to function for users and software often rely on it.

The `$HOME` directory is NOT backed up and important data should NOT be stored here. Anything you wish to be backed up should be placed in a Storage Platform location, this includes scripts.

Compute1 Cache (Storage1)

The cache provides an interface for the batch jobs in the general queue and for condos to the RIS Storage cluster.

The goal is to be completely transparent to the user. Users don't actively need to "move" data to or from the cache layer.

Cache file sets are the representation of the corresponding storage1 file set to the compute1 cluster.

Data is continuously synced between the cache and storage layers.

Before a file is finished syncing between cache to storage, it will appear in the compute environment but not when connecting to storage via SMB.

This can take time depending on the load on either storage layer.

Read more about how this here: [cache interfaces](#)

Only file reads and writes (not metadata requests) causes a file to be moved between layers.

A storage allocation maps to a cache space of the same name at `/storage1/fs1/${STORAGE_ALLOCATION}`.

Data remains in the cache layer until a **soft quota** is reached, at which time **cache eviction** is triggered and data is removed from the cache.

This quota is what triggers "eviction" of the cache file set, meaning files that have already been written back to the storage1 "home" file set are deleted.

Files that have not yet been written back to storage1 (aka "dirty" files) are not deleted.

A **hard quota** limits total capacity in the cache for a given allocation.

Compute1 Scratch

Scratch space is considered a "global temporary space". Users should not store their only copy of data in scratch.

Scratch space is available at `/scratch1/fs1/${COMPUTE_ALLOCATION}`.

Data must be manually put into `/scratch`.

The `setgid` bit is set at `/scratch1/fs1/${COMPUTE_ALLOCATION}`, all contents should have **group** set to `${COMPUTE_GROUP}`.

The `/scratch` space has quotas set based on compute-group membership (as opposed to fileset quotas, that are in place on `storage1`).

Your `${COMPUTE_GROUP}` will have a default **quota** of 5% of your storage allocation size, but no less than 1Tb.

Exceeding the **quota** will trigger a denial after which no further writes will be allowed until your `${COMPUTE_GROUP}` removes data and goes below the **quota**.

Files and directories will be automatically deleted from `/scratch` when their last modified date (`mtime`) is older than the LSF job run limit, currently 28 days.

Weekly usage reports are currently written to each scratch directory in a file called `RIS_Usage_Report.txt`

When files are deleted, the list of deleted files will be written to a scratch directory in a file called `RIS_cleanup_policy.{labname}`

Storage Platforms

Active

The client (login) nodes are connected directly to the RIS Storage1 Platform.

The general-interactive queue nodes are connected directly to the RIS Storage1 Platform.

These are both set up this way so that a user's interaction with their data is more responsive than what using the cache system (documented below) currently allows for.

All connections to the Storage2 Platform are direct connections.

There may be a slight delay (up to 5 seconds) of viewing changes made on one connection via another connection. (e.g. Between execution hosts)

Storage Platforms are NOT designed as a place to share office docs.

Archive

Archive storage is intended for long term data retention and NOT data that needs to be accessed regularly. Data moved to Archive resides on tape.

If data moved to Archive needs to be accessed again, it will need to be moved back to Active, and recalled from tape.

Data being moved to Archive should be compressed via tar.

Archive is NOT meant for data that is still being analyzed or will be analyzed within a few month's time.

Archive is a "cold storage", meaning it is for inactive data.

Recall times are dependent on the following.

Size of the data

How many tapes the data is spread across

How many other backup or recall jobs are running

How many tape drives are available.

It is not possible to predict how long a recall will take because of these factors.

Misc

Data Transfer

Supported methods to transfer data into and out of the Scientific Compute Platform are:

[Globus](#)

[Rclone](#)

Submitting a job on the Compute Platform(s) using a data transfer tool (e.g. `rsync`, `wget`, `curl`, `scp`)

Please do not use these data transfer tools directly on the compute1 client nodes. This type of activity can slow down the client node and negatively affect all users connected to the client nodes.

If you require assistance submitting jobs using these tools, please open a ticket at our [Service Desk](#).

Software Debugging

We strive to provide help with software debugging and support to the best of our abilities and time. With that being said, there may be times when we cannot solve an issue related to a specific piece of software or script that is not supported by RIS. In those cases, we will attempt to provide a solution to the problem, but we cannot guarantee that the solution will be successful. We recommend reading [this section](#) for more help debugging your software as well as for guidance on software development best practices.

Below are some links to information about debugging. Something necessary for software development.

<https://en.wikipedia.org/wiki/Debugging>

https://en.wikipedia.org/wiki/Rubber_duck_debugging

<https://jonskeet.uk/csharp/debugging.html>

<https://www.geeksforgeeks.org/debugging-tips-to-get-better-at-it/>

<https://blog.hartleybrody.com/debugging-code-beginner/>

Development Best Practices

Below are some links to what are considered some of the software best practices and should be kept in mind while developing.

<https://distantjob.com/blog/software-engineering-best-practices/>

<https://www.classicinformatics.com/blog/a-handbook-to-successful-software-development-practices>

<https://www.tiempodev.com/blog/software-development-best-practices/>

<https://www.techicy.com/best-practices-for-software-development-to-follow-in-2021.html>

File Name Best Practices

File names should be precise as NTFS file name size has a limit of 255 characters

This is a hard limit of the system that the Storage/Compute platform uses.

Any files to be transferred to Storage/Compute need to be created following this limit or they cannot be transferred.

Files should be named consistently.

File names should be short but descriptive.

Avoid special characters or spaces in a file name.

Use capitals and underscores instead of periods or spaces or slashes.

Use date format ISO 8601: YYYYMMDD.

Include a version number.

Write down naming convention in data management plan.

Elements to consider using in a naming convention are.

Date of creation

Short Description

Work

Location

Project name or number

Sample

Analysis

Version number