11falsenonelisttrue

Compute Resources

Have questions or need help with compute, including activation or issues? Follow this link.

User Agreement

Docker Usage

The information on this page assumes that you have a knowledge base of using Docker to create images and push them to a repository for use. If you need to review that information, please see the links below.

https://washu.atlassian.net/wiki/spaces/RUD/pages/1705115761/Docker+and+the+RIS+Compute1+Platform?atlOrigin=eyJpIjoiNzc4YTZjNjIxYmQwNGI3OTk4M2Q0MW

https://washu.atlassian.net/wiki/spaces/RUD/pages/1864892726/Docker+Basics+Building+Tagging+Pushing+A+Custom+Docker+Image?atlOrigin=eyJpIjoiMTVjMjNlM

storageN

The use of `storageN` within these documents indicates that any storage platform can be used.

Current available storage platforms:

storage1

storage2

Image Details

Docker image hosted at http://ghcr.io/washu-it-ris/rstudio

noVNC (https://novnc.com/info.html)

RStudio (https://rstudio.com/)

R (https://www.r-project.org/)

Optional: Seurat (https://satijalab.org/seurat/)

Initial Setup

R installs a default set of packages during installation. The list of installed packages can be viewed in the lower right pane of RStudio, with currently active packages indicated with a checkmark.

If you wish to install additional packages, you can do so by first creating a file in your home directory called `.Rprofile`, if it doesn't already exist. This file will be loaded automatically when RStudio starts.

touch ~/.Rprofile]]>

Next, create a folder to host your additional R packages. In the example below, the R packages will be stored in a folder named `R_libraries` in the Active folder of your storage allocation.

Storage Allocation Name:

Make sure to replace ${STORAGE_ALLOCATION} with the same name as the name of your storage allocation.

Backup an existing `.Rprofile` file if you have one.

Create a new `.Rprofile` file to store your additional R packages in storageN.

$HOME/.Rprofile vals <- paste('/storageN/fs1/${STORAGE_ALLOCATION}/Active/R_libraries/',paste(R.version$major,R.version$minor,sep="."),sep="") for (devlib in vals) { if (!file.exists(devlib)) dir.create(devlib) x <- .libPaths() x <- .libPaths(c(devlib,x)) } rm(x,vals) EOF]]>

Now that the `.Rprofile` file is created, you can install additional R packages using `install.packages()`. Please see https://www.rdocumentation.org/packages/utils/versions/3.6.2/topics/install.packages for more information.

Interactive GUI Session

Interactions GUI sessions are done via the `Custom noVNC Image` application in Open On Demand (OOD).

You can find out more about OOD here:
https://washu.atlassian.net/wiki/spaces/RUD/pages/1683489040/Compute1+Quickstart?atlOrigin=eyJpIjoiZTE1YmE4M2IyZDY3NDYzYzg0NWVhYzczZjYxYzMyYzYiL

There are two fields beyond the basics that will need information specific to this image.

Environment Variables

Docker Image

Environment Variables

This information should be space separated in the field.

Optional variables

GUI display size. This can be changed with the following variables.

Width default: 1024

Height default: 768

- DISPLAY_HEIGHT=]]>

Docker Image

]]>

RStudio Docker Tag:

The <tag> will refer to the version of RStudio in the Docker container. Please click here to see a current list of available RStudio versions and their corresponding Docker images.

Fill out the rest of the fields with the appropriate information (explained in the quick start).

Launch the job through the methods described in the quick start.

Once in an interactive RStudio session using the following command:

rstudio]]>

You should now see the RStudio GUI

You can safely ignore XDG_RUNTIME_DIR and Session version X does not match server version X warning that may appear when starting Rstudio.

Interactive Command-Line Session

If you wish to use R in an interactive command-line session, you can do so with the following commands.

)' /bin/bash]]>

Availability Through Open On Demand (OOD)

Interactive Application: RStudio

Start an instance of RStudio (interactive application: RStudio) in OOD.

RStudio will start at time of launch.

Interactive Application: Custom noVNC Image

Start an instance of Custom noVNC Image in OOD denoting the preferred RStudio image and tag.

Launch the session and enter rstudio in the terminal.

RStudio using THPC

Start an instance of THPC (interactive application: Compute RIS Desktop) in OOD.

Load and launch RStudio-server through the terminal.

After launching RStudio, you need to open a new shell.

Right click in the noVNC window.

Select “Application” > “Shell” > “Bash”

Finally, you need to load and launch Firefox, connecting to RStudio, in the new shell.

Extending the RStudio Docker Image

The steps in the initial setup will work for some but not all R packages. For example, `devtools` requires the following dependencies to be installed to the Docker image: `build-essential libcurl4-gnutls-dev libxml2-dev libssl-dev`.

This will require extending the existing Docker image to include these dependencies. Below is a sample Dockerfile that includes these dependencies. Please see this section for more information on setting the Docker tag in the Dockerfile.

Dockerfile Best Practices:

It is recommended to set the Docker tag to a specific R version. This will prevent using an updated version of the RIS RStudio image, which may have compatibility issues with R packages previously installed. As an example, in the Dockerfile below, the tag is set to `3.6.3`.

Please see our Docker Tutorial for more information on building and pushing a Docker image. You can also open a ticket at our Service Desk for further help.

Available RStudio Versions

Current Version:

Stand Alone Docker Image Version

ghcr.io/washu-it-ris/rstudio

4.4.0

THPC Version

4.2.1