Terminology Used By the Research Object Store System (ROSS)

ROSS, the Research Object Store System currently is implemented with a Dell EMC Elastic Cloud Storage (ECS) appliance. The ECS system interchangeably supports both the S3 and Swift protocols for object access, as well as the NFS and HDFS protocols for file-system-like access to objects kept in the store. If you really insist, we also have a way to support SMB through a gateway node (good for Windows-based file shares). Although we haven't benchmarked it, we suspect that SMB access would be slower than other styles of access due to the extra hop through the SMB gateway node. The ECS system also supports the EMC ATMOS and CAS protocols (from the EMC Centera world); however, the UAMS HPC management team is not actively supporting those protocols. Users who chose to utilize those protocols are on their own. Please consult the ECS Data Access Guide for details about the object access APIs supported by our object store. In most cases, the ECS Data Access Guide merely outlines the differences between the APIs as implemented in the ECS system and the reference APIs. Hence, a user may also need to consult the upstream API documentation to get more details about the protocols.

Some grasp of ECS terminology and concepts is useful for understanding how to interact with the ECS system. The ECS system divides the storage into **N amepspaces**. An **Object User** is assigned to a particular Namespace, which is the default Namespace used when no specific Namespace is identified in a RESTful API call. An Object User cannot be assigned to multiple Namespaces. A person who needs to be an Object User in multiple Namespaces with be given multiple Object User IDs, one for each Namespace. Each Namespace holds a set of **Buckets**, which in turn each hold a set of **Objects**. An Object User may create as many buckets with whatever characteristics they wish within the Namespace to which the Object User belongs. Each Bucket in a namespace belongs to a **Bucket Owner** in that Namespace, which by default is the Object User who created that Bucket. The Bucket Owner can set an **Access Control List (ACL)** on their Buckets that dictate which other Object Users may search or modify that Bucket, including who can create, read, update, or delete Objects held within that Bucket. The Bucket Owner can also set ACLs for each individual object, as desired, to allow other Object Users to access those objects.

It is possible for Object Users that belong to one Namespace to access Buckets and Objects in other Namespaces. To do so, the Object User must have been given permission (e.g. through an ACL attached to the Object or the Bucket in which the Object resides) by the owner of the Object. The Object User must also identify the other namespace in the RESTful API call, for example using the x-emc-namespace element in the HTTP header of the RESTful call.

For the object store at UAMS, each HPC user who requests personal space in the object store is given their own Namespace tied to their account on Grace. When referencing their account on Grace the HPC user must append "@hpc.uams.edu" to their Grace username. In other words, the name of HPC user's personal Namespace in the object store is their HPC username (the same as their home directory name on Grace), and the sole Object User within that namespace is identified as username@hpc.uams.edu.

Using their Grace login information in the form of username@hpc.uams.edu, a user may access the ECS Portal via a web browser, where the user has limited rights as a **Namespace Administrator** to manage their personal Namespace (e.g. create or replace access keys, create/delete Buckets, set Bucket attributes, set up NFS or HDFS shares on buckets, etc.). Please note that this GUI allows you to fetch your secret key for accessing S3, or to set your Swift password.

Departments, labs, or other entities may arrange to have shared group Namespaces in the object store. The department, lab, or other entity designates one or more persons to manage that Namespace. These group **Namespace Administrators** normally would access the ECS Portal using their UAMS credentials in the form of "username@uams.edu" (i.e. not their HPC credentials). Note that each UAMS user can only be the Namespace Administrator of one shared Namespace (not counting the user's personal Namespace tied to their HPC account on Grace). There is a workaround for this rule, but it requires multiple IDs, one for each Namespace. From the ECS Portal the Namespace Administrators can either add ACLs to Buckets or Objects to allow users to access data in the Namespace, can create local Object Users (Object Users who are not tied to any particular domain) within the Namespace, and can configure buckets for object users.

S3 Object Users can choose between path-based addressing of the object store or virtual host based addressing. Only the HTTPS protocols should be used to maintain security.

In S3 path-based addressing, the bucket name is the first path element in the URL. For example, to access a bucket named foo, the URL would be https://s3.ross.hpc.uams.edu/foo/. If the bucket is not in the Object User's namespace, the RESTful call must also include x-emc-namespace in the HTTPS heade r. By default, without the x-emc-namespace key/value pair in the HTTP header, ECS assumes that the Namespace that the bucket foo resides in is the Namespace to which the Object User belongs. Object Users can only belong to one Namespace. A particular person could have multiple Object User IDs if they work in multiple namespaces, and adding the header is inconvenient.

In S3 virtual host based addressing, the namespace and optionally the bucket can be prepended to the base host name of s3.ross.hpc.uams.edu. For example, to access a bucket named foo in a namespace named bar, the URL would be https://foo.bar.s3.ross.hpc.uams.edu or alternatively https://bar.s3.ross.hpc.uams.edu/foo/. With virtual host based addressing, there is no need to include the x-emc-namespace key/value pair in the header, since the Namepace is inferred from the virtual host name.

If using the Swift protocol, the base URL to access the API would be https://swift.ross.hpc.uams.edu/. The Namespace and the Bucket names are included in the path of the Swift API URL, as defined by the Swift API. The Namespace is equivalent to the <account> path component in the API descriptions in the ECS Data Access Guide. The Bucket is equivalent to <container>in the Swift API

The ECS Data Access Guide also outlines alternate ports for reaching the respective APIs. These ports are enabled on the https://ross.hpc.uams.edu. Th ere is no need to prepend the API type (e.g. s3 or swift) to the host name since the port takes care of identifying the API. For example, to access a Bucket named "foo" using the S3 protocol the URL would be https://ross.hpc.uams.edu:9021/foo. When using the alternate ports, only path-based S3 addressing is allowed.

A popular method for accessing the object store from the command line is the s3curl command. Since our object store is based on Dell EMC ECS technology, one should use the Dell EMC ECS variant of s3curl, found at https://github.com/EMCECS/s3curl.

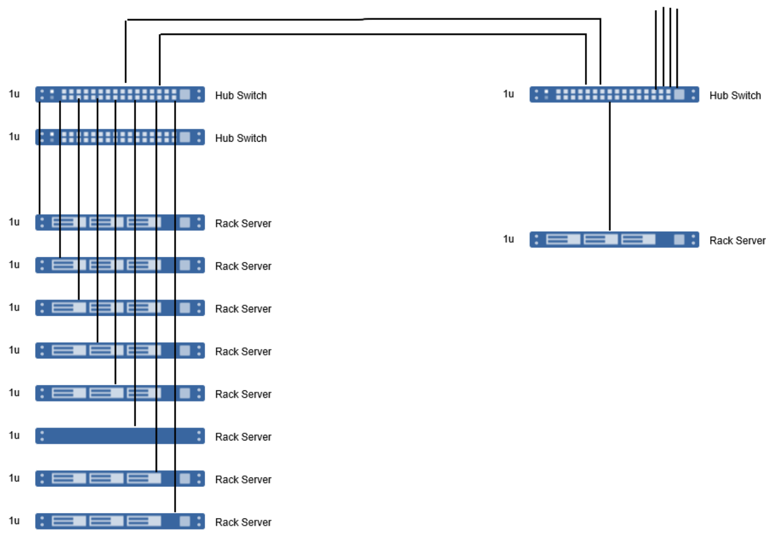
While s3curl is a commonly used tool, it does not offer great performance for accessing the object store. Access to the object store is parallel in nature. Many other tools leverage that inherent parallelism to improve performance. In our experience, the open-source rclone tool is one of the easiest to use, with its rsync-like syntax. The rclone tool also allow for mounting a bucket as if it were a directory-oriented file system, similar to an NFS or network share mount. Early benchmarking indicates that rclone seems to have better performance than NFS-mounting a bucket directly, particularly for reads.

An even faster tool, according to our early benchmarking, is ecs-sync.   The ecs-sync tool offers the best performance of any tool that we have tried thus far. However, it is a bit more difficult to use. We have set it up on a DTN node, to facilitate moving data between ROSS and Grace over higher bandwidth links, for better performance.

There are a lot of other options for accessing the object store, including both free and paid software. UAMS IT has a limited number of licenses for S3 Browser, which is a GUI-based Windows tool for accessing ROSS. We do not endorse any particular tool. Users may pick the tool that they feel most comfortable with.

A Namespace Administrator can also set up a Bucket with File Access enabled at Bucket creation time using the ECS Portal. After creating the file-enabled Bucket, the Namespace Administrator for the Namespace in which the Bucket resides can set up, for example, an NFS share using that Bucket along with user ID mapping. Users can then mount and access the bucket using the NFS protocol, similar to any other network file share. Enabling a bucket for file access does seem to add a tiny bit of overhead to that bucket, and does disable a few advanced features such as S3 object versioning. And accessing a bucket via NFS is slower than accessing using the object protocols (e.g. S3 or Swift). If an Object User needs file access, they might consider using rclone mounting instead of the NFS mounting, as it might be faster.

If a user desires high speed NFS storage, UAMS IT offers at cost a research NAS that the researcher may use. Please contact UAMS IT directly if you would like Research NAS storage.


1u
Hub Switch
1u
Hub Switch
1u
Rack Server
1u
Rack Server
1u
Rack Server
1u
Rack Server
1u
Rack Server
1u
Rack Server
1u
Rack Server
1u
Rack Server
1u
Hub Switch
1u
Rack Server