# Introduction

!!! warning “Draft Draft Draft”

This is a draft of the storage description, layout and recommended usage.  
Comments/suggestions/critiques to us in Slack #hpc-community.

# Storage Namespaces

The following table outlines the available storage namespaces in which storage is either already available or can be provisioned upon request.

| Unix Path | Disk/Server Redundancy | Snapshots | On-Site Replica | Off-Site Replica | AWS Glacier Deep Archive | Best used for/data which is: |
| --- | --- | --- | --- | --- | --- | --- |
| ${HOME} | High | Yes | Yes | No | No | * private data/applications |
| /hpc/mydata/${USER} | High | Yes | Yes | No | No | * personal working space |
| /local/scratch /hpc/nodes/${NODE} | Low | No | No | No | No | * scratch data * easily reproducible data * intermediate job files |
| /hpc/scratch/${NAME} | Low/Medium | No | No | No | No | * scratch data * easily reproducible data * intermediate job files |
| /hpc/projects/${NAME} | High | Yes [[1]](#footnote-21) | Yes | Optional [[2]](#footnote-22) | Optional [[3]](#footnote-23) | * working/active data sets * results * difficult to reproduce data |
| /hpc/archives/${NAME} | High | Yes [[4]](#footnote-24) | Yes | Optional [[5]](#footnote-25) | Optional [[6]](#footnote-26) | * completed projects * data with retention requirements |
| /hpc/user\_apps/${GROUP} | High | Yes [[7]](#footnote-27) | Yes | Yes [[8]](#footnote-28) | No | * shared application installations |
| /hpc/reference/${NAME} | High | Yes [[9]](#footnote-29) | Optional | Optional | No | * easy to re-download * easy to reproduce * write-once, read-many * storage optimized for reads |
| /hpc/instruments/${NAME} | High | Yes [[10]](#footnote-30) | Yes | Optional [[11]](#footnote-31) | Optional [[12]](#footnote-32) | * raw or preprocessed instrument data * directory per-instrument |
| /hpc/websites/${VHOST} | Variable | Optional | Optional | Optional | Optional | * static web content * requires a vhost name for website |

# Namespace Details

## ${HOME}

The user ${HOME} directory, also known as ~ and on Scientific Computing systems, always located at /home/${USER}, is for private user data. The system uses this location to store sensitive data, website caches/cookies, Kerberos ticket data, credentials, etc., and because of this should always have permissions of 0700 and ownership of ${USER}:${USER}.grp. This location should NEVER be shared with another account or user on the system. If there is a need for multiple people to share an account, please request a shared service account that is not a real person.

## /hpc/mydata/${USER}

This location is provided to give each user 1 TB of personal working space where they can relax permissions to share data or work on small collaborations without needed to request storage spaces. It’s intended to provide a space to try things out and once a project has matured, to then request project, archive or other explicit space for the project.

## /local/scratch, /hpc/nodes/${NODE}

!!! Warning

This space is subject to being purged at each node reboot.

Each node has some amount of local working storage. This can always be accessed via /tmp which is a per-user namespace only available when on the node. /local/scratch is available for use where a project might want to stage data which remains on the node across multiple jobs. /local/scratch on each node is then NFS exported to all other nodes and can be found at /hpc/nodes/${NODENAME}. Please consult with us before using this space as local storage is not treated as a consumable by Slurm and filling a node’s local disks can negatively impact other jobs.

## /hpc/scratch/${NAME}

!!! warning

NEVER LEAVE THE ONLY COPY OF IMPORTANT DATA ON A SCRATCH SPACE.

Shared scratch storage areas are available via this namespace. The locations will be tuned for maximum performance and the cost of reliability. Scratch spaces are subject to being lost as a result of very few disk failures or being purged and rebuilt on relatively short notice to address performance or space issues.

## /hpc/projects/${NAME}

Project spaces are intended to be good working spaces with high reliability. Snapshots, where possible, provide a history to help recover from mistaken deletes or changes, but as noted above the amount of retained history is subject to the usage pattern of the project. The level of replication/backup and long term archiving can be adjusted as needed for a given projects requirements.

## /hpc/archives/${NAME}

Archive spaces are intended to be write-once and then never or extremely rarely updated, until they reach a pre-determined end-of-life and are removed. The underlying on-prem/site storage will be optimized for reliability and read performance and this is a good location to park data which needs to be made available via the web or [Globus](https://globus.org) on a long term basis.

## /hpc/user\_apps/${GROUP}

This location provides an optimized location for installing shared applications. It is organized by groups, with each PI/lab group having a location where they can install applications, with permissions to share those as widely as need. Optionally a module can be added so that applications here can be managed with module load appname/version after doing module load ${GROUP}.

## /hpc/reference/${NAME}

The reference location is provided as a place to store on-site copies of public or other reference data to avoid re-downloading the same datasets repeatedly. Many NCBI datasets fall into this category, where it makes sense to manage one copy everyone can access and use. Only things easy to reproduce or re-download should be kept here. This space is optimized for reading to support being used in scale-out workflows that need to access reference data.

## /hpc/instruments/${NAME}

Storing data from instruments presents some special problems. It’s common to have an instrument controlled by a Windows PC which cannot be updated due to vendor/support restrictions and is at increased risk for malware or ransomware. To limit the impact of an infected instrument PC, each instrument can have a separate service account such that a given instrument can only reach/modify its own data. Additionally, by having this space accept data but be treated as read-only to all other systems, it’s possible to maintain a long snapshot history to enable recovering from a potential ransomware infection with minimal data loss.

## /hpc/websites/${VHOST}

It’s often convenient to be able to produce static website content from the HPC environment and make it available via a public website. The most basic use of this location is to create a hostname, say myproject.czbiohub.org, then any data written to /hpc/websites/myproject.czbiohub.org will be available via <https://myproject.czbiohub.org>. Additional features, like authentication and access control can be made available upon request.

# Example Use Cases

## Analysis Pipeline

Suppose we have a pipeline, managed by a workflow tool like nextflow. The incoming raw data for this pipeline is generated from one or more lab instruments. The pipeline processing will produce

* intermediate/scratch files
* quality control results to be web-accessible
* processed data made available for researcher secondary analysis via local filesystem and globus

The steps in our hypothetical data flow might be:

1. Data moves from instrument(s) to /hpc/instruments/${INSTRUMENT\_NAME}(s). Can be push or pull.
2. Pipeline copies data to /hpc/scratch/${NAME} as a working location.
3. HPC Cluster jobs perform analysis:
   * Reference data read from /hpc/reference/${DBNAME}
   * Running jobs use /tmp and/or /local/scratch and/or /hpc/scratch/${NAME} for intermediate and temporary files.
   * Results written to /hpc/projects/${NAME}
4. QC output formatted and written to /hpc/websites/${NAME}.czbiohub.org
5. Data to be retained long-term copied to /hpc/archives/%{NAME}

This example uses shared and local scratch, instrument storage, projects storage, website storage and archive storage. As all these are also available (or can be) via [Globus](https://globus.org), any of these spaces can be optionally used for data delivery to or sharing with collaborators.

1. Snapshot polices will vary based on space constraints and the underlying storage snapshot features. Dynamic workflow patterns, with frequent writes/re-writes/deletes, typically requires reducing snapshot lifespan. [↑](#footnote-ref-21)
2. Off-site replicas available on a per-project basis, subject to constraints on space and available inter-site bandwidth. [↑](#footnote-ref-22)
3. Long-term archiving a copy of data to Deep Archive is available upon request. [↑](#footnote-ref-23)
4. Snapshot polices will vary based on space constraints and the underlying storage snapshot features. Dynamic workflow patterns, with frequent writes/re-writes/deletes, typically requires reducing snapshot lifespan. [↑](#footnote-ref-24)
5. Off-site replicas available on a per-project basis, subject to constraints on space and available inter-site bandwidth. [↑](#footnote-ref-25)
6. Long-term archiving a copy of data to Deep Archive is available upon request. [↑](#footnote-ref-26)
7. Snapshot polices will vary based on space constraints and the underlying storage snapshot features. Dynamic workflow patterns, with frequent writes/re-writes/deletes, typically requires reducing snapshot lifespan. [↑](#footnote-ref-27)
8. /hpc/user\_apps may be replicated read-only and require a specific host or hosts for writes. [↑](#footnote-ref-28)
9. Snapshot polices will vary based on space constraints and the underlying storage snapshot features. Dynamic workflow patterns, with frequent writes/re-writes/deletes, typically requires reducing snapshot lifespan. [↑](#footnote-ref-29)
10. Snapshot polices will vary based on space constraints and the underlying storage snapshot features. Dynamic workflow patterns, with frequent writes/re-writes/deletes, typically requires reducing snapshot lifespan. [↑](#footnote-ref-30)
11. Off-site replicas available on a per-project basis, subject to constraints on space and available inter-site bandwidth. [↑](#footnote-ref-31)
12. Long-term archiving a copy of data to Deep Archive is available upon request. [↑](#footnote-ref-32)