Discoverer Use Policy (DUP)

Operational and technical rules for the Discoverer supercomputer

Version 1.0 · 8 June 2026

Related documents:

- Discoverer Access Policy
- Discoverer Acceptable Use Policy (AUP)

Veselin Kolev
v.kolev@discoverer.bg

Discoverer Petascale supercomputer
Sofia Tech Park, Sofia, Bulgaria
https://discoverer.bg
https://docs.discoverer.bg

Contents

Preface

Scope of the document

This document describes operational and technical rules for using the Discoverer petascale supercomputer at SofiaTech Park, Sofia, Bulgaria. The infrastructure comprises two clusters: Discoverer (CPU cluster) and Discoverer+ (GPU cluster). This policy covers both clusters, including associated storage, software stacks, and support services provided by PetaSC.

PetaSC is the consortium that operates the Discoverer petascale supercomputer — both the Discoverer CPU cluster and the Discoverer+ GPU cluster — including shared storage, access systems, scheduling, and user support (see Appendix B).

The Discoverer Use Policy (DUP) is binding on all End Users with active access. It does not replace the Access Policy (which governs allocation) or the Acceptable Use Policy (AUP) (which governs lawful and ethical conduct).

Relationship to other policies

Per the Access Policy (§3.8.2), detailed rules for backup, retention, and related operational matters are specified here and in the AUP. Users must comply with all three documents and with the operational documentation published at https://docs.discoverer.bg (§1.4).

1. Introduction

1.1 Purpose

This policy gives End Users practical rules for running workloads efficiently, safely, and fairly on shared infrastructure. It describes how Discoverer and Discoverer+ are operated, how resources are requested and accounted for, and how storage, software, and support services are to be used.

1.2 Audience

This document is written for Principal Investigators (PIs), project members, commercial customers, and technical contacts who submit and run jobs on Discoverer or Discoverer+. PetaSC operational and support staff use it as the baseline for day-to-day service delivery.

1.3 Document conventions

Numeric limits (walltimes, quotas, partition names, and file paths) may be updated in the documentation at docs.discoverer.bg or in call documentation without republishing the full DUP, provided the underlying policy rule is unchanged. Where a section refers to "published limits", the current values in that documentation or in onboarding material take precedence over examples in this document. Appendix D lists the main documentation pages that correspond to DUP sections.

1.4 Operational documentation

By accessing Discoverer or Discoverer+, every End User is obliged to follow the procedures, limits, and instructions in the documentation published at https://docs.discoverer.bg. That site is the authoritative source for day-to-day operational detail (login steps, module names, partition settings, example job scripts, and similar) that may change more frequently than this policy. Where the documentation and this DUP address the same topic, both apply; where only the documentation gives current numeric limits or hostnames, the documentation controls.

2. System overview

2.1 Architecture summary

PetaSC is the consortium that operates the Discoverer petascale supercomputer. The service is delivered as two separate clusters:

Cluster	Role	Scheduler partitions (examples)
Discoverer	CPU cluster — primary home for CPU-only and large-scale parallel workloads	`cn`
Discoverer+	GPU cluster — primary home for GPU-accelerated workloads	`common`

Each cluster has its own login nodes and compute nodes. Login nodes are shared servers with limited resources; they must not be used as personal workstations (§11.2). Per-user resource shaping applies (§11.3). Jobs are submitted from the login nodes of the cluster where the target partition resides; compute nodes are not shared across the two clusters.

Both clusters draw user identities from the same LDAP directory (see §3.6 and §3.7). Storage, interconnect details, and node specifications are described in the sections below and in the Resource Overview on docs.discoverer.bg.

2.2 Discoverer (CPU cluster)

Discoverer is the CPU cluster. CPU compute work is run on partitions such as `cn`. Typical workloads include MPI and OpenMP applications, large serial or embarrassingly parallel batches, and memory-intensive simulations that do not require accelerators.

Detailed node counts, processor type, memory per node, and interconnect specifications are published in the Resource Overview and may be updated when the system is upgraded.

2.3 Discoverer+ (GPU cluster)

Discoverer+ is the GPU cluster. GPU compute work is run on partitions such as `common`. Typical workloads include CUDA-accelerated machine learning, GPU molecular dynamics, and other accelerator-bound applications. GPU nodes are equipped with NVIDIA H200 accelerators (or successor hardware as documented in the user guide).

Detailed GPU count per node, CPU companion cores, memory, and interconnect specifications are published in the Resource Overview.

2.4 Naming and environment modules

Software is managed with Lmod environment modules on both clusters. Users select compilers, MPI stacks, CUDA, and application frameworks with `module avail`, `module load`, and `module purge`. Site defaults are documented under Software. GPU jobs on Discoverer+ typically require loading a CUDA module (for example `nvidia/cuda/12.6`) compatible with the installed driver before launching work.

2.5 Shared vs dedicated access

Most access modes provide time-shared use of Discoverer and Discoverer+ through the SLURM scheduler. Benchmark access and certain call conditions may grant short exclusive or full-system windows; such arrangements are stated in the access award or call documentation and take precedence over routine shared scheduling.

3. Accounts, projects, and allocations

3.1 Account types

The following account categories are used

- personal LDAP/POSIX login — one per End User, with a unique UID and primary GID (§3.6), used for SSH and home-directory ownership;
- SLURM project account — one per awarded project, identified by `-account`; the account name follows the accepted project ID; tied to the allocation and used for accounting; account names, definitions, and allocation sizes are visible to all cluster users and are not anonymised (§3.8, §4.7);
- project POSIX group — created in LDAP at project onboarding; owns the project storage directory; members are added per §3.7;
- QoS — quality-of-service class identified by `-qos`; scheduling rules differ for non-commercial and commercial projects (§3.8, §4.4);
- service accounts — non-interactive LDAP accounts usually provisioned for federation purposes (for example identity or resource federation with external services); other uses require explicit PetaSC approval;

3.2 Linkage to access awards

Every job must be submitted with the project's SLURM account and an authorised QoS combination issued at onboarding (§3.8). These identifiers link consumption to the access award described in the Access Policy. When an allocation expires or is closed, SLURM submission rights for that account are disabled; the SLURM account itself may nevertheless be retained for accounting purposes (§14.5). Project and home data remain subject to retention rules in §8.4 and §8.5 until removed or exported.

3.3 Principal Investigator responsibilities

The PI (or commercial customer technical lead) is responsible for

- maintaining an accurate list of team members authorised to use the project account;
- ensuring all members read and comply with the Access Policy, AUP, this DUP, and the documentation at docs.discoverer.bg;
- distributing compute use within the awarded limits;
- ensuring project members agree internally on how to utilise the fixed project storage quota (§7.2); PetaSC does not arbitrate internal sharing within a project;

- requesting roster changes through PetaSC when collaborators join or leave, so project POSIX group membership stays accurate (§3.7);
- ensuring stored data and running workloads remain within approved project scope (see §7.6);
- arranging export of project-directory data within 30 days after the project end date (§8.4);
- notifying PetaSC when collaborators leave so roster and retention rules (§8.5) apply correctly.

3.4 Allocation units

Resource consumption is tracked in units appropriate to each service

- CPU time — typically core-hours or node-hours on Discoverer partitions;
- GPU time — GPU-hours on Discoverer+ partitions;
- storage — allocated capacity on `/valhalla` (Lustre group quota), optional `/weka` (folder quota), and home directories (ext4 UID quota), including inode limits where applicable (§7.2);
- support — staff effort where separately contracted.

Exact conversion factors and reporting views are described in Computational resources allocation and accounting and the project onboarding letter (Onboarding guide).

3.5 Extensions and renewals

Extensions and renewals of compute access follow the Access Policy process for the relevant access mode. PetaSC notifies PIs before allocation expiry where possible. A successfully renewed project may receive a new award period with SLURM settings recorded in the renewal decision.

Storage quotas are fixed at project onboarding from the storage requested in the approved project application (§7.2). Quota extensions or renegotiation during the project lifetime are not accepted. If a renewal award includes a new storage entitlement, quotas are set again at onboarding for that renewal; mid-project increases are not granted on request.

Users should plan data export before expiry if renewal is uncertain.

3.6 Identity directory, POSIX users, and cluster access

All End Users are provisioned in a single LDAP directory server shared by Discoverer and Discoverer+. The directory is the source of POSIX identities used on both clusters: usernames, numeric UIDs, primary GIDs, supplementary group memberships, and SSH public keys are consistent everywhere the account appears.

User creation

When an account is created, PetaSC staff assign

- a POSIX username (login name);
- a POSIX name for the user's primary (default) group, assigned together with the username;
- a unique POSIX UID;
- a unique POSIX GID for that primary group.

Username and primary-group naming

By default, the POSIX username and primary group name are derived from the Given Name and Family Name supplied at onboarding:

- start with the first letter of the Given Name;
- append the Family Name converted to ASCII lowercase (non-ASCII characters in names are transliterated or normalised to ASCII as part of provisioning).

If that combination is already taken, PetaSC staff extend the name by adding further letters from the Given Name before the family name, or by appending a numeric suffix at the end, until a unique name is found.

Anonymised or random POSIX usernames (and matching primary group names) are available only if requested during onboarding. PetaSC does not rename accounts to anonymised or alternative names after onboarding; the assigned username remains fixed for the lifetime of the LDAP record (§3.6, account retention).

The user's home directory is owned by that POSIX username with the user's primary POSIX GID as the owning group. Files created in `$HOME` normally inherit this ownership unless the user changes it deliberately.

Each LDAP account record also holds the following personal data

- Given Name;
- Family Name;
- e-mail address.

These data are collected during onboarding, kept current for operational contact (support, security notices, allocation correspondence), and used together with the POSIX username and PetaSC-administered SSH public keys to administer access (key changes require approval; see §11.1). A change to Given Name or Family Name after onboarding does not change the assigned POSIX username.

Cluster access control

Cluster access is not automatic for every provisioned user. PetaSC assigns LDAP group membership according to the active access award, access mode, and project affiliation. Separate LDAP groups may control entitlement to Discoverer, Discoverer+, or both, in addition to project groups described in §3.7. Together with SLURM account and QoS settings, membership determines which login nodes, partitions, and allocations a user may use.

In summary

- one LDAP/POSIX account per End User, visible on both clusters;
- each user receives a distinct UID and primary GID at creation;
- LDAP group membership defines cluster entitlement and project affiliation;
- PetaSC adds or removes memberships when access is granted, changed, or revoked;
- attempting to use a cluster or project storage without the corresponding membership is a policy violation (see the AUP §5.4 and §7.1).

Account retention after access ends

When a user is no longer a member of any active project, PetaSC disables cluster access (LDAP group memberships for Discoverer and Discoverer+, VPN where applicable, and SLURM use). Even then:

- the POSIX account record remains in the LDAP database;
- the assigned UID and primary GID numbers remain reserved to that username and are not reassigned to another person.

Disabling access therefore removes entitlement to log in and use clusters; it does not delete the directory identity or recycle numeric IDs. Home-directory data may still be removed under §8.5 after the grace period. PetaSC may disable SSH keys or login capability while retaining the LDAP entry for audit, federation, or possible future reactivation.

The Operational Manager (OM) is responsible for directory provisioning in line with decisions from the Business Development Manager (BDM) and project onboarding records.

3.7 Project POSIX groups and project storage ownership

When a project is onboarded, PetaSC creates a dedicated POSIX group in LDAP for that project. The project POSIX group owns the project storage directory on `/valhalla` (for example `/valhalla/projects/<account>/`) and, when Weka is allocated, the corresponding folder on `/weka`.

To join a project, a user's POSIX username is added as a member of the project's POSIX group. Removing a user from the group revokes shared access to the project directories

while leaving the user's home directory and UID/GID unchanged. The PI requests roster changes through PetaSC; the OM applies additions and removals in LDAP.

All files and directories under the project paths should remain group-accessible so that Lustre group quota on `/valhalla` and folder quota on `/weka` accrue correctly to the project.

Project group membership is independent of, but usually issued together with, SLURM project account access and cluster entitlement groups (§3.6). A user must hold the project POSIX group membership to work with that project's stored data even if they have cluster login access.

3.8 SLURM project account and QoS provisioning

At project onboarding, PetaSC provisions exactly one SLURM account per accepted project. The SLURM account name is derived from the accepted project ID recorded in the allocation (the same identifier is typically used in project storage paths such as `/valhalla/projects/<project-id>/`).

Non-commercial projects

For open research and innovation and other non-commercial access modes, PetaSC provides equal access to cluster resources across projects:

- each project receives one default QoS for production work;
- PetaSC does not assign QoS definitions that give one non-commercial project systematically higher scheduler priority than another;
- pending jobs from different non-commercial projects compete on a FIFO-like basis at a common priority tier (see §4.4).

Supplementary QoS names may still be attached to a non-commercial project when needed, but only to express job limits (for example maximum walltime, node count, or GPU count for debug or test jobs), not to grant preferential queue priority over other non-commercial projects.

Commercial projects

Commercial access may use separate QoS and scheduling arrangements defined in the commercial contract. Those arrangements do not change the equal-access rule for non-commercial projects.

Documentation

PetaSC documents QoS names, limits, and intended use in the onboarding letter and in Accounting and Computational resources allocation and accounting. Users may use

only QoS values attached to their project account. Requests for extra QoS during the award period are evaluated by PetaSC but are not granted routinely; new QoS is normally defined at onboarding or at renewal.

After the project end date, the SLURM account is disabled for new submissions but may remain in the accounting system so that historical resource usage totals are preserved (§14.5).

4. Job submission and scheduling

4.1 Workload manager

Discoverer and Discoverer+ use SLURM for job submission and scheduling. Users interact through standard commands including `sbatch`, `srun`, `salloc`, `scancel`, and `squeue`. Batch scripts are the preferred submission method for production runs; interactive allocations should be short-lived (see §11.4).

4.2 Partitions and quality-of-service (QoS)

Each project submits jobs with its single SLURM account (§3.8) and an authorised QoS for that project.

Cluster	Primary partition	Non-commercial QoS
Discoverer	`cn`	default production QoS; supplementary QoS for limit variants only
Discoverer+	`common`	default production QoS; supplementary QoS for limit variants only

Additional partitions or QoS levels may exist for maintenance, testing, or special calls. Only partitions and QoS values issued to a project may be used. Commercial projects follow contract-specific QoS where applicable.

4.3 Job limits and defaults

Published limits include maximum walltime per job, maximum nodes or GPUs per job, concurrent job count, and array task size. Defaults and ceilings are set per QoS and may differ between access modes or between production and debug QoS on the same project. Current values are listed in Computational resources allocation and accounting and Running jobs, and enforced by SLURM.

4.4 Scheduling policy and queue priority

Non-commercial projects

PetaSC schedules non-commercial workloads so that no project receives standing preferential priority over other non-commercial projects. All such projects share equal access to cluster resources in the sense that:

- no QoS is used to make one non-commercial project's jobs routinely outrank another's in the pending queue;

- when resources become available, eligible pending jobs from non-commercial projects are ordered FIFO-like at a common priority level, subject only to partition fit, resource request size, and valid limits;
- historical usage does not increase or decrease a non-commercial project's relative priority versus other non-commercial projects.

Backfill may still start smaller jobs when idle resources would otherwise be wasted, provided those jobs do not bypass the equal-access principle for non-commercial work. Preemption applies only where explicitly enabled for operational or commercial arrangements documented separately.

Commercial projects

Commercial workloads may be scheduled under different QoS or priority rules specified in the commercial contract. Those rules apply only to the commercial accounts they cover.

4.5 Required job metadata

Every production job must specify at minimum

- `-account` — the project's SLURM account (project ID; §3.8);
- `-partition` — target partition on the correct cluster;
- `-qos` — the project's authorised QoS (default production QoS for non-commercial projects unless a documented limit-variant QoS applies);
- output and error paths — typically `-o` and `-e` under project storage;
- meaningful `-job-name` for operational traceability.

GPU jobs on Discoverer+ must request GPUs explicitly (for example `-gres=gpu:1`).

4.6 Prohibited scheduling practices

Users must not

- submit to partitions or clusters for which the project is not authorised;
- flood the scheduler with excessive pending jobs or array tasks;
- hold resources idle without workload;
- circumvent limits by splitting work across many accounts without approval (each project has one SLURM account; §3.8).

Such behaviour may be treated as misuse under the Access Policy and AUP.

4.7 Visibility of SLURM accounts and usage

PetaSC does not anonymise SLURM usage. SLURM project accounts (`-account` values), their definitions, allocation sizes, and consumption figures are visible to all users with

cluster access through standard commands and interfaces (for example `sacct`, `sshare`, `squeue`, and `scontrol show account`).

Any user can inspect which accounts exist, how they are configured, and how much resource each account has used or been granted, subject to the permissions built into the SLURM reporting tools. This transparency supports shared operational awareness. It is separate from optional anonymised POSIX usernames at LDAP onboarding (§3.6): even where a login name is anonymised, jobs and accounting remain attributable through SLURM account names, Unix usernames on jobs, and scheduler records.

Users must not attempt to obscure SLURM accounting by mis-tagging jobs with another project's account without authorisation.

4.8 SLURM accounting database access

SLURM accounting is backed by a SQL database operated by PetaSC. Users must query accounting and usage data only through the supported SLURM client tools and interfaces documented in Accounting.

The following are prohibited

- actively scanning or probing the SLURM SQL accounting database outside approved interfaces;
- issuing useless, repetitive, or high-volume queries that impose substantial load on that database;
- automated harvesting or enumeration of accounting records beyond normal operational needs.

PetaSC reserves the right to ban users who engage in such behaviour, including temporary or permanent suspension of cluster access, without prior notice where immediate protection of the accounting service is required. Related sanctions appear in the AUP (§5.6, §11.3).

5. Discoverer (CPU cluster) partition usage

Workloads submitted to partition `cn` run on Discoverer compute nodes. Submit these jobs from a Discoverer login node (SSH via VPN; see §11.1).

5.1 Intended workloads

Discoverer CPU nodes are intended for tightly coupled MPI jobs, OpenMP or threaded applications, high-throughput task farms, and memory-bound scientific codes that do not require GPUs.

5.2 Resource requests

Request CPU resources explicitly in batch scripts, for example

- `-cpus-per-task` — cores for each task;
- `-ntasks` / `-ntasks-per-node` — for MPI parallelism;
- `-mem` — memory required per node or per CPU;
- `-time` — upper bound on run duration.

Avoid over-requesting memory or cores, as unused resources reduce scheduler efficiency.

5.3 Performance and scalability

Users should validate scalability with benchmark runs before large campaigns. I/O-heavy jobs should use project storage (§7) rather than login nodes. Thread and process binding behaviour follows SLURM defaults unless the application documentation requires explicit binding flags; see NUMA and SLURM guide for CPU topology on Discoverer.

5.4 Example batch scripts

Canonical CPU submission patterns, including module load sequences, project paths under `/valhalla/projects/<account>/`, and `TMPDIR` placement, are maintained in Organizing your Slurm batch scripts and Running jobs.

6. Discoverer+ (GPU cluster) partition usage

Workloads submitted to partition `common` run on Discoverer+ compute nodes. Submit these jobs from a Discoverer+ login node (direct SSH; see §11.1).

6.1 Intended workloads

Discoverer+ GPU nodes are intended for CUDA (or supported GPU framework) applications such as deep learning training and inference, GPU-accelerated simulation, and large-scale linear algebra on accelerators.

6.2 Resource requests

GPU jobs must request accelerators explicitly, for example `-gres=gpu:1` (or a higher count when authorised). Request sufficient CPU cores and memory per GPU as documented in Running jobs and ML/AI on Discoverer. Multi-GPU jobs must request all GPUs in a single job unless a documented workflow requires otherwise.

6.3 GPU binding and visibility

SLURM sets `CUDA_VISIBLE_DEVICES` (or equivalent) for allocated GPUs. Users must not attempt to access GPUs not assigned to their job. Multi-Instance GPU (MIG) partitioning, if enabled, is configured by PetaSC and documented separately.

6.4 Software stack

GPU nodes run NVIDIA drivers and CUDA toolkits exposed through environment modules. Users must match application builds to the supported CUDA/driver combination published for Discoverer+. Containerised GPU workflows are subject to §12.

6.5 Example batch scripts

Canonical GPU submission patterns for partition `common`, including CUDA module load and project storage paths, are maintained in Organizing your Slurm batch scripts and ML/AI on Discoverer.

7. Storage and filesystems

7.1 Filesystem layout

Shared storage is available from both clusters. PetaSC operates more than one backend; quota rules depend on the filesystem (§7.2).

Area	Path pattern (example)	Backend	POSIX ownership	Purpose
Home	user home directory (`$HOME`)	ext4	user : primary GID	personal settings, small files
Project (Lustre)	`/valhalla/projects/<account>/`	LustreFS on `/valhalla`	project POSIX group	primary project data, software environments, job I/O
Project (Weka)	project folder on `/weka`	WekaFS	project POSIX group	high-performance project I/O where allocated

Exact mount points, Weka folder paths, performance characteristics, and availability may differ between Discoverer and Discoverer+ login paths; current paths are listed in Storage, Home folder, and Per-project scratch and storage folder.

A project may receive storage on `/valhalla` only, on `/weka` only, or on both, depending on the access award. Weka space is provisioned only when explicitly included in the allocation.

7.2 Quotas and inode limits

PetaSC enforces capacity limits on all provisioned storage. Inode limits apply on ext4 and LustreFS; Weka does not count inodes, so no inode quota is enforced there.

Home directories (ext4)

Home-directory quota is managed on the underlying ext4 filesystem. Usage is charged against the UID that owns the files — the End User's POSIX UID — regardless of group membership elsewhere. Capacity and inode limits apply per user home as published in the allocation or default site policy.

Project storage on `/valhalla` (LustreFS)

On `/valhalla`, project quota is implemented as a LustreFS group quota. The limit

applies to the project's POSIX group: all files attributed to that group on the Lustre filesystem count towards the project's capacity and inode allowances, consistent with Lustre project-quota semantics.

Project storage on `/weka` (WekaFS)

When a project receives a Weka allocation, PetaSC assigns a dedicated folder on `/weka` and enforces quota on the data stored in that folder (and its subtree). Capacity limits follow the awarded Weka storage; inode counts are not tracked on Weka and are therefore not limited by PetaSC on that filesystem.

Project storage quotas at onboarding

Storage capacity for `/valhalla` and, where applicable, `/weka` is set when the project is onboarded. The quotas provisioned or confirmed with the PI at onboarding must match the storage requirements stated in the approved project application (or commercial contract). PetaSC records those values as the project's fixed storage entitlement for the award period.

Further negotiation or extension of project storage quotas after onboarding is not accepted. Projects must stay within the allocated capacity and inode limits until the award ends. If usage approaches the limit, the PI and project members must free space or reprioritise internally; PetaSC will not raise project quotas on request.

Internal use of project storage

All members of a project share one project quota on each allocated filesystem. The PI and project members must agree among themselves how to use that shared capacity (for example split by subdirectory, workflow stage, or team member). PetaSC does not mediate internal allocation of space within the project quota.

Reporting

Quota usage is reported through filesystem tools and monitoring provided at onboarding. Users approaching limits should reorganise or delete data before jobs fail with "disk quota exceeded" or equivalent errors. See Calculating the disk usage basics and Move and copy files and folders. Home-directory quota on ext4 follows the per-UID rules above and is separate from project storage.

7.3 Performance guidance

Large parallel I/O jobs should use the filesystem best matched to the workload (`/valhalla` Lustre for general project storage, `/weka` where allocated for low-latency I/O). On Lustre and ext4, avoid creating very large numbers of small files where possible, because inode

quotas apply (§7.2). Weka does not enforce inode limits, but extremely fragmented namespaces may still perform poorly. Checkpoint at intervals compatible with §8.1. Striping and tuning parameters are documented in Storage and Per-project scratch and storage folder.

7.4 Shared project directories

Each project storage directory is owned by the project POSIX group created at onboarding (§3.7). Users who are members of that group can access the directory according to filesystem permissions set by PetaSC (typically group read/write/execute on directories and group read/write on files, with setgid where configured so new files remain group-accessible).

The PI is responsible for ensuring the team respects the fixed project quota (§7.2) and has agreed internally how shared space is used. PetaSC treats all usage under the project group as counting against the single project quota regardless of which member wrote the data.

Users must not change ownership of the project root or weaken permissions in a way that exposes data to users outside the project group. Data must remain readable and writable by the project group unless the PI explicitly accepts a different arrangement for a defined publication or export step.

7.5 Temporary and local storage

Batch jobs should set `TMPDIR` to a location under project storage (for example `/valhalla/projects/<account>/tmp`) rather than relying on node `/tmp` for large temporary files. Node-local storage, where present, is ephemeral and not backed up.

7.6 Acceptable stored content and PetaSC enforcement

Data stored on Discoverer filesystems must relate to the purpose for which access was granted under the approved project, contract, or access mode. Users must not store content that violates applicable law or compromises system security (including malware, exploit tooling, unauthorised credential stores, or material intended to attack PetaSC infrastructure or other tenants).

PetaSC reserves the right to restrict or cease access to storage — including disabling read/write access to specific directories, quarantining datasets, or removing data — where stored content does not match the purpose of the provided access, violates applicable law, or poses a risk to system security. Enforcement actions may be taken without prior notice where immediate action is necessary to protect the platform or comply with law. Where

practicable, PetaSC will notify the PI before or promptly after such action. Related grounds for sanctions appear in the AUP (§5, §11).

8. Data backup and retention

8.1 User responsibility for backup

Per the Access Policy (§3.8.2), users must perform a full task backup at least every 48 hours while work is active. A full task backup means preserving everything required to restart or recover the scientific task after failure, including checkpoints, critical output datasets, configuration, and source or workflow state needed to continue from the last consistent point.

8.2 Platform backup scope

PetaSC does not provide unlimited archival backup of user datasets unless explicitly agreed in writing. High-performance filesystems are designed for active project use, not long-term archival. Users remain responsible for copying data they wish to retain to external storage.

8.3 Compensation on hardware failure

If a hardware or platform failure causes demonstrable loss of work, PetaSC may compensate the affected project with replacement compute time up to a maximum equivalent of 48 node-hours (or the GPU-hour equivalent documented for Discoverer+), subject to verification by the Operational Manager. Compensation does not cover loss attributable to failure to meet the backup obligation in §8.1.

8.4 Project directory retention after project end

Each project has 30 days after the project end date to transfer any data stored in the project directory (for example under `/valhalla/projects/<account>/`).

The project end date is the date on which the access award or contract expires, or the date PetaSC formally closes the project, whichever is recorded in the allocation records. PetaSC notifies the PI of the end date through onboarding correspondence or renewal notices where possible.

During the 30-day period, users may retain read access to project storage even if SLURM submission is already disabled, unless PetaSC must revoke access earlier for security or policy reasons. After 30 days, PetaSC may delete the project directory and its contents without further notice. The PI is responsible for ensuring export completes within this window.

8.5 Home directory retention when project membership ends

Home directories (`$HOME`) hold personal settings and files not stored under a project path. They are tied to the LDAP account, not to a single project. SSH login keys are not managed through `~/.ssh/authorized_keys` in the home directory (see §11.1).

When an End User no longer belongs to any valid project that has access to the clusters, a 30-day grace period begins. A valid project is one with an active access award or contract and current cluster entitlement. During the grace period, the user may still log in if PetaSC has left login access enabled solely to export home-directory data.

If, after 30 days, the user still does not belong to any valid project with cluster access, PetaSC may delete the contents of the home directory without further notice. Cluster login and project access remain disabled; the LDAP POSIX account, username, UID, and reserved primary GID stay in the directory as described in §3.6 and are not reused for another user.

Users who move from one project to another before the grace period expires are not subject to home-directory deletion while they remain on at least one valid project roster.

8.6 Extended retention requests

Longer retention on platform storage for project or home data must be agreed with PetaSC before the applicable retention window in §8.4 or §8.5 expires. Approval depends on capacity, access mode, and commercial terms where applicable.

8.7 Deletion and secure wipe

PetaSC deletes data at end of retention using standard filesystem deletion procedures. Users with GDPR erasure requests relating to account personal data should contact PetaSC under the AUP (§8.1). Commercial customers with confidentiality requirements should agree handling procedures at contract stage.

9. Software environment and modules

9.1 Environment modules (Lmod)

Both clusters provide Lmod. Users run `module avail` to list software, `module load <module>` to activate a stack, and `module purge` before loading conflicting versions. Login and batch scripts should record loaded modules for reproducibility. See Software.

9.2 Centrally installed software

PetaSC maintains compilers, MPI libraries, mathematical libraries, and common scientific and machine-learning packages as modules. The installed set evolves with user demand and licensing constraints. Compiler and library stacks are listed under Software and Compilers.

9.3 Requesting new software

Requests for additional centrally installed software go to the support desk. The Business Development Manager (BDM) software team evaluates feasibility, licensing, and maintenance cost. Commercial users requiring licensed third-party products must disclose licensing needs at onboarding.

9.4 User installations

Users may install software under project storage (for example conda or Python virtual environments under `/valhalla/projects/<account>/virt_envs/`; see Conda on CPU and Conda on GPU). System-wide installation with `sudo`, modification of module trees without authorisation, and `pip install -user` into home directories at large scale are discouraged; use project-local prefixes instead.

10. Data transfer and external connectivity

10.1 Inbound and outbound data movement

Users transfer data with standard tools such as `scp`, `rsync`, and `sftp` from login nodes after authenticating as described in §11.1. Additional endpoints (for example Globus, object storage gateways, or dedicated transfer nodes) are documented in Data transfer when available.

10.2 Rate limits and acceptable transfer patterns

Large transfers should not saturate login nodes or interfere with other users. Long-running transfers belong in batch jobs or on dedicated transfer services where provided. Concurrent many-small-file syncs may be throttled.

The VPN tunnel required for Discoverer login access (§11.1) is an administrative access control; it is not intended as the primary channel for large dataset movement. Users moving substantial data to or from Discoverer should follow the procedures in §10.1 once logged in.

10.3 Publication and open-data requirements

Access modes that require open publication (see Access Policy Table 1) may also require a data management plan at application time. Users must honour those commitments and cite Discoverer resources as described in the AUP (§10.1).

10.4 Acceptable network use on cluster nodes

Login nodes and compute nodes may initiate network connections only for legitimate project work. The following are prohibited on login nodes, compute nodes, and any other Discoverer-managed hosts reachable from user workloads:

- network scanning, port scanning, or probing of PetaSC systems or external hosts;
- soliciting, spamming, or unsolicited bulk contact over the network;
- attacks or attack preparation, including denial-of-service, exploitation, credential stuffing, or command-and-control activity;
- downloading, storing, or distributing content that violates applicable law (including illegal material);
- running unauthorised listening services or open proxies.

Legitimate scientific use includes transferring research datasets, accessing approved exter-

nal archives, software repositories, and collaboration endpoints required by the project. Users must not use the cluster network as a general-purpose internet browsing or file-sharing platform unrelated to approved work.

Monitoring and enforcement

PetaSC reserves the right to monitor network use on the cluster infrastructure and to analyse traffic metadata and payloads where necessary to protect the platform, enforce this policy, and comply with law. Monitoring is conducted to the extent permitted by applicable law and operational need.

Users who breach fair use of the cluster network may be sanctioned under the AUP (§5.4, §7.2, §11.3), including warning, job termination, access suspension, or permanent ban.

11. Login and development nodes

11.1 Access to login nodes

Login nodes are shared servers with limited local resources (CPU, memory, and I/O). They are the only interactive entry point to each cluster. They are not personal workstations: users must not treat them as a private desktop or long-running development environment. PetaSC shapes use of local resources per user account so that no single session can monopolise a login node; limits are described in §11.3.

Access is restricted as follows.

Authentication

- the only permitted access method to login nodes is the SSH protocol;
- authentication is based on SSH public keys held and administered by PetaSC, not on keys placed by users in `~/.ssh/authorized_keys` on login nodes;
- password authentication is disabled and is not available on login nodes.

SSH public keys

PetaSC registers SSH public keys as part of onboarding and stores them in the central identity and access system (together with LDAP attributes; see §3.6). Login nodes authenticate against that registry. For storage at rest in that system, PetaSC keeps SSH public keys as cryptographic hashes only, not as recoverable public key material, as described in Quantum-resistant LDAP secure store of public OpenSSH keys. That limits the value of a later breach to an adversary planning harvest-now-decrypt-later attacks against public-key systems when cryptographically relevant quantum computers become available.

Users cannot add, change, or remove their own login keys by editing `~/.ssh/authorized_keys` or equivalent files on Discoverer systems; such files are not used for interactive login authorisation.

To add a new key, rotate a compromised key, or remove an old key, the user must submit a request through the support channel and receive PetaSC approval. The Operational Manager (or delegate) applies approved changes. Until a replacement is provisioned, the previously registered key remains the only accepted credential.

Users must protect the private key that corresponds to each PetaSC-registered public key and must not share private keys.

SSH cryptography

PetaSC reserves the right to change SSH protocol settings on login nodes and related services at any time, including:

- strengthening or replacing cipher suites;
- changing accepted public-key algorithms on the server side;
- updating hash functions used in SSH authentication or in stored key material.

Such changes may require users to update SSH client configuration, upgrade client software, or submit new public keys if older algorithms are retired. PetaSC will announce material changes through SSH Security, docs.discoverer.bg, support notices, or e-mail where practicable. Users are responsible for maintaining SSH clients that meet the then-current server requirements.

Network path (differs by cluster)

Cluster	Login access path
Discoverer (CPU)	SSH to the Discoverer login node through a previously established VPN tunnel to the PetaSC network
Discoverer+ (GPU)	Direct SSH to the Discoverer+ login node (no VPN required)

Users must complete VPN setup and receive VPN credentials before first login to Discoverer. Login hostnames, VPN configuration, and SSH key registration procedures are published in the Onboarding guide, Access (VPN, SSH Access), and Login nodes.

Authorisation

- a valid SSH key alone does not grant cluster access;
- the LDAP account must hold group membership authorising access to the target cluster (see §3.6);
- users entitled to both clusters use the same LDAP username and SSH key but follow the appropriate network path for each cluster.

PetaSC may publish canonical login hostnames and VPN instructions separately; host-

names may change during maintenance without altering the access rules above.

11.2 Appropriate use

Login nodes are shared servers intended only for short interactive tasks: editing source code, preparing batch scripts, lightweight compilation smoke tests, file management, and job submission. Users must not use them as personal workstations (for example persistent GUI sessions, long-running builds, local training jobs, or storing personal tooling unrelated to active project work). Long-running compute, large-scale training, or production data processing must run in SLURM jobs on compute nodes. Network use on login nodes must comply with §10.4.

Because local resources are limited and shared, users must keep interactive work within the per-account shaping limits in §11.3.

11.3 Resource shaping and limits on login nodes

PetaSC applies per-user-account resource shaping on login nodes. Each LDAP/POSIX account receives enforced ceilings on local CPU, memory, process count, and related server resources so that login nodes remain usable for all connected users.

Shaping mechanisms may include cgroups, PAM limits, session policies, or equivalent controls documented in Login nodes, Resource limits on login.discoverer.bg, and Resource limits on login-plus.discoverer.bg. Published thresholds may change without amending this DUP.

Processes or sessions that exceed the shaped limits for an account may be throttled or terminated without warning. Idle sessions may be disconnected after the published timeout. Repeated attempts to bypass shaping may be treated as misuse under the AUP.

11.4 Interactive and debug jobs

Short interactive work uses `salloc` or equivalent to obtain compute-node resources. See Submitting, monitoring, and canceling jobs. Debug or test QoS may be available with reduced resource limits but not with elevated priority over other non-commercial projects (§4.4). Interactive GPU sessions must be requested on Discoverer+ with explicit `-gres` allocation.

12. Containers and user-built software

12.1 Container policy

Where Apptainer/Singularity (or successor rootless container runtimes) is provided, users may run containers built without root on compute nodes. GPU containers must be executed through SLURM GPU allocations on Discoverer+. Users are responsible for the contents of images they execute; container images stored on project filesystems are subject to §7.6.

12.2 Building software on compute nodes

Software builds that require significant CPU or time should run as batch jobs on compute nodes, not on login nodes. Small configuration or compile checks on login nodes must stay within §11.3 limits. See Where and how to compile code.

13. Quality of service and support

13.1 Service levels

PetaSC targets high availability of compute and storage services during published service hours. Planned maintenance is announced in advance where possible. The support desk operates on at least an 8×5 basis (09:00–17:00 Eastern European Time) as stated in the Access Policy (§3.8).

13.2 Support channels

Users contact support through the ticket system and e-mail address published on the Discoverer website and in Getting help. Operational incidents escalate to the Operational Manager (OM); allocation and onboarding questions escalate to the Business Development Manager (BDM).

13.3 Training and documentation

PetaSC publishes operational guides, tutorials, and reference material at https://docs.discoverer.bg. All users are obliged to follow that documentation when using the clusters (§1.4). EuroHPC Competence Centres may offer additional local support for application porting and industrial outreach.

13.4 Application porting and optimisation

Basic guidance is included in standard support. Extended porting or optimisation may be available under separate agreements or specific access modes (Development, Benchmark) as described in the Access Policy. General guidance appears in Optimisation guides and Compatibility guides.

14. Accounting, monitoring, and fair use

14.1 Usage accounting

SLURM accounting records CPU, GPU, and walltime consumption per project account. PIs may inspect usage with `sacct`, `sshare`, and reporting tools documented in Accounting. Commercial customers receive consumption reports aligned with billing cycles; see SLURM GPU billing where applicable.

Accounting data are not anonymised: SLURM account identifiers, limits, and usage summaries are visible cluster-wide as described in §4.7. Users should assume their job submissions and project consumption can be viewed by other authenticated users. Direct abuse of the SLURM SQL accounting database is prohibited (§4.8).

14.2 Under-use of allocations

Significant under-use of awarded compute time without justification may be treated as misuse under the Access Policy (§3.2.4). PetaSC may contact PIs to review utilisation for Regular access allocations.

14.3 Oversubscription

Calls may allocate more total core-hours or GPU-hours than physically available in a period, relying on statistical multiplexing. Oversubscription can increase queue wait times but does not entitle users to exceed individual project limits.

14.4 Commercial metering and billing

Commercial access is metered separately for computation, storage, network, and optional support services. Pricing follows Access Policy §3.6.1 and individual contracts.

14.5 SLURM account and job record retention

PetaSC reserves the right to retain SLURM project accounts after a project is no longer active (submission disabled, allocation expired or closed) in order to preserve aggregated resource usage figures for that project ID.

Retention of per-job detail is limited separately

- PetaSC may delete individual job records from SLURM accounting six months after the project expiration date;

- aggregated account-level usage totals for reporting, audit, and historical reference may be kept longer while the SLURM account record remains;
- the project expiration date is the same date used for storage retention in §8.4.

PIs who require job-level accounting exports for final reports should extract them before the six-month window closes. PetaSC is not obliged to restore deleted per-job records.

15. Maintenance, outages, and disaster recovery

15.1 Planned maintenance

Planned maintenance windows are announced through the Discoverer website, e-mail lists, or ticket system. Running jobs may be drained according to published SLURM policies before nodes are taken offline.

15.2 Unplanned outages

During unplanned outages, PetaSC updates users through the same channels. Jobs interrupted by platform failure may be eligible for requeue or compensation as described in §8.3.

15.3 Disaster recovery

Users must not rely on platform storage as their sole copy of critical data. Recovery time objectives for infrastructure services are internal operational targets; project-level recovery depends on user backups (§8.1).

16. Changes to this policy

Amendments are approved by the Discoverer Management Team (DMT) and published on the Discoverer website. Material changes affecting active projects are communicated to PIs by e-mail or support notice. The document history appendix records versions.

17. Contact and documentation

- Operational documentation (mandatory for all users): https://docs.discoverer.bg — section index in Appendix D
- Website: https://discoverer.bg
- Access Policy: discoverer-access-policy.pdf
- Acceptable Use Policy (AUP): https://docs.discoverer.bg/aup.pdf
- Support: Getting help and contact details on the Discoverer website

Commercial access enquiries follow the commercial section of the Discoverer website.

Appendices

Appendix A — Quick reference tables

Cluster	Resource	Partition	Login access	Key SLURM directives
Discoverer	CPU	`cn`	SSH via VPN	`-partition=cn`, `-cpus-per-task`, `-mem`
Discoverer+	GPU	`common`	Direct SSH	`-partition=common`, `-gres=gpu:N`

Filesystem	Example path	Quota basis	Inode limit
Home (ext4)	`$HOME`	UID owner	yes
Project Lustre	`/valhalla/projects/<account>/`	project/group quota	yes
Project Weka	`/weka/...` (project folder)	folder contents	no

Retention: project paths — export within 30 days after project end (§8.4); home — deletion 30 days after user leaves all valid projects (§8.5).

Publication acknowledgement (minimum)

> This work used resources of the Discoverer supercomputer hosted by PetaSC at SofiaTech Park, Sofia, Bulgaria, co-funded by EuroHPC JU and the Bulgarian government.

Adjust wording to match EuroHPC and call-specific requirements in the AUP (§10.1) and the published Acknowledgements page.

Appendix B — Glossary

PetaSC — the consortium that operates the Discoverer petascale supercomputer at SofiaTech Park, comprising the Discoverer (CPU) cluster and the Discoverer+ (GPU) cluster, together with shared storage, identity and access systems, scheduling, and user support. Juridical representation and consortium composition are described in the Access Policy.

Other terms align with the Access Policy, AUP, and DUP §3.6–§3.8. Key abbreviations: DMT (Discoverer Management Team), OM (Operational Manager), BDM (Business Development Manager), PI (Principal Investigator), QoS (quality of service), UID/GID (POSIX user and group identifiers).

Appendix C — Document history

Version	Date	Author	Summary of changes
1.0	8 June 2026	Veselin Kolev <v.kolev@discoverer.bg> on behalf of PetaSC support team	Published: two-cluster model, LDAP/POSIX identity, SLURM accounting, storage and retention, login and SSH policy, cross-links to docs.discoverer.bg (Appendix D)

Appendix D — Operational documentation index

The site https://docs.discoverer.bg is the authoritative source for day-to-day operational detail (§1.4). The table below maps DUP sections to the main pages on that site. Application-specific software guides, SSH client how-tos, and model-specific ML pages are listed under Software and ML/AI on Discoverer rather than here.

DUP section	Topic	Documentation page
§2	System architecture and hardware	Resource Overview
§3.4, §3.8	Allocations, SLURM accounts, QoS	Computational resources allocation and accounting, Accounting
§3.6, §3.7	Onboarding, LDAP identity, project storage	Onboarding guide, Per-project scratch and storage folder
§4	Job submission, limits, scheduling	Running jobs, Organizing your Slurm batch scripts, Submitting, monitoring, and canceling jobs
§4.8, §14	SLURM accounting and billing	Accounting, SLURM GPU billing
§5–§6	CPU and GPU partition usage, examples	Writing Slurm batch scripts, NUMA and SLURM guide, ML/AI on Discoverer
§7	Storage layout, quotas, I/O	Storage, Home folder, Per-project scratch and storage folder, Calculating the disk usage basics, Move and copy files and folders
§9	Software modules and user installs	Software, Compilers, Conda on CPU, Conda on GPU
§10	Data transfer	Data transfer
§11	Login access, SSH, VPN, shaping	Access, Login nodes, VPN, SSH Access, SSH Security, Quantum-resistant LDAP secure store of public OpenSSH keys, Using SK keys on login-plus.discoverer.bg, Resource limits on login.discoverer.bg, Resource limits on login-plus.discoverer.bg
§12	Compilation and builds	Where and how to compile code
§13	Support, porting, optimisation	Getting help, Optimisation guides, Compatibility guides
Appendix A	Publication acknowledgements	Acknowledgements

Upstream process pages (not day-to-day cluster operation): Apply for access.

Policy-only topics with no dedicated documentation page include fixed storage quotas at onboarding with no mid-project extension (§7.2), 30-day project and home retention (§8.4–§8.5), prohibition on abusive SLURM SQL queries (§4.8), and PetaSC right to quarantine storage (§7.6).