The NOMAD (Novel Materials Discovery) Laboratory – a European Centre of Excellence

Deliverable 6.2

Final report on the HPC platform

Deliverable No: 6.2

Expected Delivery Date: 31/10/2018, M36
Actual Delivery Date: 20/11/2018, M37

Lead Beneficiary: CSC - IT Centre for Science Ltd.
Contributing Beneficiaries: Max Planck Society (MPG), Barcelona Supercomputing Centre (BSC), Leibniz-Rechenzentrum (LRZ), Humboldt-Universität zu Berlin (HUB), Aalto Univeristy (AALTO)

Authors: Atte Sillanpää (CSC), T. Zastrow, H. Lederer (MPG-MPCDF), Lauri Himanen (Aalto), Ruben García-Hernández (LRZ)

Executive Summary

Through work package (WP) 6, the NOMAD technology platform has been designed, implemented, deployed and put into operation. After defining and developing the necessary overall data workflow, a development platform was first put in place for all partners to start with their specific development work. After development of the first NOMAD services from other WPs, a production platform for serving the end user was installed and operated by WP6. This production platform has been further optimized in line with user needs, security and redundancy requirements. An essential feature of the NOMAD technology platform is the centralized user management with single sign on capability. A master instance of the NOMAD technology platform is installed and operated at MPG-MPCDF in Germany, while a clone of selected parts of the NOMAD technology platform is operated at CSC in Finland. The modular design of the NOMAD technology platform allows for easy installation and operation of additional instances at further locations.

TABLE OF CONTENTS

1 Introduction 4

2 Technology Platform and its Evolution 4

2.1 Evolution of the Platform 4

2.2 The Final Technology Platform 4

2.2.1 Virtual Machines and Containers 6

2.2.2 Storage 7

2.2.3 Authentication & authorization 8

2.2.4 Monitoring 10

2.2.5 Deployment of the NOMAD production platform 10

2.3 Technology Platform optimization 10

2.3.1 Lowering the Barrier for Accessing NOMAD Services 10

2.3.2 Security Aspects 11

2.3.3 Software Deployment Automation 11

2.3.4 Extended Compute services 12

2.3.5 Improving the Encyclopedia Ingesting Process 12

3 Lessons Learned and Conclusion 13

Revision History

Version 1.0, submitted 20/11/2018

Original version

1 Introduction

The objectives of WP6 were to:

- define and develop the necessary overall data workflow for the *NOMAD Laboratory CoE*, with input from the respective scientific partners,
- deploy and optimize its respective applications with a focus on HPC systems and map them to an adequate infrastructure, so that effort and resources in WP6 thus support all scientific WPs, and
- operate the underlying infrastructure for the Materials Science community.

This deliverable, D6.2 'Final report on HPC platform', describes the technology platform delivered at the end of the project. It explores how the platform changed and evolved during the project to meet user needs, and what lessons can be learnt for future projects in similar domains.

2 Technology Platform and its Evolution

2.1 Evolution of the Platform

The design and implementation of the development platform was initiated at the project start and carried out as described in detail in D6.1 'HPC platform requirements and architecture report'. Essential requirements and features were to support the:

a) needs of WP1 for the creation of the normalized data from the *Repository* data,
b) implementation of the *NOMAD Encyclopedia* by WP2,
c) integration of visualization services by WP3, and
d) Big-Data analytics, including machine-learning approaches, by WP4.

According to this design, the master setup of the complete development platform was implemented by and deployed at MPG-MPCDF. This setup is highly flexible and can be easily adapted to increasing needs both for increased resource and software requirements and - optional - extensions to other computing centers.

2.2 The Final Technology Platform

The final technology platform evolved as a production platform from the development platform, according to the requirements. The production platform was implemented in addition to the development platform according to Figure 1. The development platform has been maintained alongside the production platform for on-going development and testing purposes.

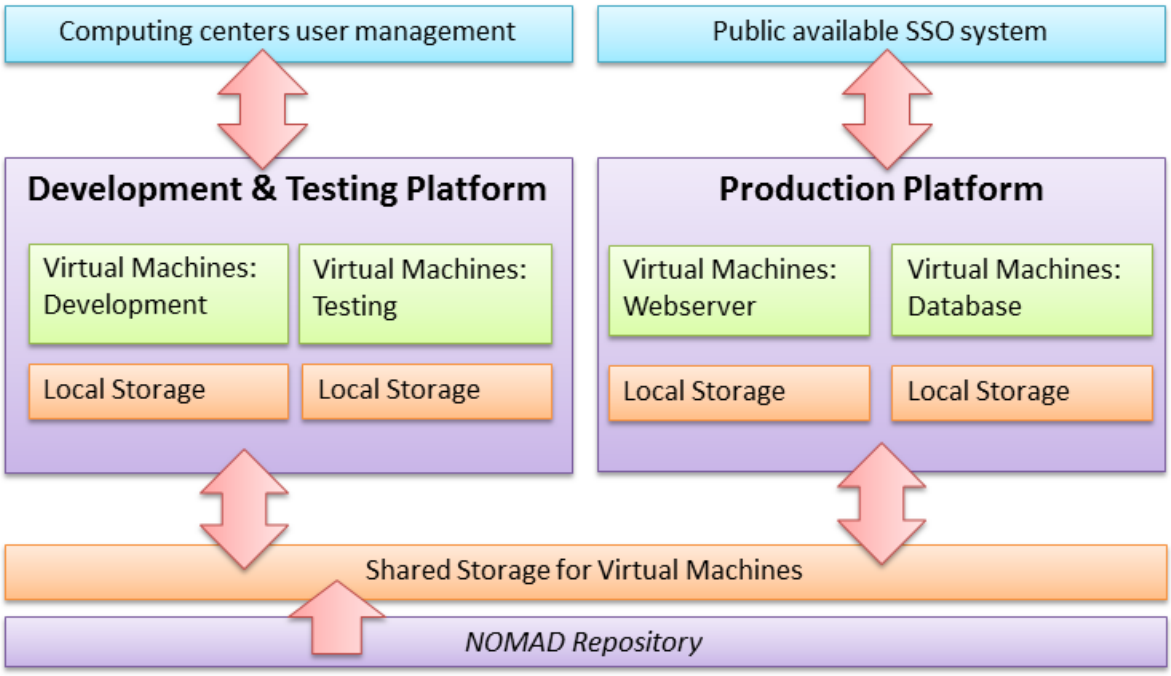

Figure 1: Evolution of the production platform from the development platform.

NOMAD services are directly accessed with a web browser without the need for any local client installations. In addition, it is also possible to access *e.g.* the *Encyclopedia API* programmatically via the RESTful interface and the *NOMAD Repository* and *Archive* contents directly. Some of the services are accessible without authentication, while some require it. The common user management for using *NOMAD* services is described in section 2.2.3.

Thus, from a user perspective, most of the services can be accessed either from the main website or sites linked to it, while the backend allows for using distributed resources for deployment. The full set of services has been put into production at MPG-MPCDF where the largest part of NOMAD hardware resources has been provided. Through the flexible design, the system is ready for setup also at other NOMAD sides, as a whole or only in parts. This gives additional flexibility to setting up the services locally, reduces the capacity requirements, adds redundancy and resilience and finally lowers the barrier to deploy the services. This also makes it easier for scientists outside of NOMAD to take some tools for local developments. The current situation of deployment of different services is shown in Figure 2.

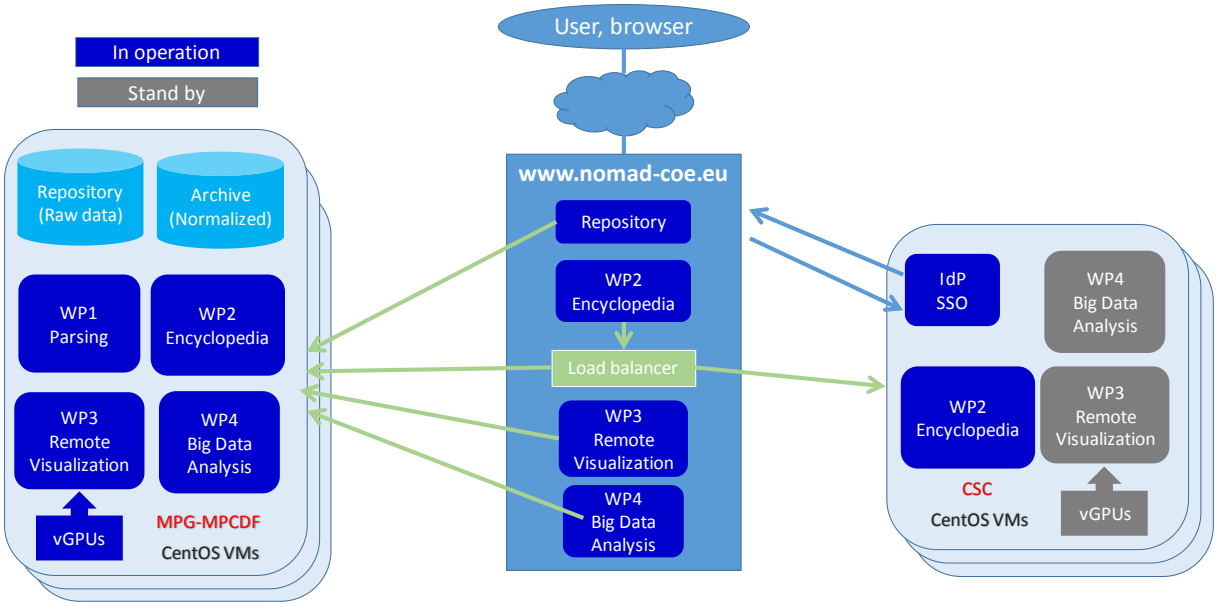

Figure 2: Current Infrastructure setup. The main web page is an entry point to all services. The *Encyclopedia* API backend runs both at MPG-MPCDF and at CSC. The load balancer redirects queries to them. The blue services are online, the grey services are on "stand by", i.e. the functionality exists, but resources have not been allocated and getting them online requires some manual work.

The production platform runs the following services:

- The *NOMAD Repository* - the database including the uploaded materials science calculation input and outputs, the GUI and API to query and upload the data and the links forward to the WP1 workflow.
- The parsing infrastructure (WP1) - the scientific code-specific libraries that convert the uploads into code-independent data in the *NOMAD Archive.*
- The *Archive* - database providing access to the code-independent materials data.
- The *Encyclopedia* GUI and API (WP2), including the load balancer.
- The Remote Visualization service (WP3) providing GPGPU powered visualization service of local high volume data directly via a browser and functionality to convert data for VR viewing.
- The *Analytics Toolkit* (WP4) with Big-Data Analytics Notebooks for querying and analyzing the materials data from the *Archive*, and
- Identity Provider - user registration and authentication service.

Due to the number of services and distributed character of the *NOMAD Laboratory CoE* infrastructure, a "Single Sign On" (SSO) System for external users became necessary (see 2.2.3). An SSO handles user accounts in a central service ("Identity Provider" (IDP)) and validates user logins for any NOMAD service ("Service Provider" (SP)).

2.2.1 Virtual Machines and Containers

The requirement for easy scalability has been met by extensive usage of virtual machines (VMs) and containers orchestrated with Kubernetes, OpenStack, Ansible and other scripts. Figure 3 presents the deployment of different NOMAD services. The *Repository* runs directly on VMs, whereas the parsing infrastructure (WP1), the remote visualization (WP3) and the *Analytics Toolkit* (WP4) all run on Docker containers, managed by a shared Kubernetes cluster. Whenever a user logs in, a set of new

containers are deployed by Kubernetes. The *Encyclopedia* uses a combination of both technologies: it is run on dedicated VMs, where its different services are encapsulated in Docker containers. These services are the GUI-webserver, the API, and the database for each production system, which is extended by the data processing for the development systems.

Containers: The use of Kubernetes as an orchestration engine for Docker containers allows for an easy future development and deployment path. It has become widely used and is likely to be supported in the future. During this reporting period also the *Encyclopedia*, VR-conversion scripts and *Remote Visualization Services* have been migrated to run on Docker containers, further simplifying the Infrastructure. This flexibility allows for optimal use of existing heterogeneous resources at participating HPC sites without the need for all sites set up an identical infrastructure. From an HPC-center point of view, the cloud (VMs) and container solutions are also preferred, as they enable dynamic scaling of the committed resources, thus minimizing idle time and making efficient use of resources.

VMs: The first NOMAD VMs running at MPG-MPCDF were based on SLES. The first services running at CSC cloud infrastructure were built on CentOS VMs. To harmonize the build and management scripts, all VMs were migrated to support CentOS. It is a well-supported flavor without license constraints and therefore can be used at any site without additional costs.

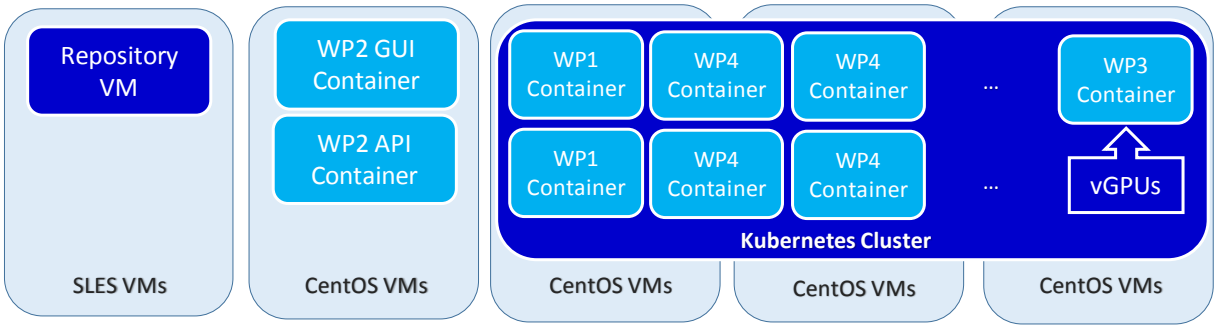

Figure 3: *NOMAD Laboratory CoE* Infrastructure Platform. The platform is based on VMs. The *Repository* runs directly on VMs, whereas the *Encyclopedia* services (WP2) run on containers on separate VMs, and the parsing infrastructure (WP1), the *Remote Visualization* (WP3) and the *Analytics Toolkit* (WP4) all run on Docker containers, managed by a shared Kubernetes cluster.

2.2.2 Storage

A differentiation between two kinds of storage has been made:

- Storage for developers on the development platform: storage resources for storing and creating the data on which the *NOMAD Laboratory CoE* is based on; this includes shared storage systems as well as local storages, mounted directly to servers or VMs, and
- Storage for the production platform: data made available through the production platform is separate from the storage of the development platform (MPG-MPCDF) or a separate file system (CSC) and has read-only access via the public NOMAD services; through the NOMAD tools (for analysis, visualizations etc.), the data produced by external users is available for further NOMAD services on a separate file system.

The 200 TiB GPFS storage at MPG-MPCDF has been suitable as the shared storage of the technology platform. Local storage on the dedicated servers has been increased upon requests to around 10 TB per server. Special storage solutions have also been used; for example, SSD disk has been

implemented for the *Encyclopedia* database on the production platform to speed up the query times. Additional storage capacity was added to the platforms to address the significant growth of the *NOMAD Repository* and the other databases created from the *Repository*.

For example, the largest data collection is the *Repository*. The number of contributions uploaded to it has grown from 634,001 (01 Oct 2015) to 1,235,389 (30 Oct 2015 or M1) to 3,407,035 (M18) to 6.240.929 (M36). The disk footprint of the uploaded data has grown from 0.8 TB to 9.9 TB, respectively. As described in the *Data Management Plan*, WPs 1 - 4 use a slightly augmented and differently compressed version of this data. The *Encyclopedia* has grown accordingly.

2.2.3 Authentication & authorization

Authentication for developers on the development platform has been handled by the already existing local user management systems of the computing centers. The production platform required a separate user management solution and several already existing Single Sign On (SSO) solutions were considered and evaluated. The only solution that could cater to all developer needs and NOMAD user requirements was based on MPASSid[1], which in turn is based on Shibboleth and SAML. The MPASSid IdP proxy was used to build the NOMAD user management system.

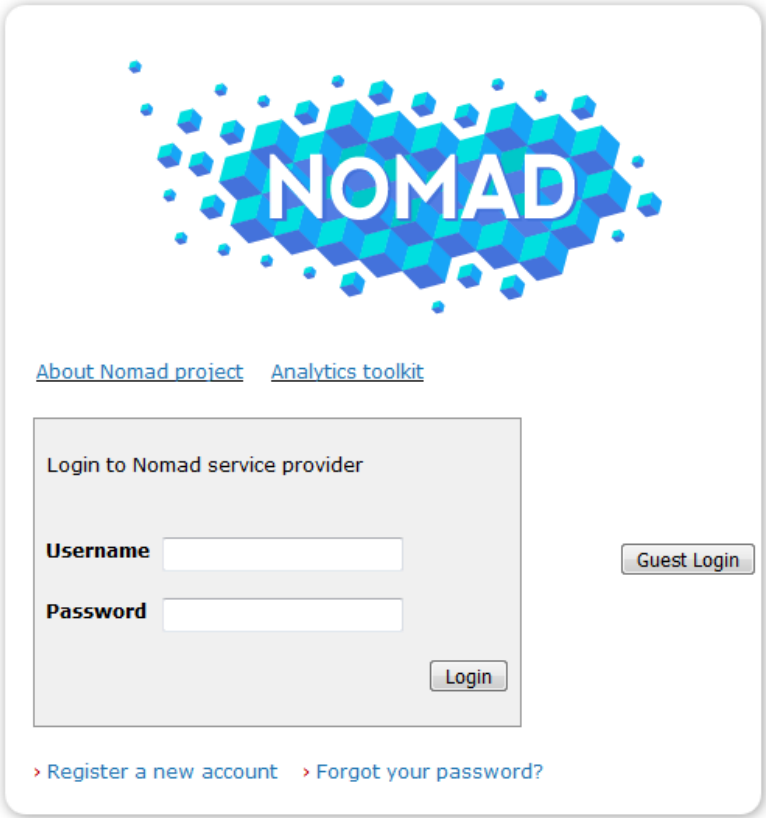

Figure 4: NOMAD Identity Provider landing page for logging in or managing the account.

The implemented SSO system consists of two parts: (i) an Identify Provider (IdP), which is responsible for the whole user account management: registration and login procedure, and (ii) an arbitrary number of Service Providers (SP). The user interface is shown in Figure 4. These SPs are the NOMAD

[1] https://mpass.fi/english/

applications, which use the IdP for authentication and authorization of the users as shown in Figure 5.

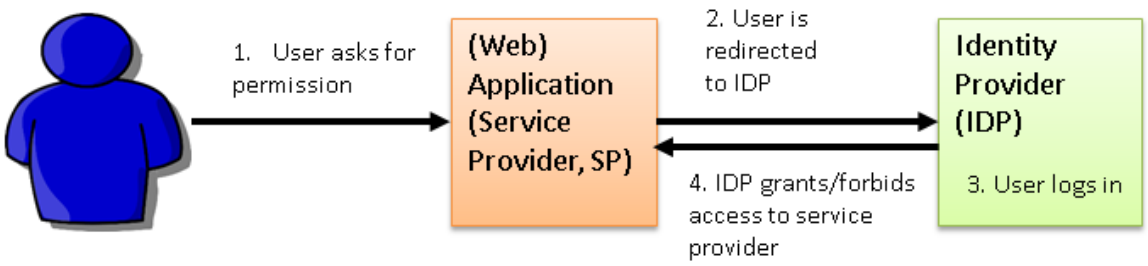

Figure 5 Workflow in a SSO system

The NOMAD IdP was developed, operated and modified according to the needs of WPs 2 - 4, although the integration of the *Repository* user management with the IdP is still in progress. The IdP comes with a linked local LDAP for storing the user data. A self-managed system gives flexibility to create new user attributes as needed. As MPASSid is based on Shibboleth it is compatible with most academic authentication federations (e.g. HAKA[2] and Switch[3]) and can later be linked to European or national AAI infrastructures. It also allows linking to social media accounts making it easier for users to authenticate.

Upon request from WP4 developers, separate container and VM deployments of the IdP complete with its own LDAP to manage user data were created. The motivation was firstly to help development work by allowing access to a non-production IdP server. Such an IdP could be set up inside the local development infrastructure, thereby removing the need to configure tunneling access outside the development system firewalls. Another benefit was that this further facilitates setting up the NOMAD system at a new site (e.g. in a private company) that does not want to access a common IdP for intellectual property (IP) or security reasons. The IdP container is managed with Ansible scripts which helps keeping the maintenance and management work low.

User account creation and management are self-service via email confirmation, but at the request of the *Industry Advisory Committee (IAC)* and other industry colleagues, a guest mode has also been configured. The guest mode allows access without disclosing any personal information to the system. After successful authentication, the IdP will assert the user identity to the requesting service using a SAML token. Individual NOMAD requests, that mostly take place via the REST API, include the token in their messages, thereby achieving user identification and authentication. All NOMAD services can utilize the same token, so no additional action from the user is needed to access the full range of NOMAD services. The IdP runs on a virtual machine in the CSC cloud platform and when needed, a redundant IdP can easily be added at another site or server.

[2] https://wiki.eduuni.fi/display/CSCHAKA/Federation
[3] https://www.switch.ch/aai/about/federation/

2.2.4 Monitoring

The involved computing centers monitor the availability of the NOMAD infrastructure. In case of future regulation for the access to NOMAD resources, monitoring at service level will be used as the basis for accounting.

Regarding the actual services, the most important resource to monitor has been the storage. The computing capacity on the deployed servers at MPG-MPCDF has so far been adequate for the usage.

Accounting and limiting of resources for external usage can be done in two ways:

- users are granted access to a platform where the NOMAD services are installed and are given a dedicated amount of resources (CPUs, RAM, storage, etc), or
- the NOMAD applications themselves track resource use of the user and potentially limit further usage above a set quota.

2.2.5 Deployment of the NOMAD production platform

The *NOMAD Repository* is the main source for data in the *NOMAD Laboratory CoE*. Currently, it is the only place where users can upload materials science data in the NOMAD ecosystem. The *NOMAD Repository* is operated at MPG-MPCDF. The data of the *NOMAD Repository* is normalized and then made available for the *NOMAD CoE* applications – the *NOMAD Laboratory*. The master installation in the *NOMAD CoE* was realized at MPG-MPCDF while a second instance has been deployed at CSC.

Both normalized data and metadata can be synchronized by the respective service operators between the two sites via conventional commandline tools as shown in Figure 6.

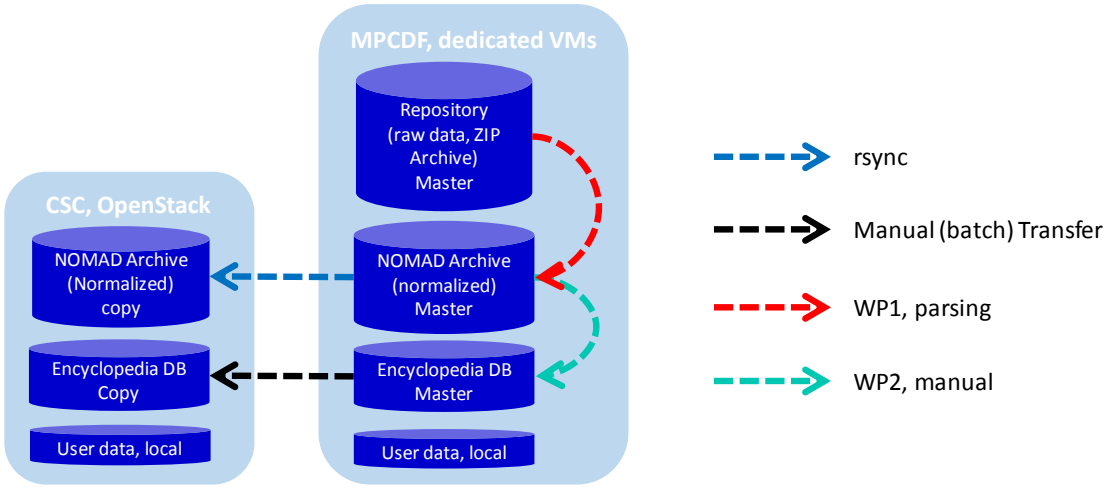

Figure 6: Data flow between the Master site (MPG-MPCDF) and a remote site (CSC) in the distributed NOMAD infrastructure.

2.3 Technology Platform optimization

2.3.1 Lowering the Barrier for Accessing NOMAD Services

Stepwise access to NOMAD services for ease of use:

1) The first step, trying out the web service, can be done with an anonymous account. Only a web browser is required.
2) More services and resources can be accessed and stored easily and without cost by creating a NOMAD user account. Only a working email address is required. Users can also access these additional services through a web browser alone, as all the processes run on NOMAD servers.
3) The modular and portable design of the technology platform enables users to set up selected NOMAD services locally at their own site. This improves IP security and enables the use of existing computational resources on site. However, few institutions would invest in the installation process without first being convinced that the service is useful (i.e. through 1-2 above).
4) The NOMAD IdP allows creating additional attributes, *e.g.* for VIP status for researchers who need extensive computational resources, but do not want a local installation. This status could allow additional resources to be deployed, perhaps for a fee.

2.3.2 Security Aspects

The *NOMAD Laboratory CoE* services are web-based and accessible from anywhere by any potential user. This setup requires a high attention on security-related issues.

The computing centers host the *NOMAD Laboratory CoE* services. At the level of hardware and operating systems, the computing centers will take care of upcoming security issues. This includes regular checks for necessary updates of the operating system and the installed basic software. Backup capabilities have been provided for all NOMAD related data, configurations and application code. For example, MPG-MPCDF offers backup functionality on the GPFS cluster it operates for *NOMAD Repository* and other *NOMAD services*: a fileset ("/nomad/backup") which is automatically backed up on tapes via Tivoli Storage Manager.

At the application level, the responsibility is on the developers of the NOMAD services. From the infrastructure point of view, using continuous deployment and largely automated deployment scripts facilitate frequent security updates and rapid fixing of reported vulnerabilities, as well as periodic clean installs from the source.

2.3.3 Software Deployment Automation

The provisioning of VMs, configurations and application setups in the NOMAD Infrastructure Platform has been automated, where possible, with tools such as Ansible, which is a free-software platform for configuring and managing servers.[4] Ansible scripts have been used for different use cases:

- Basic level (through computing centers): setup of operating system and basic tools; will be used to duplicate (virtual) machines across computing centers, set up and configure Kubernetes clusters or the IdP functionality.
- Application level (through NOMAD service developers): installation and configuration of user space applications like Kubernetes, databases, application servers etc.

[4] http://docs.ansible.com/

During the project, MPG-MPCDF has offered its GitLab platform for use by all NOMAD developers. GitLab is in first place a versioning system for distributed code development on the basis of Git, a version control system that can be used for software development and other version control tasks. Besides this basic functionality, the GitLab software offers individual wikis, issue trackers and a broad and deep integration with further development tools (*e.g.* Integrated Development Environments), which were all utilized in the CoE. Continuous integration was also configured to be possible directly from within GitLab (see Fig. 7). Some development teams linked this up with further communication tools. For example, a push to the master branch of the *Encyclopedia* GUI repository causes an automatic deploy to the development webserver, and at the same time a notification is sent to a channel in Slack, so that other developers are immediately informed that there has been a change they might want to check out.

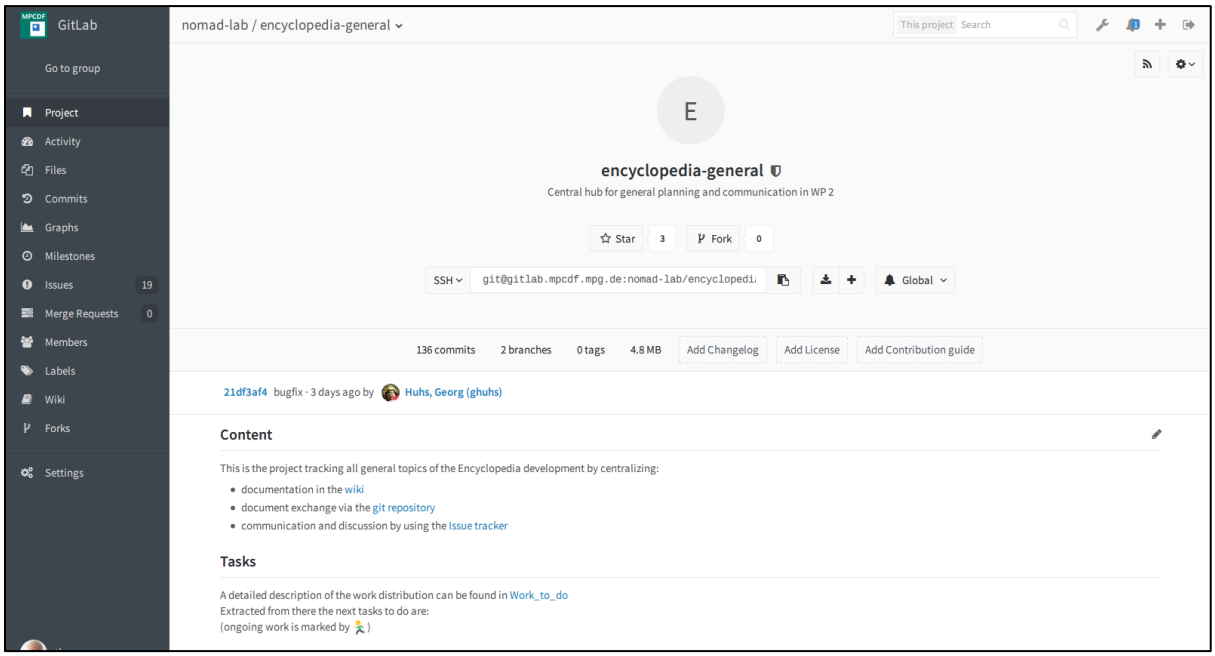

Figure 7: GitLab service at MPG-MPCDF

2.3.4 Extended Compute services

In addition to the handling of computing needs through the NOMAD computing centers, HPC resources were offered by PRACE to be able to fill in gaps in the simulation data in the *Repository*, if required. In particular, in the early phases of the parser development, there was frequent need for heavy re-parsing of NOMAD data. MPG-MPCDF provided direct HPC access that could be used instead of the smaller scale dedicated resources, which are more suitable for the reactive parsing of newly uploaded data into the *Repository*. If needed in future, the resources at NOMAD computing centers can be made available for parsing the data in a geographically distributed way and synchronize the results to the *NOMAD Archive*.

2.3.5 Improving the Encyclopedia Ingesting Process

In the first half of the project, the *Encyclopedia* ingestion process was identified as a bottleneck. The first solution to improve the performance was to restructure the database schema to allow for parallelizing the process. This improved the performance, but with ever growing data volumes, eventually a different approach was needed.

The highly hierarchical nature of the NOMAD Meta Info, frequent updates on the database schema and high insertion volumes made the original relational PostgreSQL-based solution inconvenient for the *Encyclopedia* development team. To remedy these problems, WP6 has assisted the *Encyclopedia* team in transitioning from an SQL-based database solution to a schemaless NoSQL environment based on MongoDB.

Population of the *Encyclopedia* database is done by systematically crawling through the user-uploaded calculations stored in the *NOMAD Archive*. As described above, this task was efficiently parallelized, but the following insertion step speed of the SQL database did not scale sufficiently as more processes were added to this task. This bottleneck could be avoided by storing the information as single nested data entries in the more suitable NoSQL alternative. Nested JSON-documents are thus faster to store but also to serve. MongoDB also simplifies the serving of *Encyclopedia* contents as the documents are stored in the JSON format that can be directly served through a web-API. Additionally, the schemaless approach provided by MongoDB allows more rapid prototyping as there is no need to update the database schema for every change in the data layout.

The migration to the NoSQL environment has been gradual by first making a parallel implementation that works alongside the old database. The production environment is scheduled to be updated to use the new database as soon as other parts of the service, including the API, have been updated to fully support the changes.

3 Lessons Learned and Conclusion

At the start of the project, the first aim was to try to design a full set of web-based services in a distributed technology platform. Over the course of the implementation of the different parts of the technology platform, it turned out that only parts of the NOMAD infrastructure needed to be distributed over NOMAD computing centers. A master instance of the NOMAD technology platform was implemented as close as possible to the largest database, the NOMAD *Repository*, with a tight integration of essential resources. Further instances of the NOMAD infrastructure can be easily cloned to other computing centers for ease of use for their local users. One clone of essential parts of the NOMAD technology platform has been successfully installed at CSC.

WP6 has proven to be of essential importance for the designing, implementation and deployment of the NOMAD technology platform through a collaborative effort of the technical experts involved.