Meta hosting in DARIAH-DE
(M 2.2.1)

Version 21.11.2016
Cluster 2.2
Responsible Partner GWDG

DARIAH-DE:

Transferring Research Infrastructures for the e-Humanities in the Operational Phase

SPONSORED BY THE

Federal Ministry
of Education
and Research

Project: DARIAH-DE: Transferring Research Infrastructures for the e-Humanities in the Operational Phase
BMBF Fund number: 01UG1610A to J
Duration: March 2016 till February 2019

Document status: Final
Dissemination level: Public
Authors:
     Xi Kong, GWDG

Revisions:

Date	Author	Comment

Table of Contents:

1. Introduction

This document describes the status of the DARIAH-DE meta hosting[1] service, which is deployable, portable, reviewable and sharable[2]. A user manual of the DARIAH-DE meta hosting service is available that provides a step by step guide for users to exploit this service.

The document is transferred from the internal DARIAH wiki pages to have an official public version of the DARIAH meta hosting service. And the internal DARIAH wiki pages will be maintained continuously for the extension of the service for its ease of use and maintenance.

2. Meta hosting as a DARIAH-DE service

2.1. Existing features

- The meta hosting service is available in the GWDG portal.
- There is a dedicated VM for each container containing user needed specific API.
- Example meta hosting services are available as single images in the GWDG compute cloud.
  - eXist[3], Voyant[4] and Neo4j[5] are available for the time being.
- Currently, manual activation and only for registered GWDG users.

2.2. Future extension

- VM image with web interface and the option of multiple containers for users within the VM.
- With the further development of the GWDG-AAI, the release is also possible for DARIAH users.

---

[1] See also: Sharing software in a distributed infrastructure. http://ieeexplore.ieee.org/document/7160268/
[2] https://wiki.de.dariah.eu/display/DARIAH2/Metahosting-Workshop?preview=/39324449/40436379/rybicki_metahosting.pdf
[3] http://www.exist-db.org/
[4] https://voyant-tools.org/
[5] https://neo4j.com/

3. User manual of the DARIAH-DE meta hosting

To use an existing DARIAH-DE meta hosting service, the user should follow three steps.

- Step 1: Users authenticate.
- Step 2: Create a server, which contains the container with needed API. Firewall rules should be defined in this step, and this is API specific configuration.
- Step 3: Connect with the APIs and setup the API specific settings, primarily user authentication.

In this manual, we demonstrate three existing DARIAH-DE meta hosting services, i.e., eXist, Voyant and Neo4j, as examples.

3.1. Step 1: Users authenticate

To use the service, users need to log into the GWDG Portal[6] to have their credentials authenticated.

Use your username or e-mail address and password to log into the GWDG Portal.

---

[6] https://portal.gwdg.de

3.2. Step 2: Create server

1. Go to the menu "**Server Services**".

2. Go to the menu "**Cloud Server Beta**".

3. Go to the menu "**Server**"

4. Click the "**+**" button to create the server.

5. Select the VM needed, e.g. size, and then click "**Next**".

6. Select the image to be created:

    a. DARIAH eXist-db v6

    b. DARIAH Neo4j v3.0 or

   c. DARIAH Voyant v2.1, and click "**Next**".
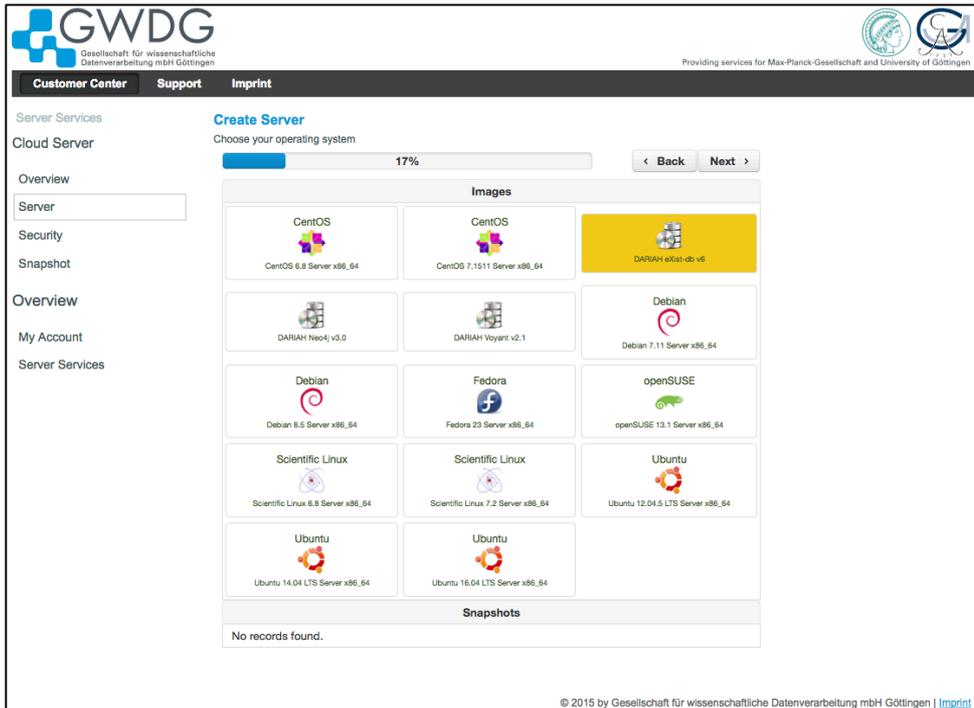
7. Security settings/Define the firewall rules

The firewall rules should be defined for each API in this step. Different APIs have different security requirements, please check the documentation of the API if you are not sure what rules should be applied. Here we demonstrate the basic setting for the APIs **eXist**, **Voyant** and **Neo4j**.

7.1. Select existing firewall rules or add new security group.
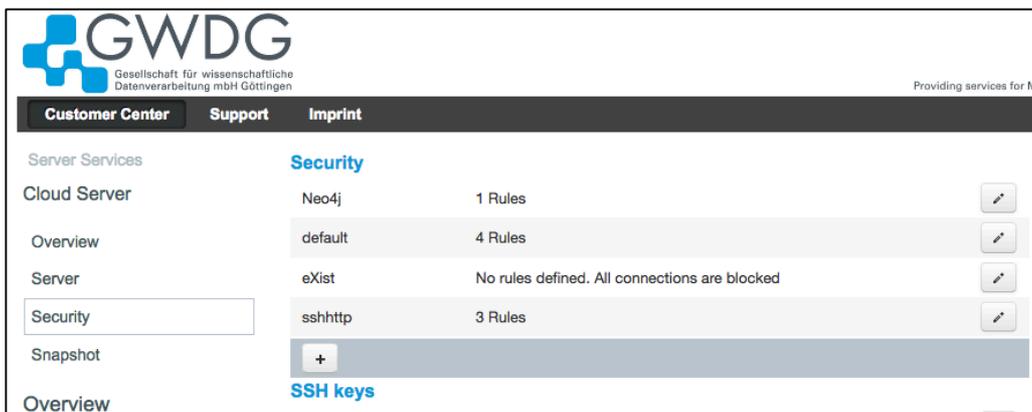7.2. In this demonstration, we "**Add Group**" for eXist.
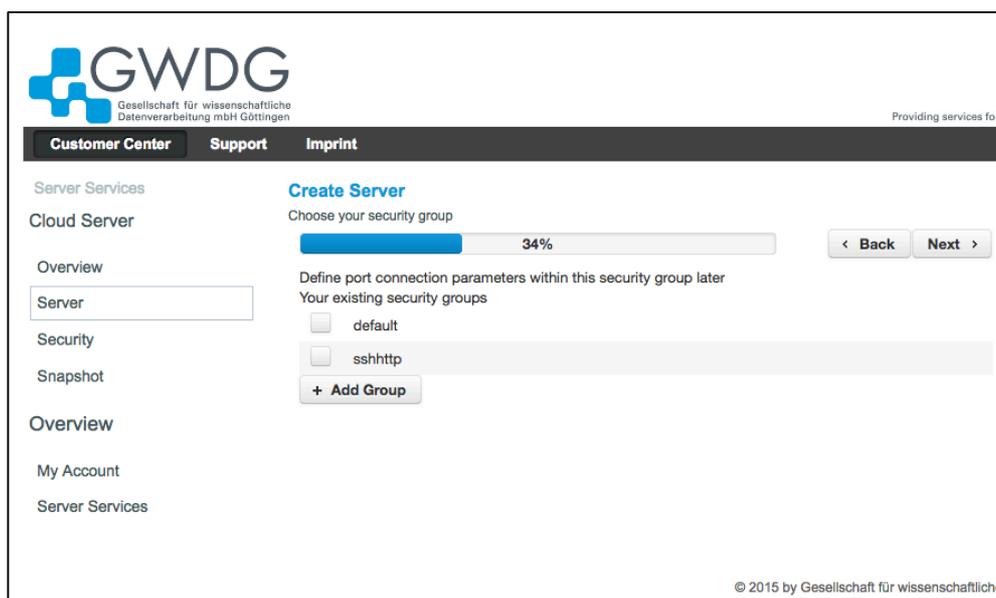
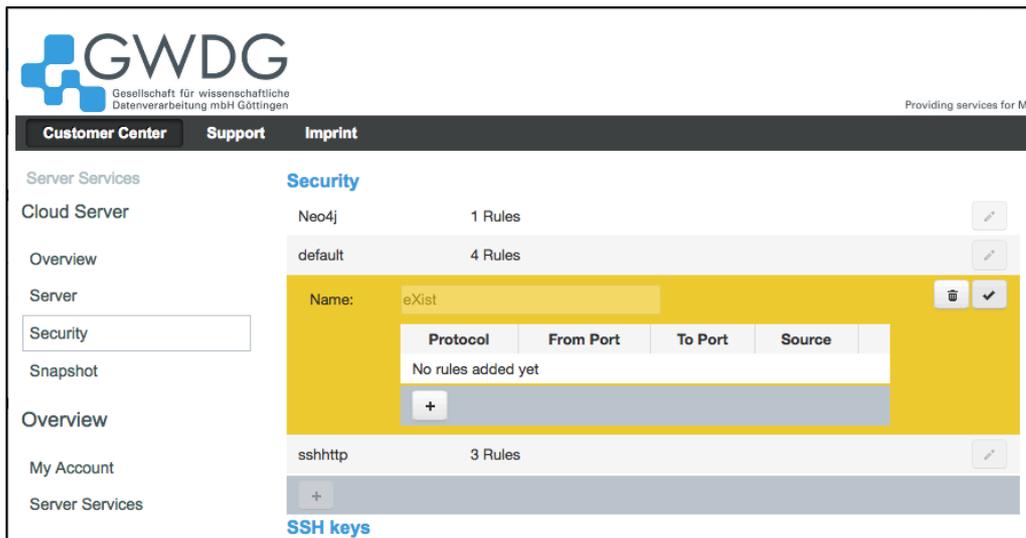7.3. Assign a new security group named "eXist" and "**save**".
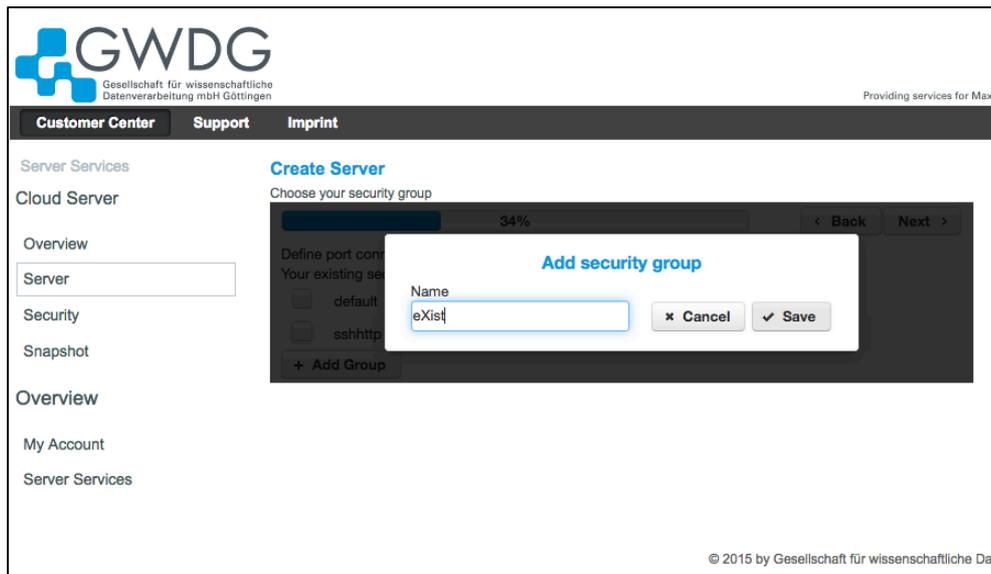
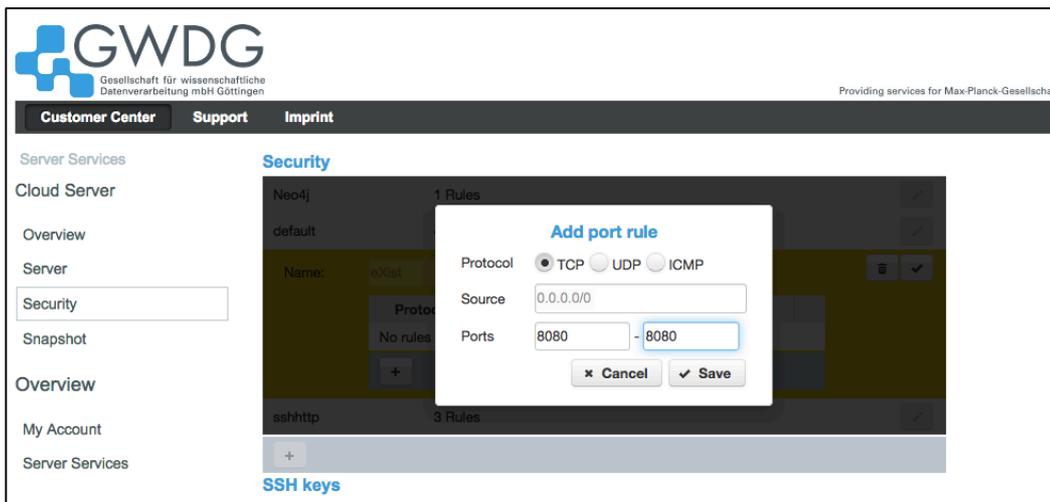7.4. Go to "**Security**" and edit the new created security group "eXist".

Until now no rules defined yet for eXist. All connections are blocked. Click ✎ to edit rules for eXist.
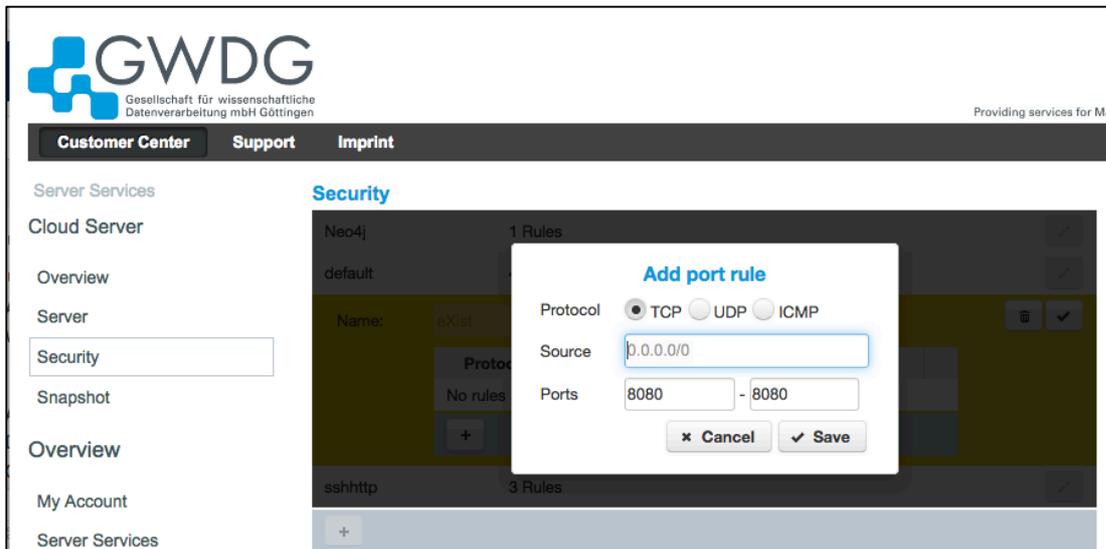
7.5. Click "**+**" to add a rule for eXist.

7.6. Port **8080** should be open for eXist. A single IP address or a range of subnet, where the API should be accessed can be assigned by "**source**". Click "**save**" to save the rules.

7.7. We've now defined a security rule for eXist.
In this example the eXist can be accessed from the GWDG network.
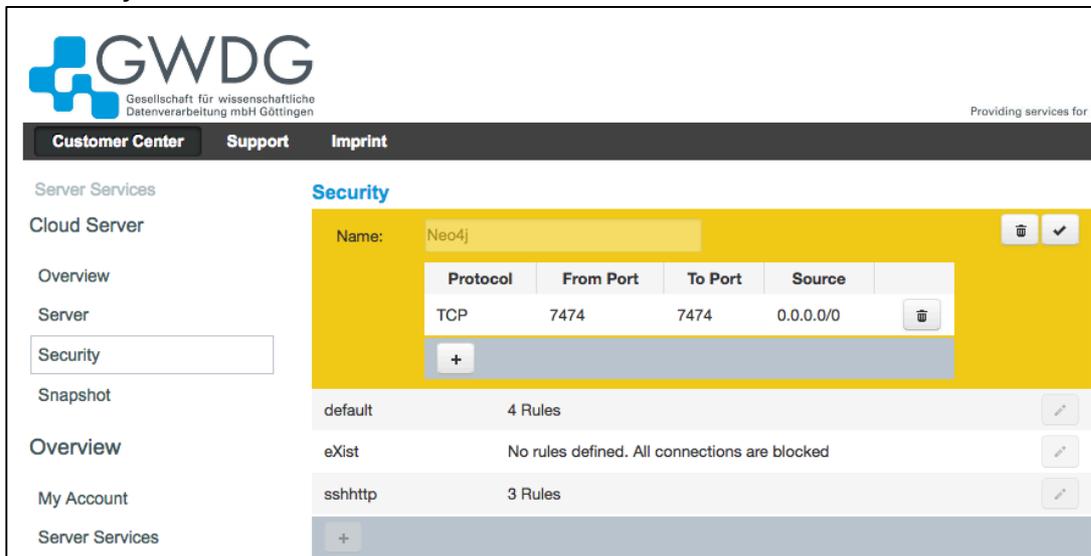
Firewall rules for Voyant
Please go through the steps of "**Security settings**", 7.1-7.7. Similar to eXist the port **8080** should be opened for Voyant.

Firewall rules for Neo4j
Go through the steps of "Security settings", 7.1-7.7. The port 7474 should be opened for Neo4j.

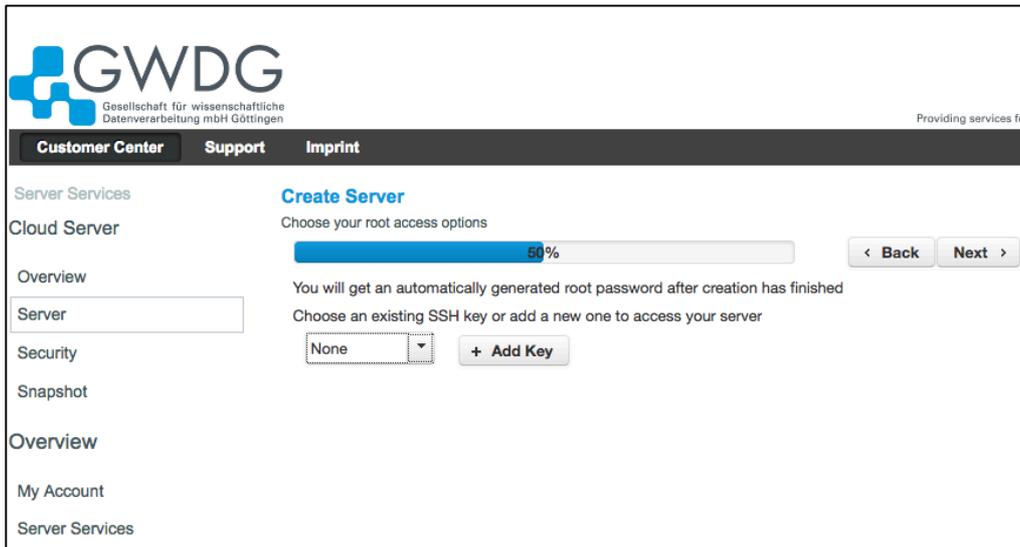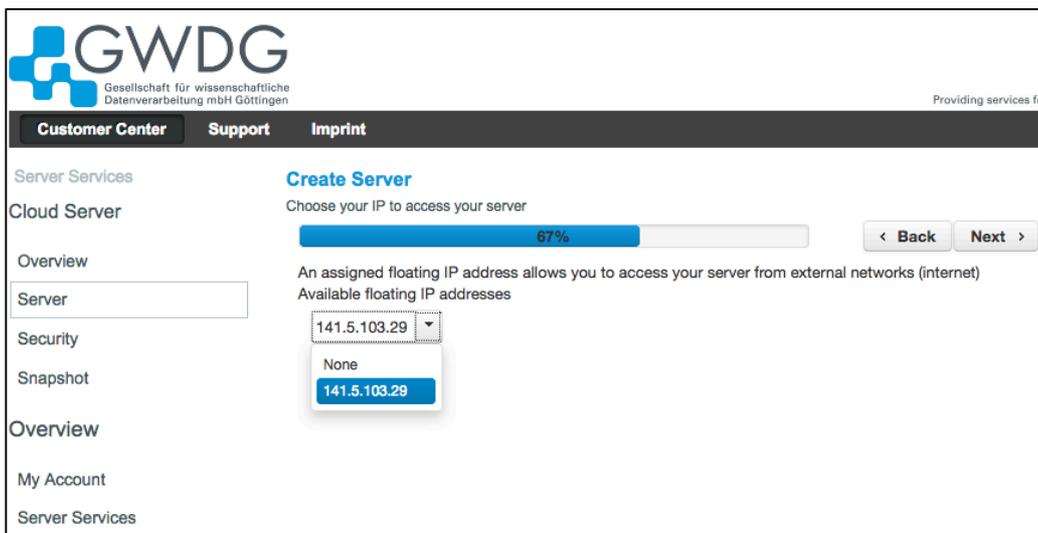**Further steps to create a server**

8. After setting up the security rules, click "**Next**". An automatically generated root password will be created.

9. Assign an IP address for the server, which allows you to access your server from external networks/internet. And click "**Next**".

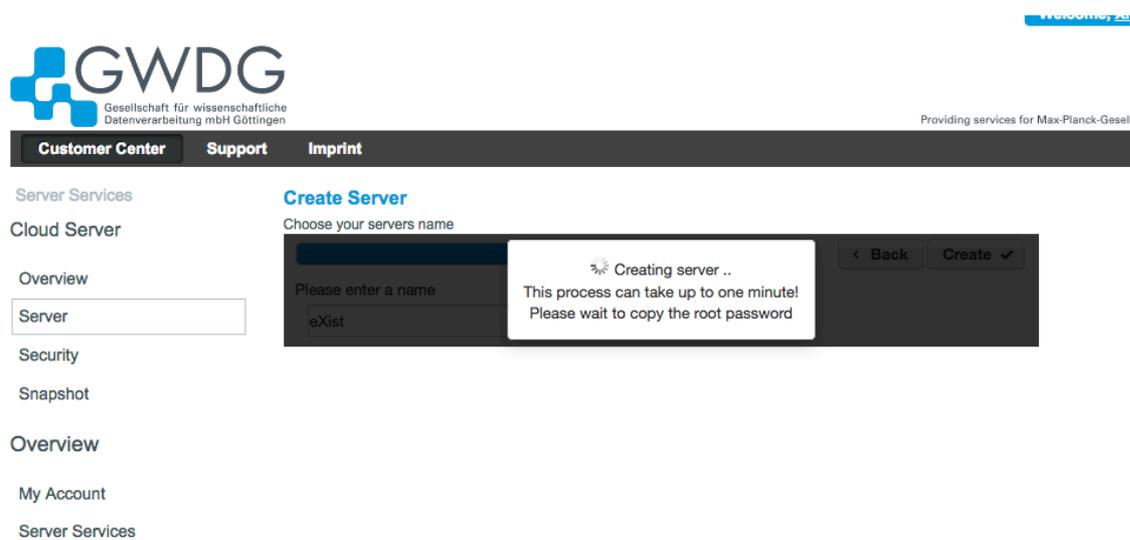10. Assign a name for the server and click "**Create**".

11. The server will be created. This process can take some minutes. Please wait to **write down** and **save** the root password.

12. The server is created. Click "**Finish**".

13. Please note the message showed in this example of step 12, "Server created unable to attach Floating IP". In this case, go to "**Server**" and **edit** the created server "eXist".

14. Manually attach the IP again. Select the IP address and click "✔".

15. The server is updated and the IP address is attached successfully.

3.3. Step 3: Connect with the APIs and setup the API specific settings

The server with the container containing eXist is now ready to use. But before you start using the API, some API specific setting should be configured. In this user manual, we show the setup related to meta hosting. For other features of the API, please look up the related API documents.

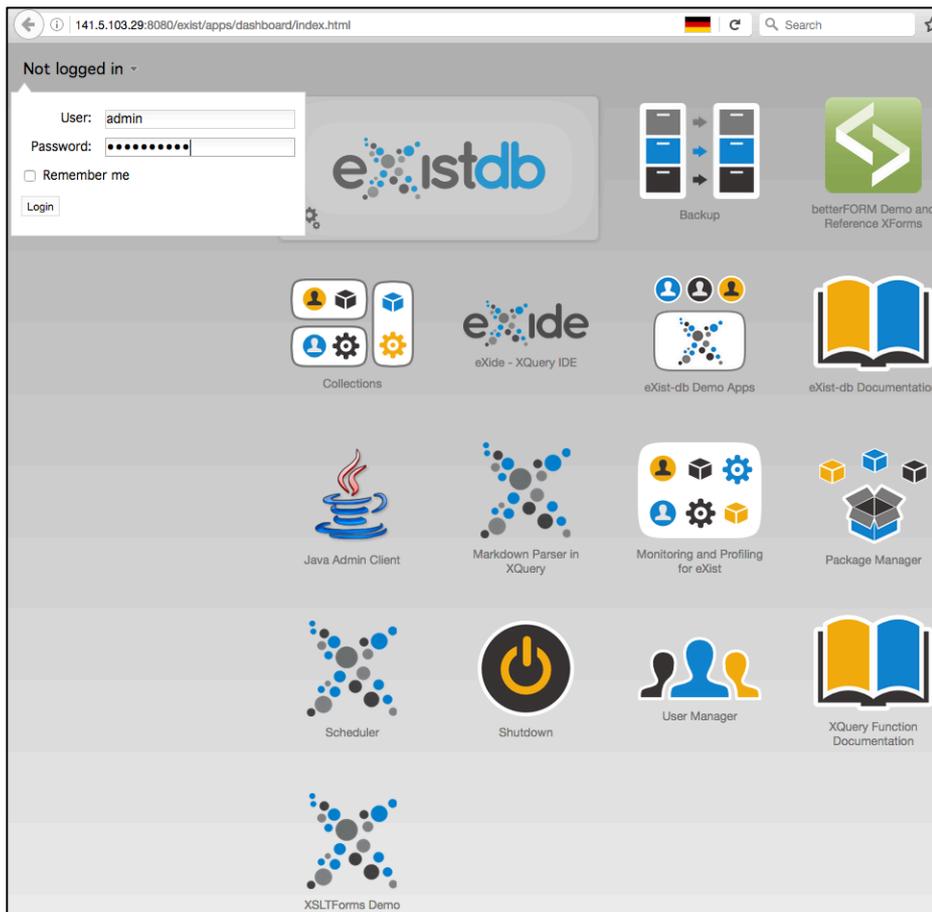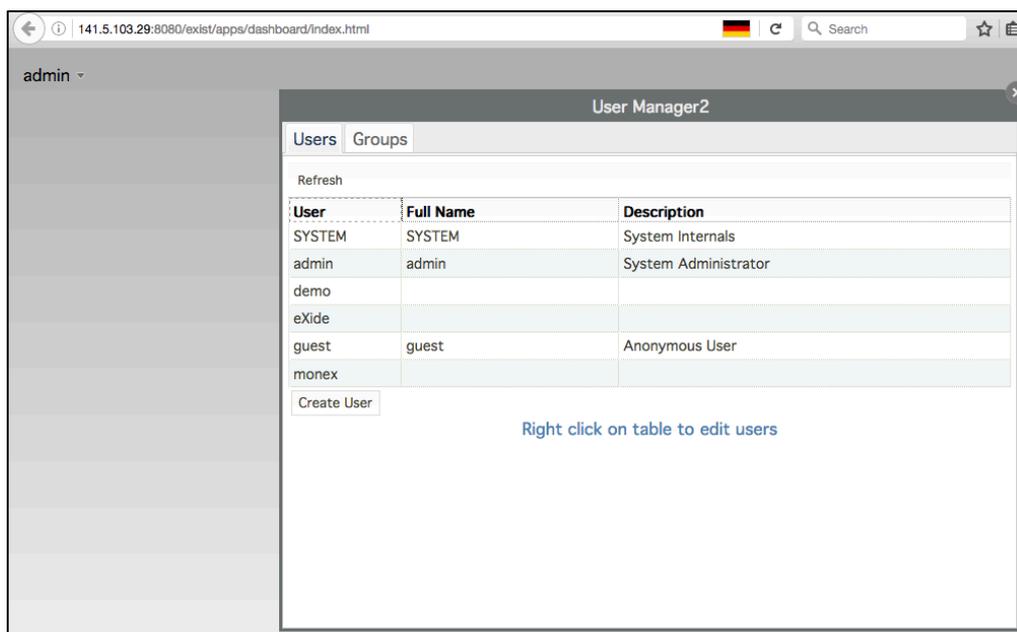### 3.3.1. API specific settings of eXist

1. Open a browser and go to [**IP Address : Port**], in this example: http://141.5.103.29:8080
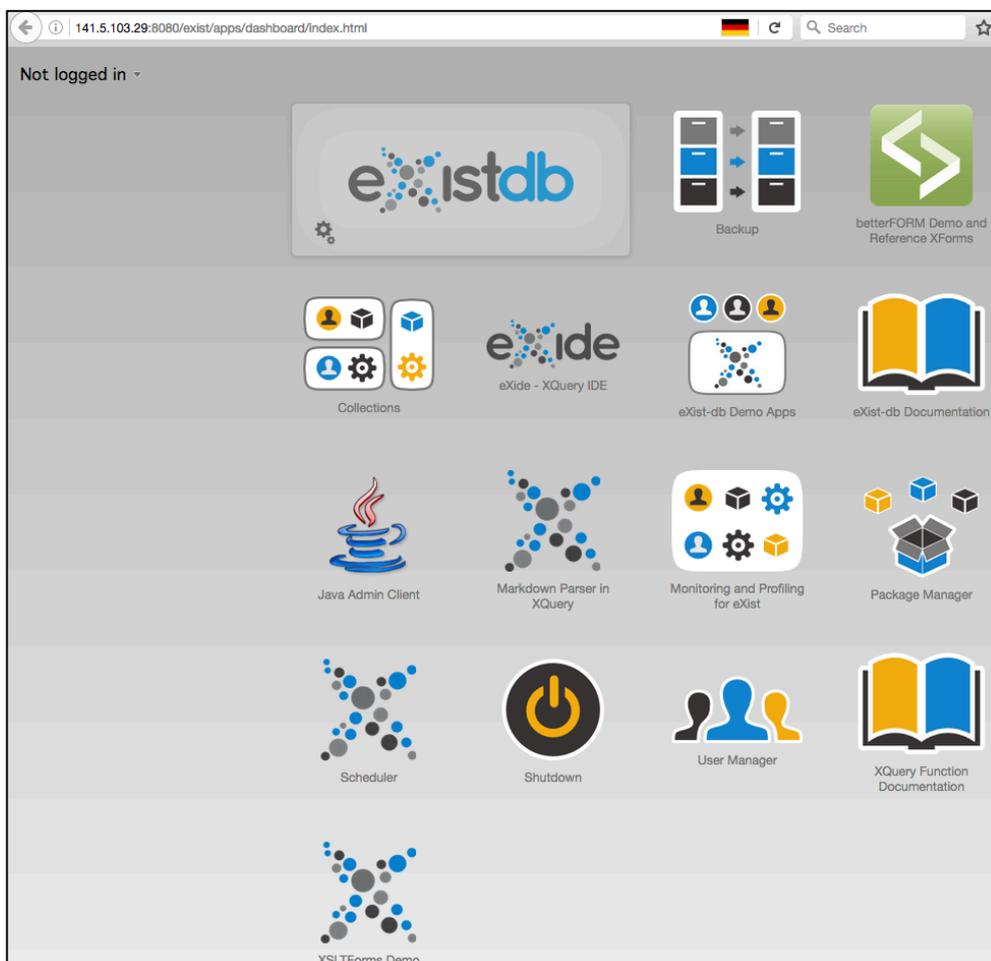
2. Go to "User Manager", and log in with the initial set up.
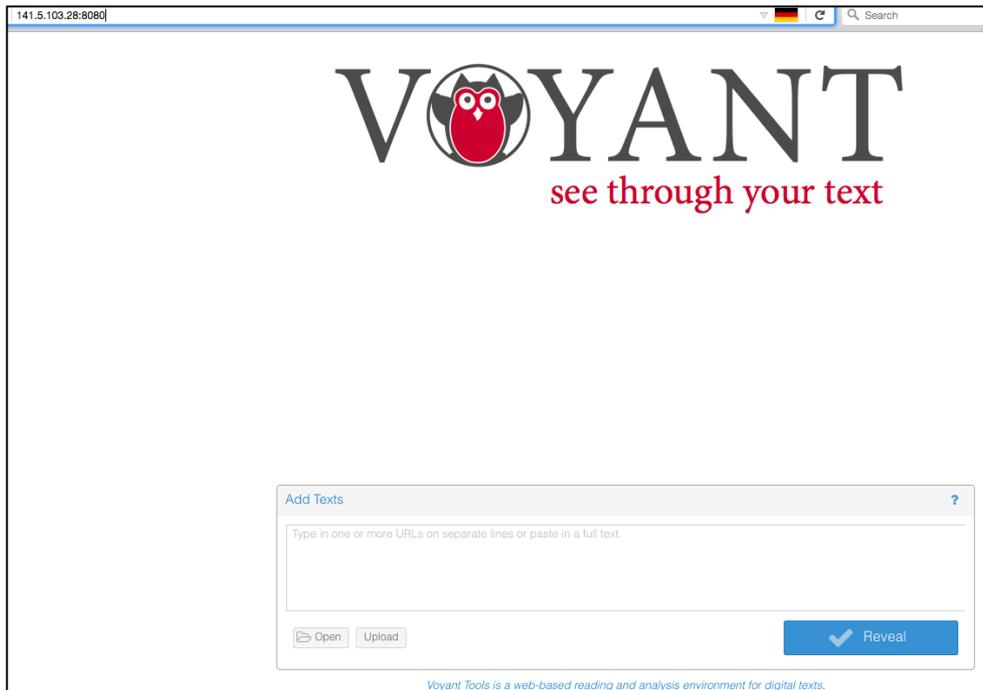   User: admin, Password: Dariah2016. Click "**Login**".

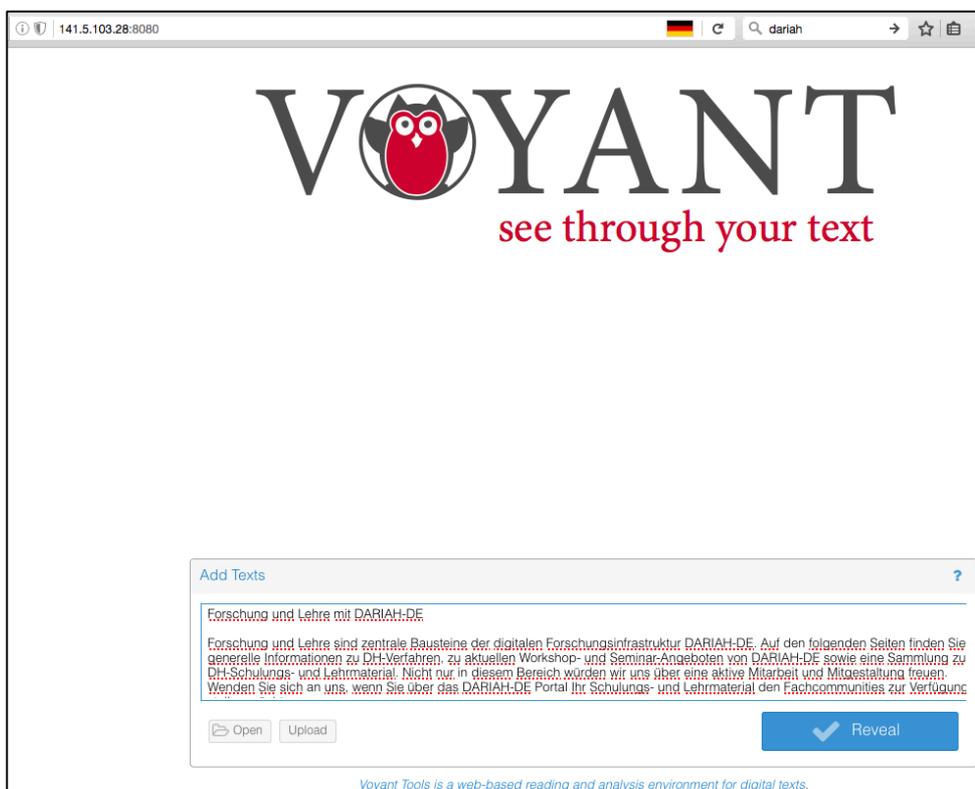3. Now, you can change the password and manage user permissions.

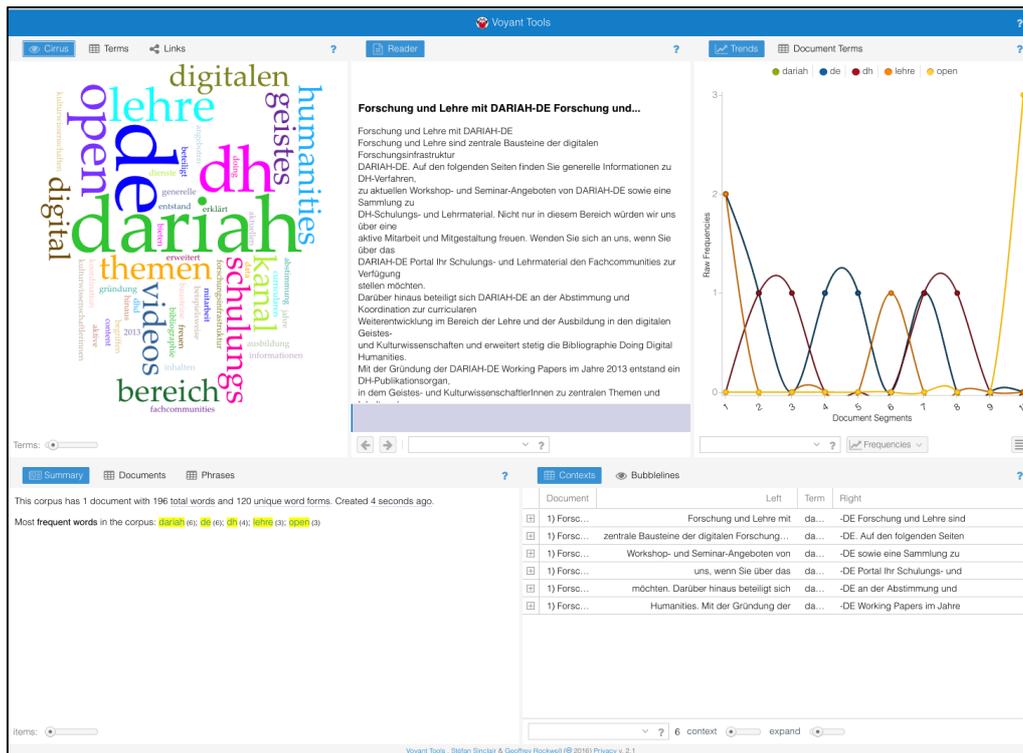4. **Enjoy:-)**

3.3.2. API specific settings of Voyant

1. Open a browser and go to [**IP Address : Port**], in this example:
   http://141.5.103.28:8080.
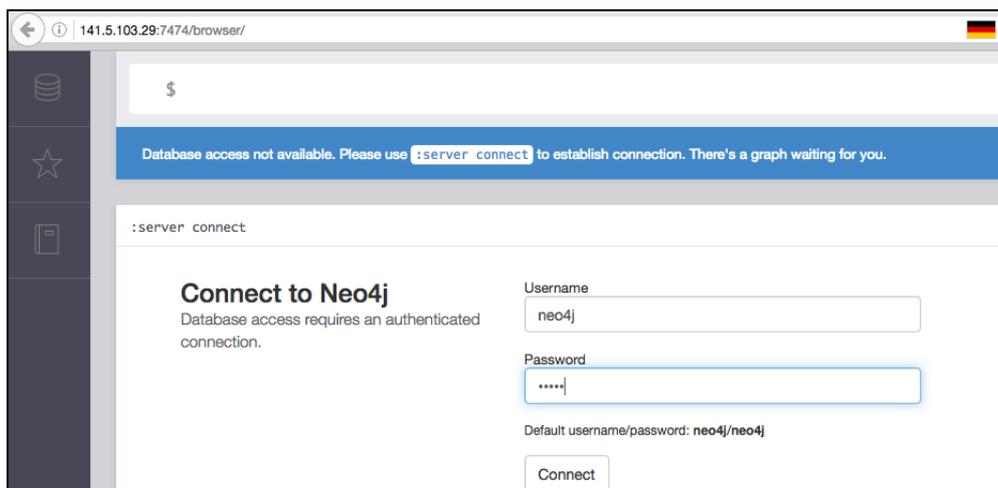

2. Add some texts to reveal.

3. Enjoy the result:-)

3.3.3. API specific settings of Neo4j

1. Open a browser and go to [**IP Address:7474**].

2. Use the default username and password, **neo4j/neo4j**, to connect with Neo4j.

3. You're now connected with Neo4j.