Deployment: Spring Boot application with Postgresql database

1 Configuring Java + Maven + Spring project for deployment

Add PostgreSQL driver as dependency in your project's **`pom.xml`**. Copy and paste the below snippet, place it as a child element of **`<dependencies>`** element.

```
<dependency>
           <groupId>org.postgresql</groupId>
           <artifactId>postgresql</artifactId>
           <scope>runtime</scope>
</dependency>
```

Create a file named **`Dockerfile`** (no file extension) in your coding project's root. Content of the **`Dockerfile`** for a **`Java + Maven`** project is (use Copy + Paste):

```
FROM eclipse-temurin:17-jdk-focal as builder
WORKDIR /opt/app
COPY .mvn/ .mvn
COPY mvnw pom.xml ./
RUN chmod +x ./mvnw
RUN ./mvnw dependency:go-offline
COPY ./src ./src
RUN ./mvnw clean install -DskipTests
RUN find ./target -type f -name '*.jar' -exec cp {} /opt/app/app.jar \; -quit
FROM eclipse-temurin:17-jre-alpine
COPY --from=builder /opt/app/*.jar /opt/app/
EXPOSE 8080
ENTRYPOINT ["java", "-jar", "/opt/app/app.jar" ]
```

The above content is a generalized **`Dockerfile`** configuration and can be used for other Spring Boot projects.

Create a new deployment profile for your application. You need to create a new file in the **`<your app>/src/main/resources/`** directory.

Name the new file **`application-rahti.properties`**.

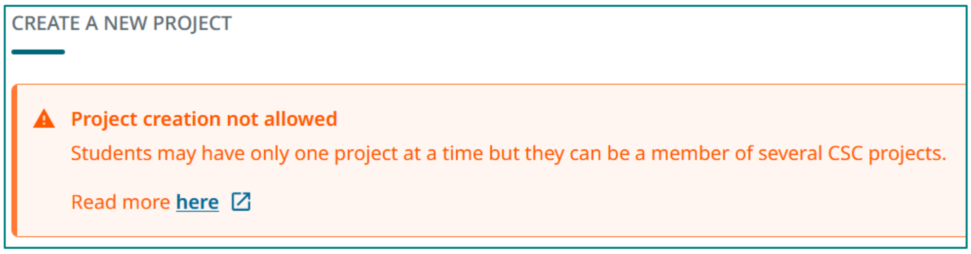
Content of the **`application-rahti.properties`** is (use Copy + Paste):

```
spring.datasource.url=jdbc:postgresql://${POSTGRESQL_SERVICE_HOST}:${POSTGRESQL_SE
RVICE_PORT}/${DB_NAME}
spring.datasource.username=${DB_USER}
spring.datasource.password=${DB_PASSWORD}
spring.jpa.show-sql=true
spring.jpa.generate-ddl=true
spring.jpa.hibernate.ddl-auto=update
```

Commit the above changes and push them to your GitHub repository.

2 Creating a project in CSC

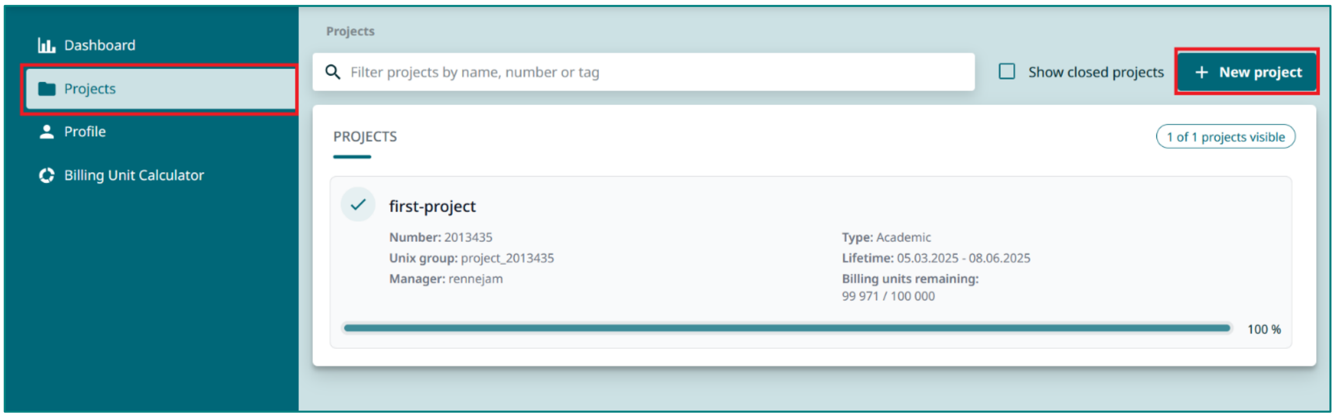
NOTE! If you already have created a project in CSC, you are unable to create another one. In that case proceed to chapter 3 of these instructions.


CREATE A NEW PROJECT

Project creation not allowed
Students may have only one project at a time but they can be a member of several CSC projects.

Read more here

Login to CSC at https://my.csc.fi/login using you Haka user account (Haaga-Helia credentials) or your CSC user account (username sent to your email). Navigate to Projects view and start creating a new project.

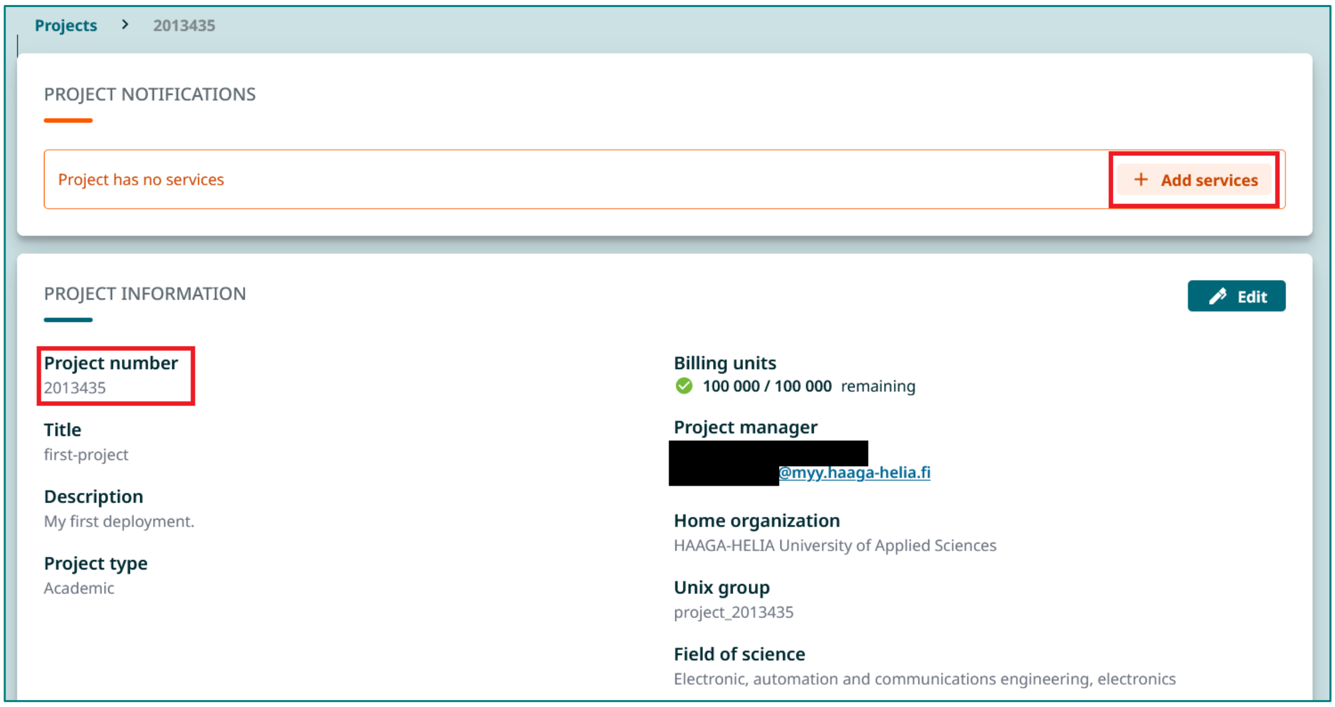


first-project
Number: 2013435
Unix group: project_2013435
Manager: rennejam
Type: Academic
Lifetime: 05.03.2025 - 08.06.2025
Billing units remaining:
99 971 / 100 000
100 %

Fill in your project’s information as instructed in the form. Read and accept all terms of use and privacy notice. Choose `Create project` to continue.

You have now created your CSC project. Within this project you can start adding CSC's services. For your deployment you only need `Rahti – Container Cloud`. Click `Add services` and select `Rahti`.

NOTE: check your project number. You will need it later!



Projects > 2013435

PROJECT NOTIFICATIONS

Project has no services

+ Add services

PROJECT INFORMATION

Edit

Project number
2013435

Title
first-project

Description
My first deployment.

Project type
Academic

Billing units
✓ 100 000 / 100 000 remaining

Project manager
[redacted]@myy.haaga-helia.fi

Home organization
HAAGA-HELIA University of Applied Sciences

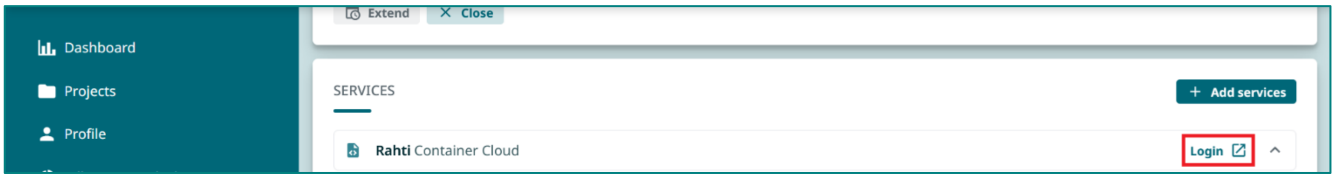
Unix group
project_2013435

Field of science
Electronic, automation and communications engineering, electronics

Once you have added the `Rahti` service you might have to wait up to 60 mins for you to gain access after activation. So be patient!

3 Creating a project in Rahti

After you have gained access to **`Rahti`** service, you can login to **`Rahti`** at your CSC project's **`Projects`** view. Choose your (only) project and scroll down to **`Services`**. Clicking **`Login`** will take you to **`Rahti`** landing page.

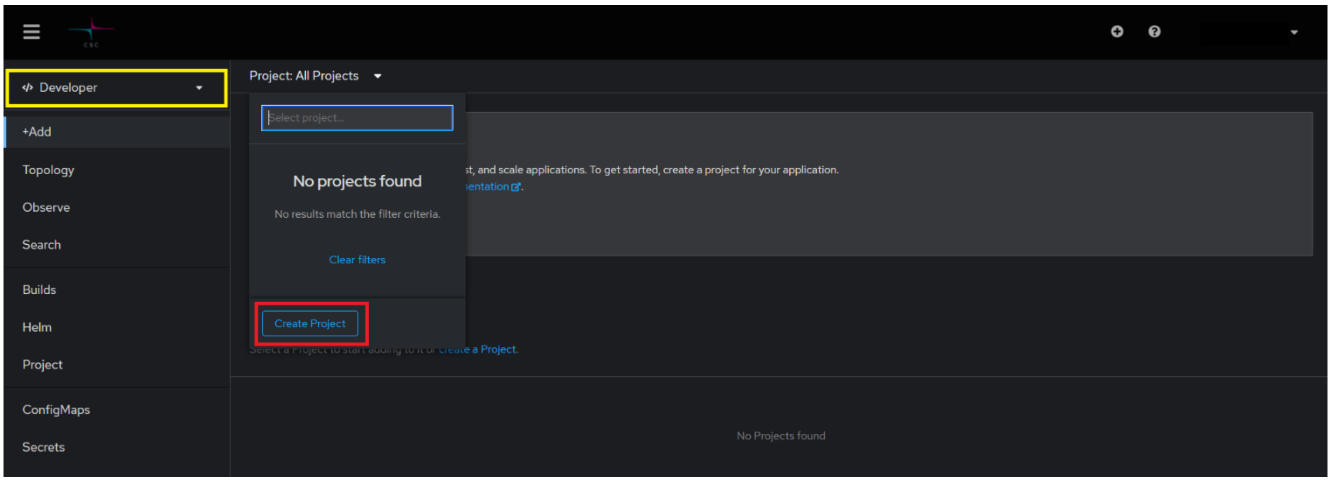


Proceed the login process, you need to click **`Login`** buttons a few times in different pages. Use either Haka or CSC as your authentication method when prompted.

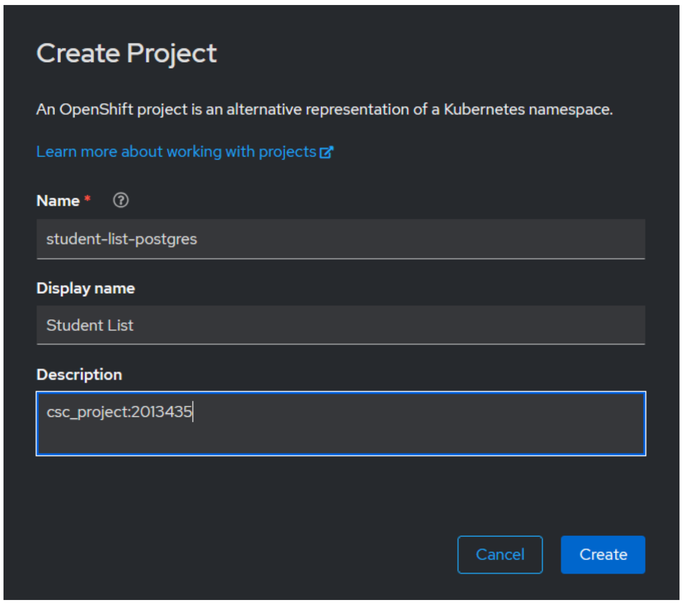
Successfully logging in directs you to **`Rahti`** console. When starting the console for the first time take the site's tour showing important navigation options. After the tour you are ready to create your **`Rahti project`**.

Click the **`Project: All projects`** dropdown menu and choose **`Create Project`** or click the text **`Create a Project`**. If you don't have these options visible in your page, make sure you are in **`Developer`** view (marked with yellow).

NOTE! You can have more than one **`Rahti projects`** in one **`CSC Project`**.


Developer
+Add
Topology
Observe
Search
Builds
Helm
Project
ConfigMaps
Secrets
Project: All Projects
Select project...
No projects found
No results match the filter criteria.
Clear filters
Create Project
...st, and scale applications. To get started, create a project for your application.
...nentation
Select a Project to start adding to it or create a Project.
No Projects found

Give your project a name and a display name. In the description write **`csc_project:<your CSC project number>`**. You will find your **`CSC project number`** in your CSC Project's **`Project information`** view. Proceed by clicking **`Create`**.



Create Project

An OpenShift project is an alternative representation of a Kubernetes namespace.

Learn more about working with projects

Name *: student-list-postgres

Display name: Student List

Description: csc_project:2013435

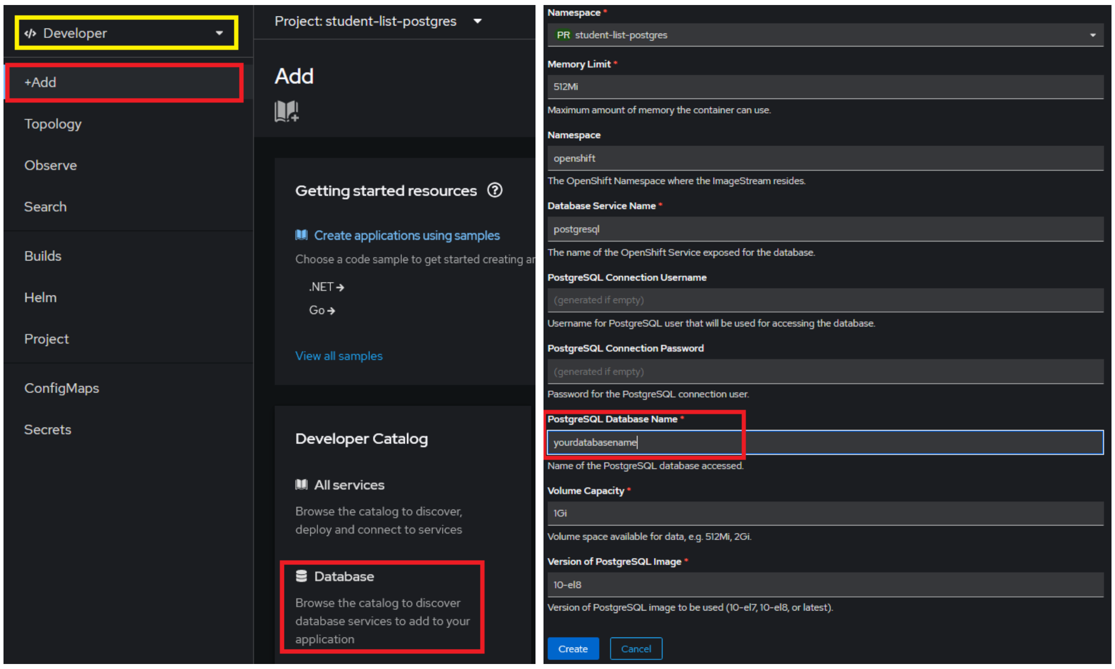
4 Creating a Postgresql database instance

After creating a project in Rahti it’s time to create a new database instance for your application.

In `Developer` view, go to `+Add` page and start creating a database for your project. Choose `PostgreSQL`. Make sure you **don’t select the Ephemeral** version! Click `Instantiate Template` to continue the process.

You can leave all the default values in the form where you create your database. Leaving `PostgreSQL Connection Username` and `Password` empty will make the system generate random credentials for you. This is fine within the scope of Haaga-Helia UAS course work.

If you want to specify `PostgreSQL Connection Username` and `Password`, **don’t use weak a password! Always use strong passwords even when practicing!**


Developer
+Add
Topology
Observe
Search
Builds
Helm
Project
ConfigMaps
Secrets
Project: student-list-postgres
Add
Getting started resources
Create applications using samples
Choose a code sample to get started creating a
.NET →
Go →
View all samples
Developer Catalog
All services
Browse the catalog to discover, deploy and connect to services
Database
Browse the catalog to discover database services to add to your application
Create
Cancel

Namespace *
PR student-list-postgres

Memory Limit *
512Mi
Maximum amount of memory the container can use.

Namespace
openshift
The OpenShift Namespace where the ImageStream resides.

Database Service Name *
postgresql
The name of the OpenShift Service exposed for the database.

PostgreSQL Connection Username
(generated if empty)
Username for PostgreSQL user that will be used for accessing the database.

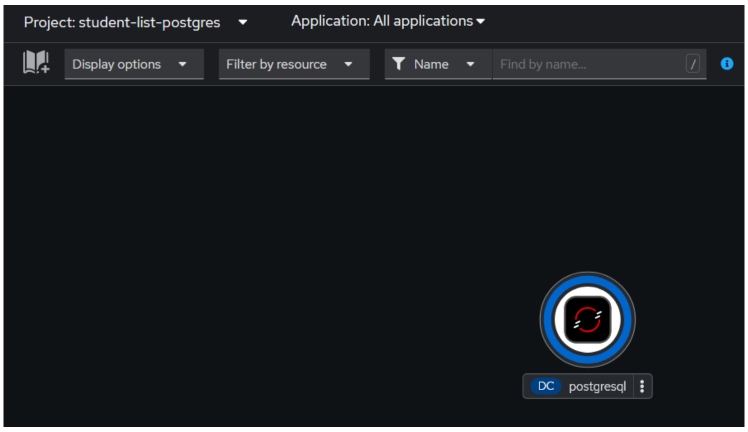
PostgreSQL Connection Password
(generated if empty)
Password for the PostgreSQL connection user.

PostgreSQL Database Name *
yourdatabasename
Name of the PostgreSQL database accessed.

Volume Capacity *
1Gi
Volume space available for data, e.g. 512Mi, 2Gi.

Version of PostgreSQL Image *
10-el8
Version of PostgreSQL image to be used (10-el7, 10-el8, or latest).

Successful creation of a database will show as a new `DeploymentConfig` object in your `Topology` view.

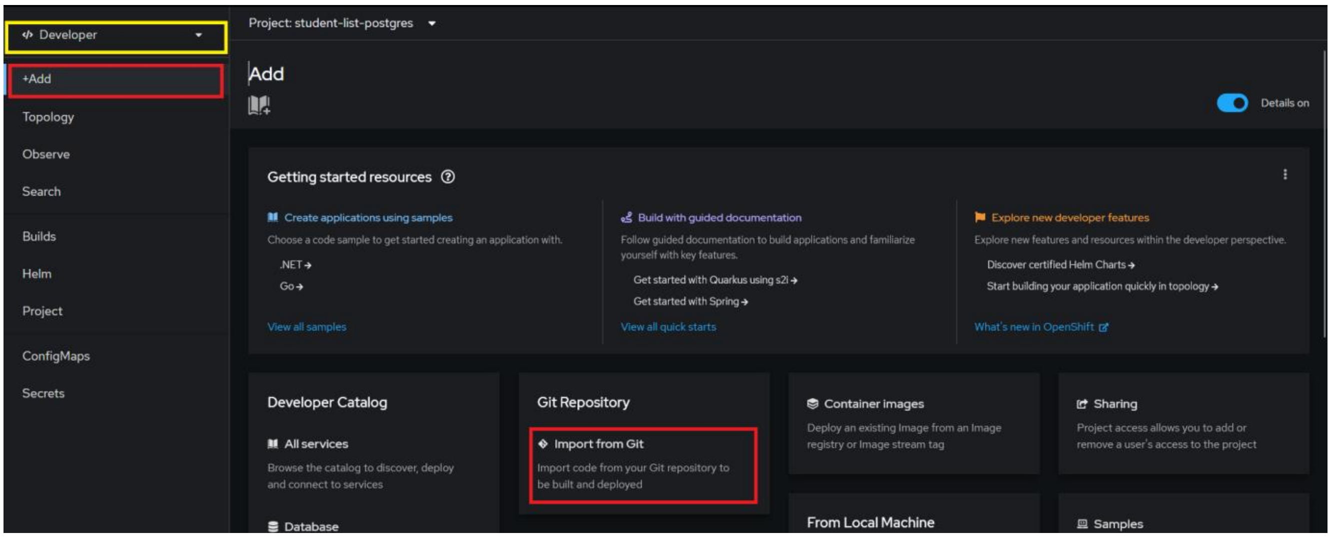


The next step is to include your SpringBoot application into your Rahti project.

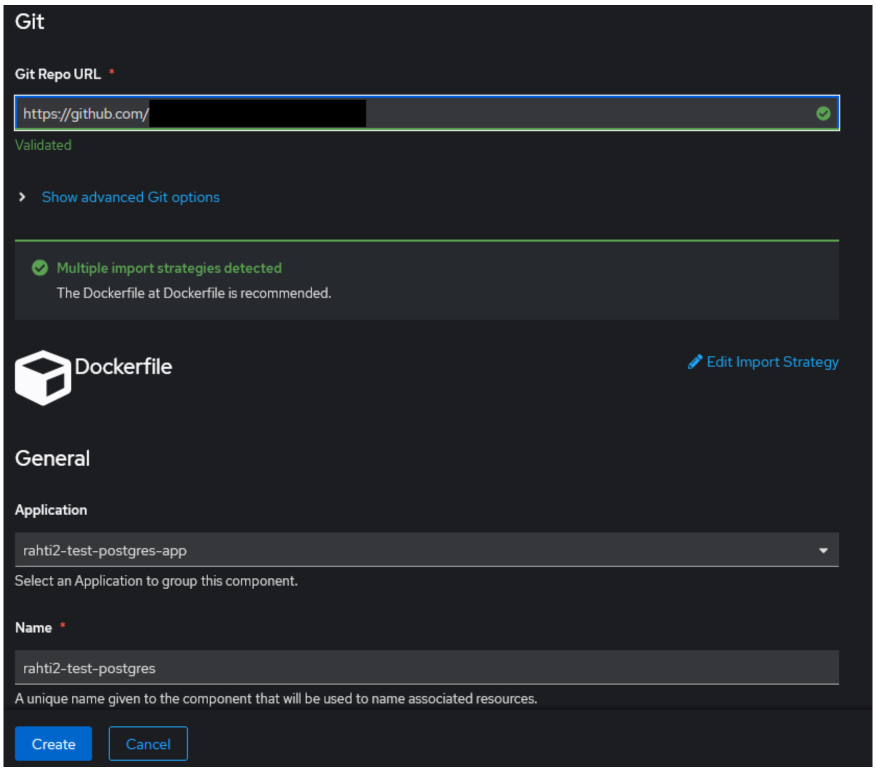
5 Including SpringBoot application into Rahti

You will need to have your SpringBoot application in GitHub before continuing these instructions. The following instructions assume that you have set your GitHub repository **`Public`**.

In **`Rahti`** dashboard make sure you are in **`Developer`** view: go to **`+Add`** page and choose **`Import from Git`**.


Developer
+Add
Topology
Observe
Search
Builds
Helm
Project
ConfigMaps
Secrets
Project: student-list-postgres
Add
Details on
Getting started resources
Create applications using samples
Choose a code sample to get started creating an application with.
.NET →
Go →
View all samples
Build with guided documentation
Follow guided documentation to build applications and familiarize yourself with key features.
Get started with Quarkus using s2i →
Get started with Spring →
View all quick starts
Explore new developer features
Explore new features and resources within the developer perspective.
Discover certified Helm Charts →
Start building your application quickly in topology →
What's new in OpenShift
Developer Catalog
All services
Browse the catalog to discover, deploy and connect to services
Database
Git Repository
Import from Git
Import code from your Git repository to be built and deployed
Container images
Deploy an existing Image from an Image registry or Image stream tag
Sharing
Project access allows you to add or remove a user's access to the project
From Local Machine
Samples

Copy and paste your GitHub repository's URL into the corresponding form field and choose **`Dockerfile`** as your **`Import Strategy`** if it is not the suggested one. You can leave rest of the form fields to their default values. Move on to **`Create`**.


Git

Git Repo URL *

https://github.com/[redacted]

Validated

Show advanced Git options

Multiple import strategies detected

The Dockerfile at Dockerfile is recommended.

Dockerfile

Edit Import Strategy

General

Application

rahti2-test-postgres-app

Select an Application to group this component.

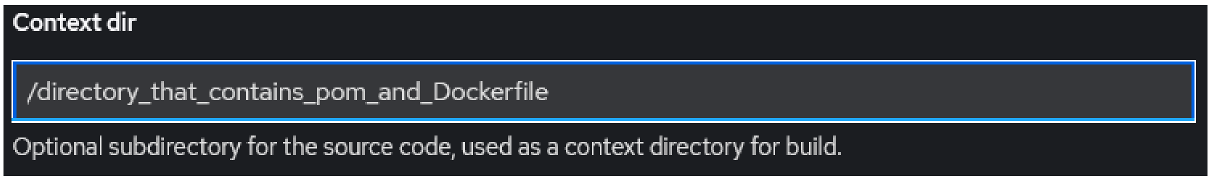
Name *

rahti2-test-postgres

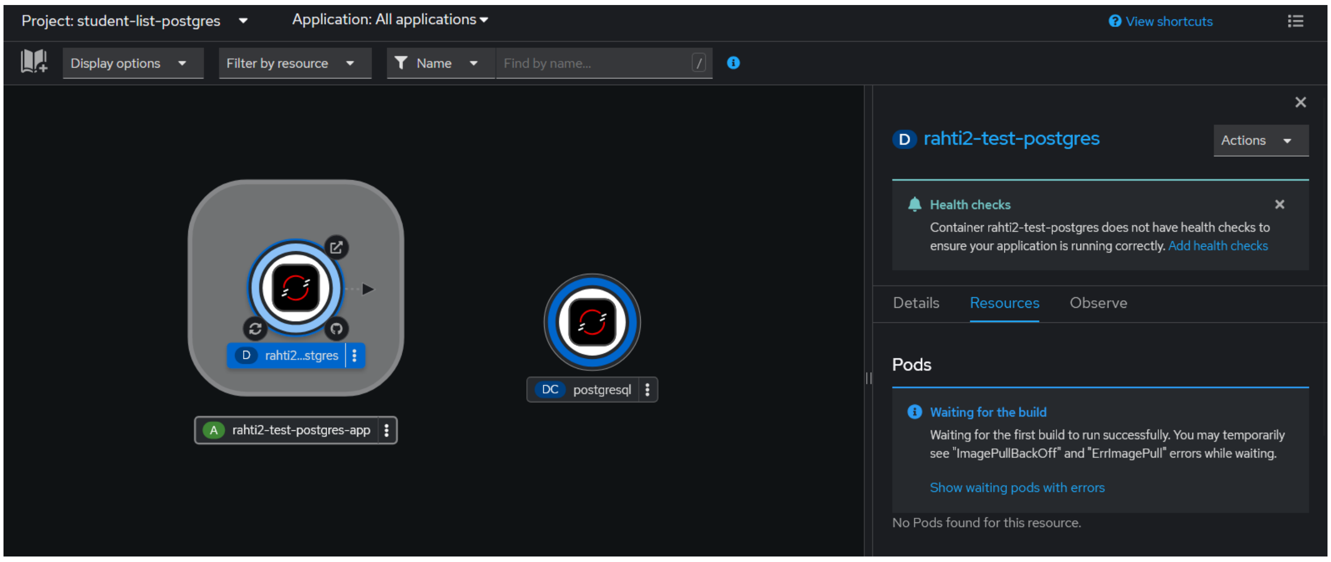
A unique name given to the component that will be used to name associated resources.

Create Cancel

NOTE! `Rahti` might not detect the `Dockerfile` if it is not in your GitHub project’s root. In this case open `Show advanced Git options` under `Git Repo URL` form field. In `Context dir` form field update the value to the name of the subdirectory that contains `pom.xml` and `Dockerfile`.


Context dir
/directory_that_contains_pom_and_Dockerfile
Optional subdirectory for the source code, used as a context directory for build.

Successful creation will take you to your project’s `Topology` view. Click the graphical representation of your deployment to open your deployment controls.



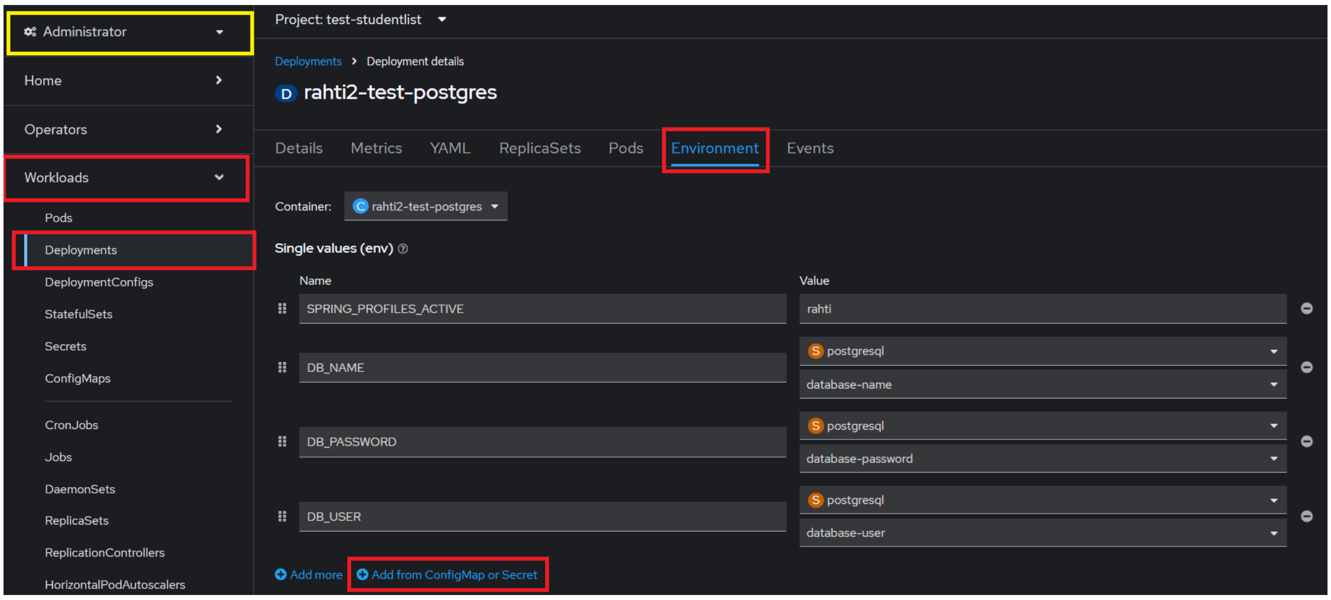
D rahti2-test-postgres Actions
Health checks
Container rahti2-test-postgres does not have health checks to ensure your application is running correctly. Add health checks
Pods
Waiting for the build
Waiting for the first build to run successfully. You may temporarily see "ImagePullBackOff" and "ErrImagePull" errors while waiting.
Show waiting pods with errors
No Pods found for this resource.

You have now deployed your application into `Rahti`. But it is not in working condition yet. You still need to configure the environment variables for the JDBC connection.

6 Configuring environment variables

In `Administrator` view: `Workloads` → `Deployments` → `<your deployment>` → `Environment`.

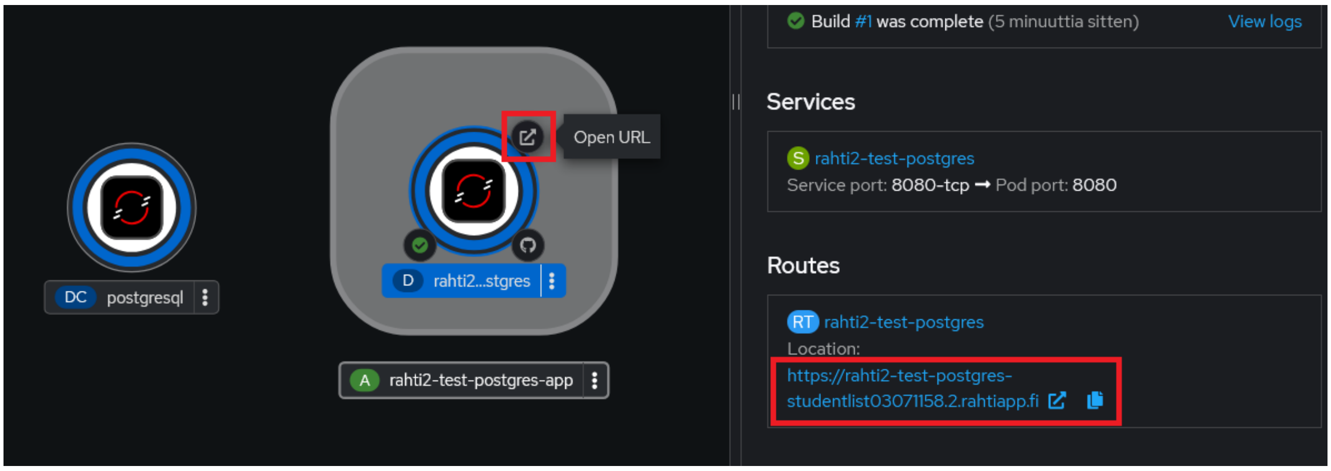
Set your environment variables as shown below. Click the circled `Add from ConfigMap or Secret` to get more form fields. Click `Save` at the bottom of the page to activate your changes.



Name	Value
SPRING_PROFILES_ACTIVE	rahti
DB_NAME	S postgresql / database-name
DB_PASSWORD	S postgresql / database-password
DB_USER	S postgresql / database-user

Congratulations! You have now deployed your Spring application with a proper database to CSC/Rahti!

To get the URL for your app go back to `Developer` view → `Topology`. Click your application's `Open URL` shortcut or navigate to deployment's quick controls (navigation pane on the right) and scroll down to `Routes`. It might take few minutes before you are able to access your deployment's URL in browser.


DC postgresql
Open URL
D rahti2...stgres
A rahti2-test-postgres-app
Build #1 was complete (5 minuuttia sitten)
View logs
Services
S rahti2-test-postgres
Service port: 8080-tcp → Pod port: 8080
Routes
RT rahti2-test-postgres
Location:
https://rahti2-test-postgres-studentlist03071158.2.rahtiapp.fi