**REST Services**

To use the rest services a /r/ is needed. For example : /webapps2019/r/project/supervisor1. If the object does not exist, an error is returned.

**Catalogue of files**

Web Pages

* index.xhtml
  + The index page. Has login buttons for admins, supervisors and students.
* Login.xhtml
  + Security page, used when a user is attempting to login. Has field for username and password.
* Loginfailed.xhtml
  + This page is the landing when a user has failed to log in, such as with invalid credentials.

*Admin*

* Administrator.xhtml
  + Is the hub page for an administrator. The landing page on a successful login.
* Adminregistration.xhtml
  + Used for registering a new system administrator by an admin.
* Studentprojectlist.xhtml
  + Displays a list of all students and their current projects/ proposed projects.
* Studentregistration.xhtml
  + Used for registering a new system student by an admin.
* Supervisorprojectlist.xhtml
  + Displays a list of all supervisors and their assigned projects
* Supervisorregistration.xhtml
  + Used for registering a new system supervisor by an admin.
* Transactionloginlogs.xhtml
  + Used to display all login transactions for students and supervisors.
* Transactionlogs.xhtml
  + Used to display all non-login transactions for students and supervisors.
* Topicregistration.xhtml
  + Used for creating a new topic. Used by both supervisor and admin

*Student*

* Student.xhtml
  + Is the hub page for a student. The landing page on a successful login.
* Projectselection.xhtml
  + Used for displaying all available projects that the student can select.
* Supervisorselection.xhtml
  + Used for creating and proposing a new project to a selected supervisor.

*Supervisor*

* Supervisor.xhtml
  + Is the hub page for a supervisor. The landing page on a successful login.
* Managedprojects.xhtml
  + Used to display all projects currently linked to this supervisor.
* Projectregistration.xhtml
  + Used to register a new project for the logged in supervisor.
* Supervisedstudents.xhtml
  + Used to display all accepted projects and the students on them.
* Topicregistration.xhtml
  + Used for creating a new topic. Used by both supervisor and admin

Java Files

*REST service*

* ApplicationConfig.java
  + Used for the application configuration. Here the REST services path is set to “/r”
* RSProject.java
  + Configs the path and methods for retreving a project.
* RSStudent.java
  + Configs the path and methods for retreving a student.
* RSSupervisor.java
  + Configs the path and methods for retreving a supervisor.

*Time stamping*

* TimeStamper.java
  + Autogenerated by thrift compiler.
* TimeStampHandler.java
  + The handler for the time stamp server.
* TimeStampBean.java
  + Used for fetching a time stamp from the time stamp server.
* TimeStampServer.java
  + The timestamp server bean. Handles the time stamping thread.

*Classes*

* Administrator.java
  + The Administrator class for each system administrator. Inherits from SystemUser.
* Project.java
  + The Project class for each project in the database.
* ProjectRequests.java
  + The ProjectRequests class used to track all project requests from student to supervisor.
* ProjectTopic.java
  + The ProjectTopic class for each project topic. A project can have multiple project topics.
* Student.java
  + The Student class for each system student. Inherits from SystemUser.
* Supervisor.java
  + The Supervisor class for each system supervisor. Inherits from SystemUser.
* SystemUser.java
  + The SystemUser class for each system user. The super class for each type of system user.
* SystemUserGroup.java
  + The SystemUserGroup class used for managing the types of SystemUser.
* TransactionLog.java
  + The TransactionLog class which holds all details for each student and supervisor transaction.

*EJB*

* InitialRegistration.java
  + Used for registering the default admin1.
* UserServiceStore.java
  + The main EJB for all business logic for the system.
* UserServiceStoreInterface.java
  + Used for allowing methods from the UserServiceStore to be accessed by the JSF beans.

*JSF Beans*

* InitialRegistrationBean.java
  + Used for initial registration of admin1.
* ListingBean.java
  + Used for requesting projects as well as lists of supervisors/students and their projects
* LoginBean.java
  + Used for logging in and out the current logged in user.
* LoginTransactionListingBean.java
  + Used for listing login transactions.
* NotificationBean.java
  + Used for handling notifications for the supervisor.
* ProjectRegistrationBean.java
  + Used for registering projects / project requests and topics.
* RegisterBean.java
  + Used for registering system users by the admin.
* TransactionListingBean.java
  + Used for listing non-login transactions.

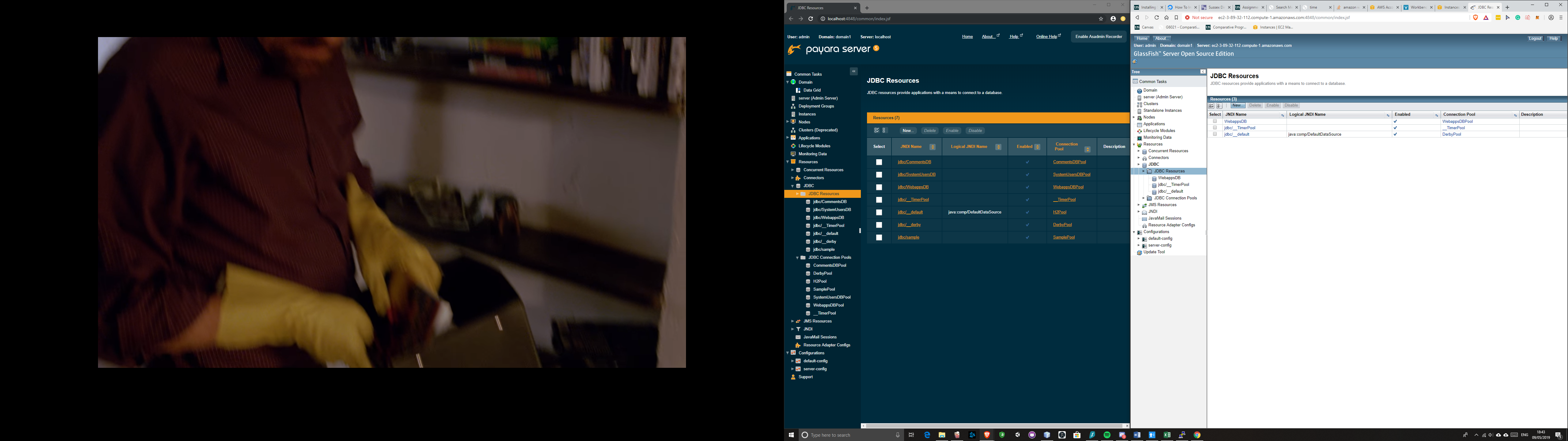
**Amazon AWS**

*Did not manage to successfully upload .war file. But completed all settings in the admin console.*

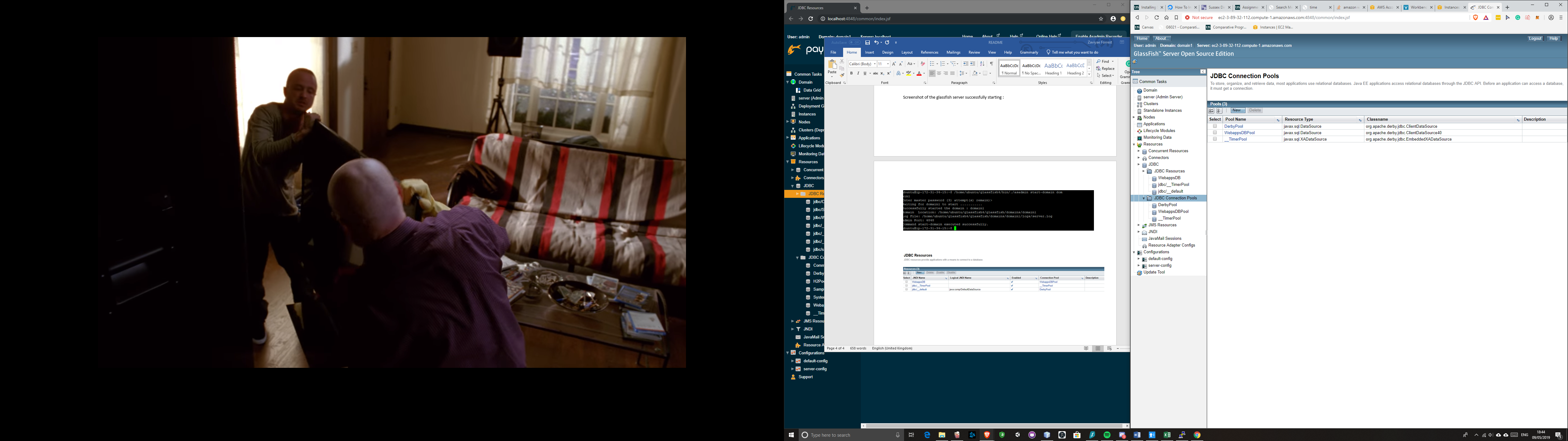
Screenshot of the glassfish server successfully starting :



Database :



Pool :



Realm :

