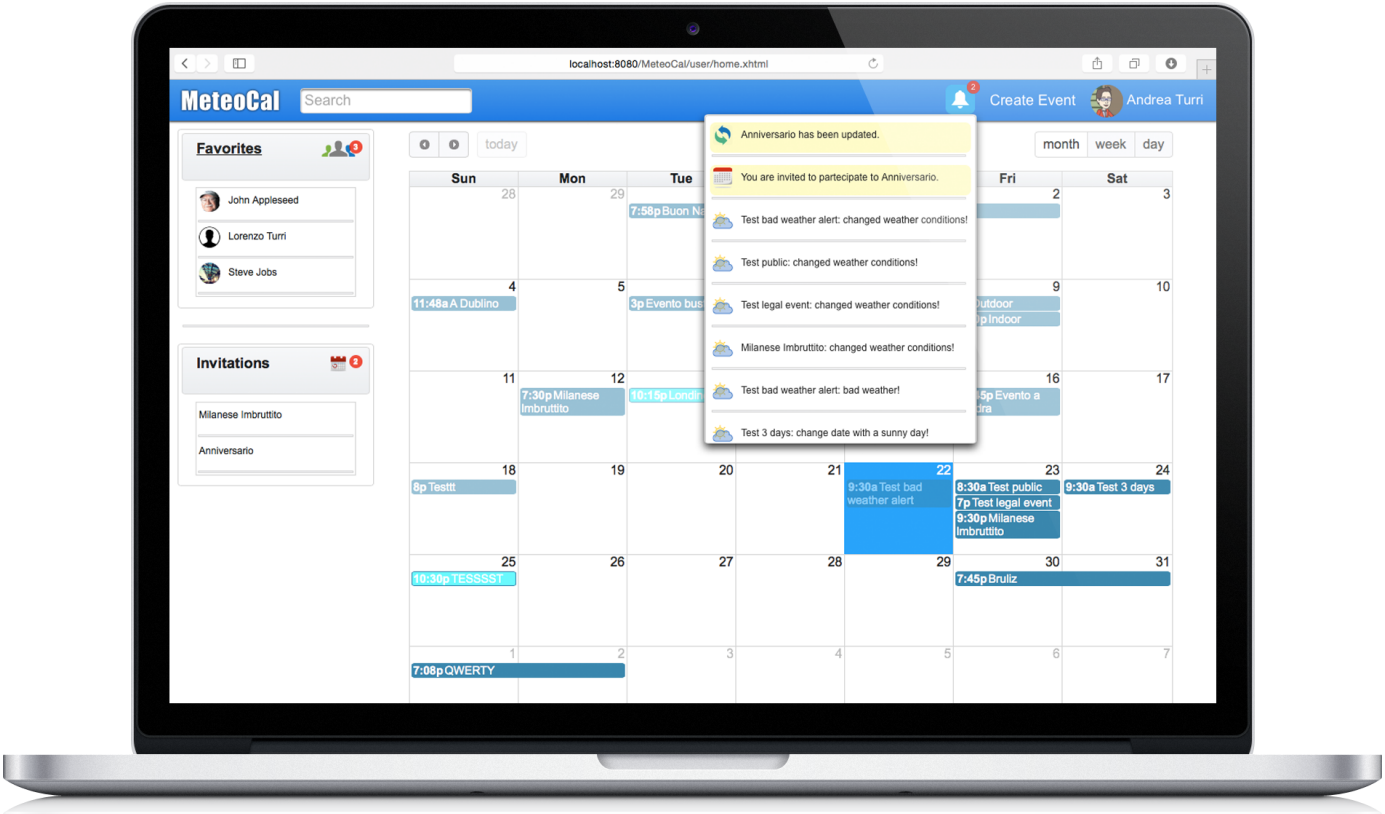


Installation manual

Andrea Turri
Andrea Salmoiraghi
Fabiano Riccardi

System requirements

Introduction

Before installing MeteoCal on your machine, please ensure that you have all the system requirements to let the application work properly.

Java

MeteoCal has been developed in JEE version 7 with JDK 1.8. In case you don't have it installed, you can download it from: http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

Glassfish Web Server

MeteoCal is a web application, so it needs Glassfish 4.1 Server to run. In case you don't have it installed, you can download it from:
https://glassfish.java.net/download.html (please, download Java EE 7 Full Platform version)

MySQL

MeteoCal needs a MySQL Server, version 5.6.21 (or above). You can download it from:
http://dev.mysql.com/downloads/mysql/

Browser

MeteoCal is currently developed for Webkit browsers, so **it is highly recommended that you open the application with Google Chrome** (version 38 or above) **or Apple Safari** (version 8 or above).

MeteoCal may not work properly on other browsers, in a future development we could add the support for them.

Setting up environment

Set up Glassfish

In order to create the Glassfish 4.1 Server please refer to the Glassfish official documentation (https://glassfish.java.net/).
Please remember that you have also to install MySQL Connector in order to allow the server to connect to MySQL Server, you can download it from:
http://dev.mysql.com/downloads/connector/j/ (N.B: In order to get the .jar file, download the compressed file)

Once downloaded, just add the .jar file to folder located at *<GlassFish-Installation-Path>/domains/<Domain-Name>/lib* (where *<Domain-Name>* is the name given during the installation of Glassfish, default is *domain1*).

After that you can start the server.

Start GlassFish

Just invoke, from the Glassfish directory, the *<GlassFish-Installation-Path>/bin/asadmin start-domain* command.
Note that since, you only have one domain configured, there is no need to mention which domain to start.

Now, you can open the Admin Console at:
http://localhost:4848
or the web server at:
http://localhost:8080

To stop GlassFish, you can use the *<GlassFish-Installation-Path>/bin/asadamin stop-domain* command.

Set up MySQL server

After installing MySQL you should access it and set up your root account.

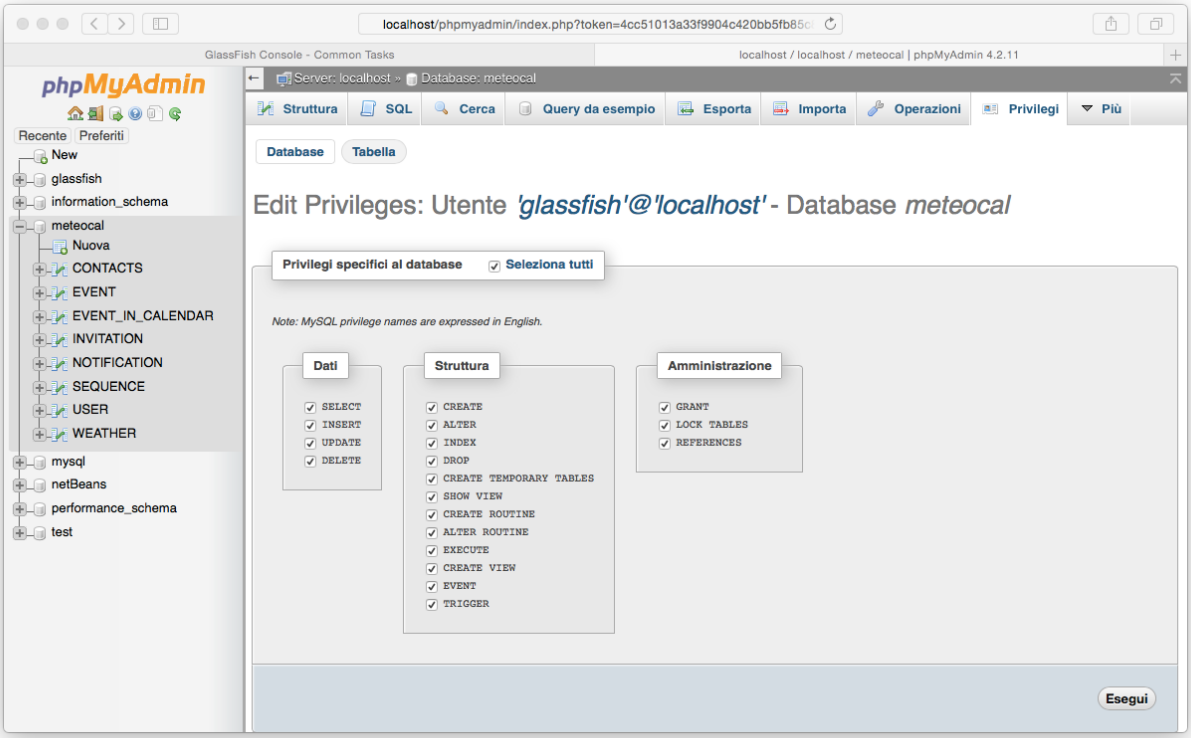
Then you can create a new database that must be called **meteocal** and a user with the following credentials:

- Username: **glassfish**
- Password: **glassfish**

Once created the database and the user, you have to give all privileges on the new database to the new user.

Ensure also that your MySQL server runs on the TCP port 3306.

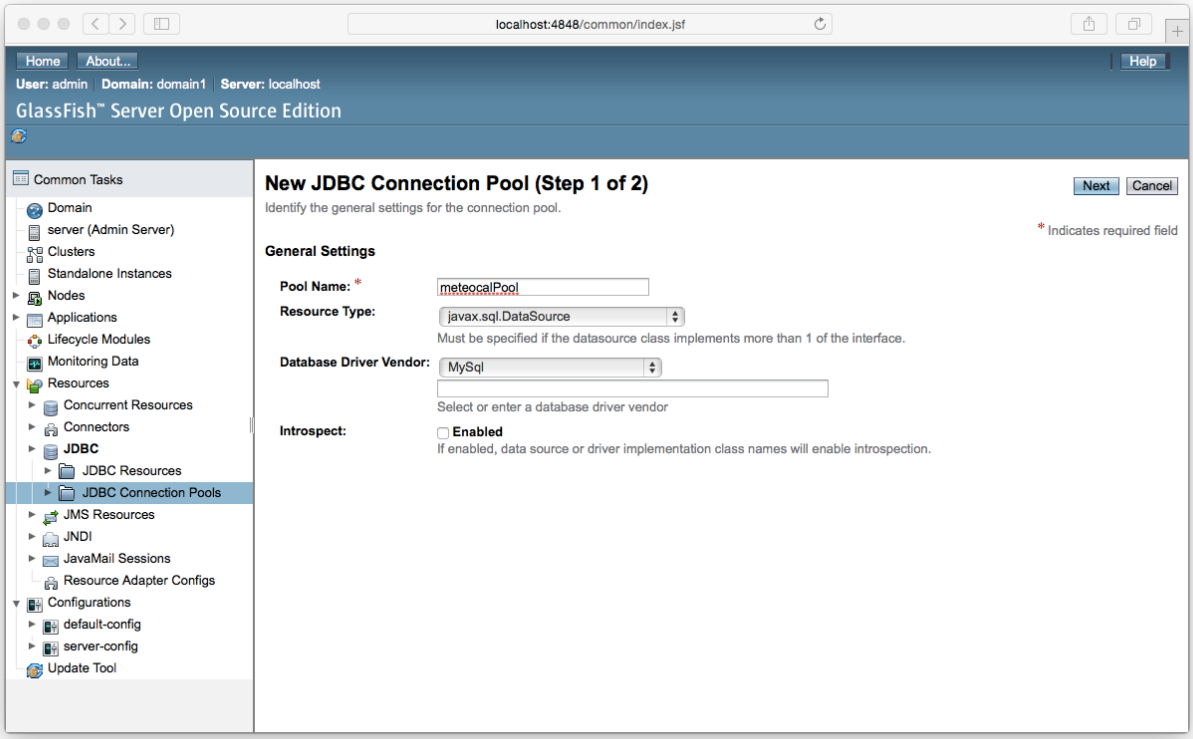
Please refer to MySQL documentation if you do not know how to perform these tasks.


localhost/phpmyadmin/index.php?token=4cc51013a33f9904c420bb5fb85
GlassFish Console - Common Tasks
localhost / localhost / meteocal | phpMyAdmin 4.2.11
phpMyAdmin
Recente Preferiti
New
glassfish
information_schema
meteocal
Nuova
CONTACTS
EVENT
EVENT_IN_CALENDAR
INVITATION
NOTIFICATION
SEQUENCE
USER
WEATHER
mysql
netBeans
performance_schema
test
Server: localhost » Database: meteocal
Struttura
SQL
Cerca
Query da esempio
Esporta
Importa
Operazioni
Privilegi
Più
Database
Tabella
Edit Privileges: Utente 'glassfish'@'localhost' - Database meteocal
Privilegi specifici al database
Seleziona tutti
Note: MySQL privilege names are expressed in English.
Dati
SELECT
INSERT
UPDATE
DELETE
Struttura
CREATE
ALTER
INDEX
DROP
CREATE TEMPORARY TABLES
SHOW VIEW
CREATE ROUTINE
ALTER ROUTINE
EXECUTE
CREATE VIEW
EVENT
TRIGGER
Amministrazione
GRANT
LOCK TABLES
REFERENCES
Esegui

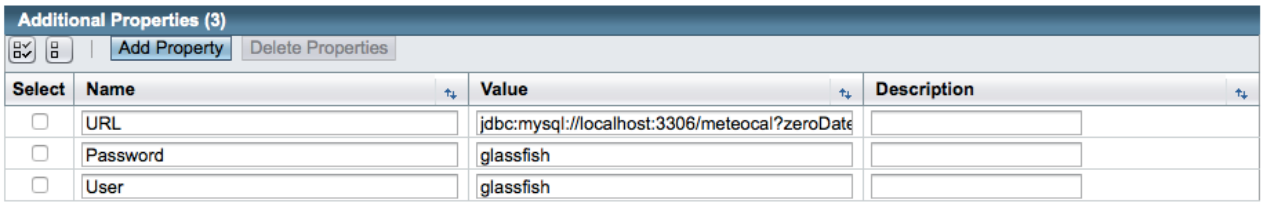
Set up JDBC Realm

Once started your Glassfish server, access to the admin area http://localhost:4848 and follow these steps:

- On the left bar, go to Resources > JDBC > JDBC Connection Pools
- Click on New
- Fill the form as in the following image and then click Next:



- Delete from the table of additional properties everything and leave only the following ones and click Finish:



URL: **jdbc:mysql://localhost:3306/meteocal?zeroDateTimeBehavior=convertToNull**

Once completed this procedure, you will see the new connection pool in the list. Click on it and ensure that you have all the properties set as it follows:

General | Advanced | Additional Properties

Edit JDBC Connection Pool

Save Cancel

Modify an existing JDBC connection pool. A JDBC connection pool is a group of reusable connections for a particular database.

Load Defaults Flush Ping

* Indicates required field

General Settings

Pool Name: meteocalPool

Resource Type: javax.sql.DataSource

Must be specified if the datasource class implements more than 1 of the interface.

Datasource Classname: com.mysql.jdbc.jdbc2.optional.MysqlDataSource

Vendor-specific classname that implements the DataSource and/or XADataSource APIs

Driver Classname:

Vendor-specific classname that implements the java.sql.Driver interface.

Ping: ☐ **Enabled**

When enabled, the pool is pinged during creation or reconfiguration to identify and warn of any erroneous values for its attributes

Deployment Order: 100

Specifies the loading order of the resource at server startup. Lower numbers are loaded first.

Description:

Pool Settings

Initial and Minimum Pool Size: 8 Connections

Minimum and initial number of connections maintained in the pool

Maximum Pool Size: 32 Connections

Maximum number of connections that can be created to satisfy client requests

Pool Resize Quantity: 2 Connections

Number of connections to be removed when pool idle timeout expires

Idle Timeout: 300 Seconds

Maximum time that connection can remain idle in the pool

Max Wait Time: 60000 Milliseconds

Amount of time caller waits before connection timeout is sent

Transaction

Non Transactional Connections: ☐ **Enabled**

Returns non-transactional connections

Transaction Isolation:

If unspecified, use default level for JDBC Driver

Isolation Level: ☑ **Guaranteed**

All connections use same isolation level; requires Transaction Isolation

Save Cancel

General | Advanced | Additional Properties

Edit JDBC Connection Pool Advanced Attributes

Save Cancel

Modify an existing JDBC connection pool. A JDBC connection pool is a group of reusable connections for a particular database.

Load Defaults

Pool Name: meteocalPool

Statement Timeout: -1 Seconds
Timeout property of a connection to enable termination of abnormally long running queries. -1 implies that it is not enabled.

Statement Cache Size: 0
Caching is enabled when set to a positive non-zero value (for example, 10)

Init SQL:
Specify a SQL string to be executed whenever a connection is created from the pool

SQL Trace Listeners:
Comma-separated list of classes that implement the org.glassfish.api.jdbc.SQLTraceListener interface

Wrap JDBC Objects: ☐ **Enabled**
When set to true, application will get wrapped jdbc objects for Statement, PreparedStatement, CallableStatement, ResultSet, DatabaseMetaData

Pooling: ☑ **Enabled**
When set to false, disables connection pooling for the pool

Connection Settings

Validate At Most Once: 0 Seconds
Specifies the time interval in seconds between successive requests to validate a connection at most once. Default value is 0, which means the attribute is not enabled.

Connection Leak Timeout: 0 Seconds
0 implies no connection leak detection

Connection Leak Reclaim: ☐
If enabled, leaked connection will be reclaimed by the pool after connection leak timeout occurs

Statement Leak Timeout: 0 Seconds
0 implies no statement leak detection

Statement Leak Reclaim: ☐
If enabled, leaked statement will be reclaimed by the pool after statement leak timeout occurs

Creation Retry Attempts: 0
Number of attempts to create a new connection. 0 implies no retries.

Retry Interval: 10 Seconds
Time interval between retries while attempting to create a connection. Effective when Creation Retry Attempts is greater than 0.

Lazy Association: ☐ **Enabled**
Connections are lazily associated when an operation is performed on them

Lazy Connection Enlistment: ☐ **Enabled**
Enlist a resource to the transaction only when it is actually used in a method

Associate with Thread: ☐ **Enabled**
When the same thread is in need of a connection, it can reuse the connection already associated with that thread

Match Connections: ☐ **Enabled**
Turns connection matching for the pool on or off

Max Connection Usage : 0
Connections will be reused by the pool for the specified number of times, after which they will be closed. 0 implies the feature is not enabled.

Connection Validation

Connection Validation: ☐ **Required**
Validate connections, allow server to reconnect in case of failure

Validation Method: auto-commit

Table Name: Populate Table Names
If table validation is selected, select or enter the table name.

Validation Class Name:
If custom-validation is selected, specify validation classname.

On Any Failure: ☐ **Close All Connections**
Close all connections and reconnect on failure, otherwise reconnect only when used

Allow Non Component Callers: ☐ **Enabled**
Enable the pool to be used by non-component callers such as Servlet Filters

Save Cancel

Edit JDBC Connection Pool Properties

Modify properties of an existing JDBC connection pool.

Save Cancel

Pool Name: meteocalPool

Additional Properties (3)

Add Property | Delete Properties

Select	Name	Value	Description
☐	URL	jdbc:mysql://localhost:3306/meteocal?zeroDate	
☐	Password	glassfish	
☐	User	glassfish	

Check that everything is ok by clicking on the ping button (MySQL Server must be turned on).

General | Advanced | Additional Properties

Ping Succeeded

Edit JDBC Connection Pool

Modify an existing JDBC connection pool. A JDBC connection

Load Defaults | Flush | Ping

Once ensured that the ping works, move to Resources > JDBC > JDBC Resources on the left bar. Here add a new Resource called **jdbc/meteocal** on the pool you created before. You should obtain this:

localhost:4848/common/index.jsf

Home | About... | Help

User: admin | Domain: domain1 | Server: localhost

GlassFish™ Server Open Source Edition

Common Tasks
- Domain
- server (Admin Server)
- Clusters
- Standalone Instances
- Nodes
- Applications
- Lifecycle Modules
- Monitoring Data
- Resources
  - Concurrent Resources
  - Connectors
  - JDBC
    - JDBC Resources
    - JDBC Connection Pools
  - JMS Resources
  - JNDI
  - JavaMail Sessions
  - Resource Adapter Configs
- Configurations
  - default-config
  - server-config
- Update Tool

Edit JDBC Resource

Edit an existing JDBC data source.

Load Defaults

Save Cancel

JNDI Name: jdbc/meteocal

Pool Name: meteocalPool

Use the JDBC Connection Pools page to create new pools

Deployment Order: 100

Specifies the loading order of the resource at server startup. Lower numbers are loaded first.

Description:

Status: ☑ Enabled

Additional Properties (0)

Add Property | Delete Properties

Select	Name	Value	Description
No items found.			

Now we are ready to create the JDBC Realm:

- Navigate to Configurations > server-config > Security > Realms on the left sidebar
- Create a new Realm named **jdbcRealmMeteoCal** and select as a class **com.sun.enterprise.security.auth.realm.jdbc.JDBCRealm**
- Add the following properties:

Properties specific to this Class

JAAS Context: *	jdbcRealm Identifier for the login module to use for this realm
JNDI: *	jdbc/meteocal JNDI name of the JDBC resource used by this realm
User Table: *	USER Name of the database table that contains the list of authorized users for this realm
User Name Column: *	email Name of the column in the user table that contains the list of user names
Password Column: *	password Name of the column in the user table that contains the user passwords
Group Table: *	USER Name of the database table that contains the list of groups for this realm
Group Table User Name Column:	 Name of the column in the user group table that contains the list of groups for this realm
Group Name Column: *	groupName Name of the column in the group table that contains the list of group names
Password Encryption Algorithm: *	MD5 This denotes the algorithm for encrypting the passwords in the database. It is a security risk to leave this field empty.
Assign Groups:	 Comma-separated list of group names
Database User:	 Specify the database user name in the realm instead of the JDBC connection pool
Database Password:	 Specify the database password in the realm instead of the JDBC connection pool
Digest Algorithm:	SHA-256 Digest algorithm (default is SHA-256); note that the default was MD5 in GlassFish versions prior to 3.1

- Save and restart the server.

Set up Email configuration

Once started your Glassfish server, access to the admin area http://localhost:4848 and follow these steps:

- Navigate to Resources > JavaMail Sessions on the left sidebar
- Create a new session named **mail/mailSession** and add the following properties:

JNDI Name:	mail/mailSession
Mail Host: *	smtp.gmail.com DNS name of the default mail server
Default User: *	afa.meteocal@gmail.com User name to provide when connecting to a mail server; must contain only alphanumeric, und
Default Sender Address: *	afa.meteocal@gmail.com E-mail address of the default user
Deployment Order:	100 Specifies the loading order of the resource at server startup. Lower numbers are loaded first.
Description:	 Makes it easier to find this session later
Status:	☑ **Enabled**

Advanced

Store Protocol: imap
Either IMAP or POP3; default is IMAP

Store Protocol Class: com.sun.mail.imap.IMAPStore
Default is com.sun.mail.imap.IMAPStore

Transport Protocol: smtp
Default is SMTP

Transport Protocol Class: com.sun.mail.smtp.SMTPTransport
Default is com.sun.mail.smtp.SMTPTransport

Debug: ☐ **Enabled**
Select the Debug checkbox to enable extra debugging output for this mail session, including a protocol trace.

Additional Properties (5)

Add Property | Delete Properties

Select	Name	Value	Description
☐	mail.smtp.socketFactory.port	465	
☐	mail.smtp.port	465	
☐	mail.smtp.auth	true	
☐	mail.smtp.password	meteocalafa	
☐	mail.smtp.socketFactory.class	javax.net.ssl.SSLSocketFactory	

- Save and restart the server.

WARNING: it may happen that Google, the provider for the email service, blocks the account because of security reasons (e.g. unknown IP, too many emails…).

In this case MeteoCal will no longer send emails, until you log in from your browser in Gmail with credentials above and you unlock the account.

Please note that for unblocking the account it may take some hour.

Application deployment

Now we only have to deploy the application on our configured Glassfish Server.
To do that, download the file from:
https://code.google.com/p/meteocal-aturri-asalmoiraghi-friccardi/source/browse/Deliveries/MeteoCal.war

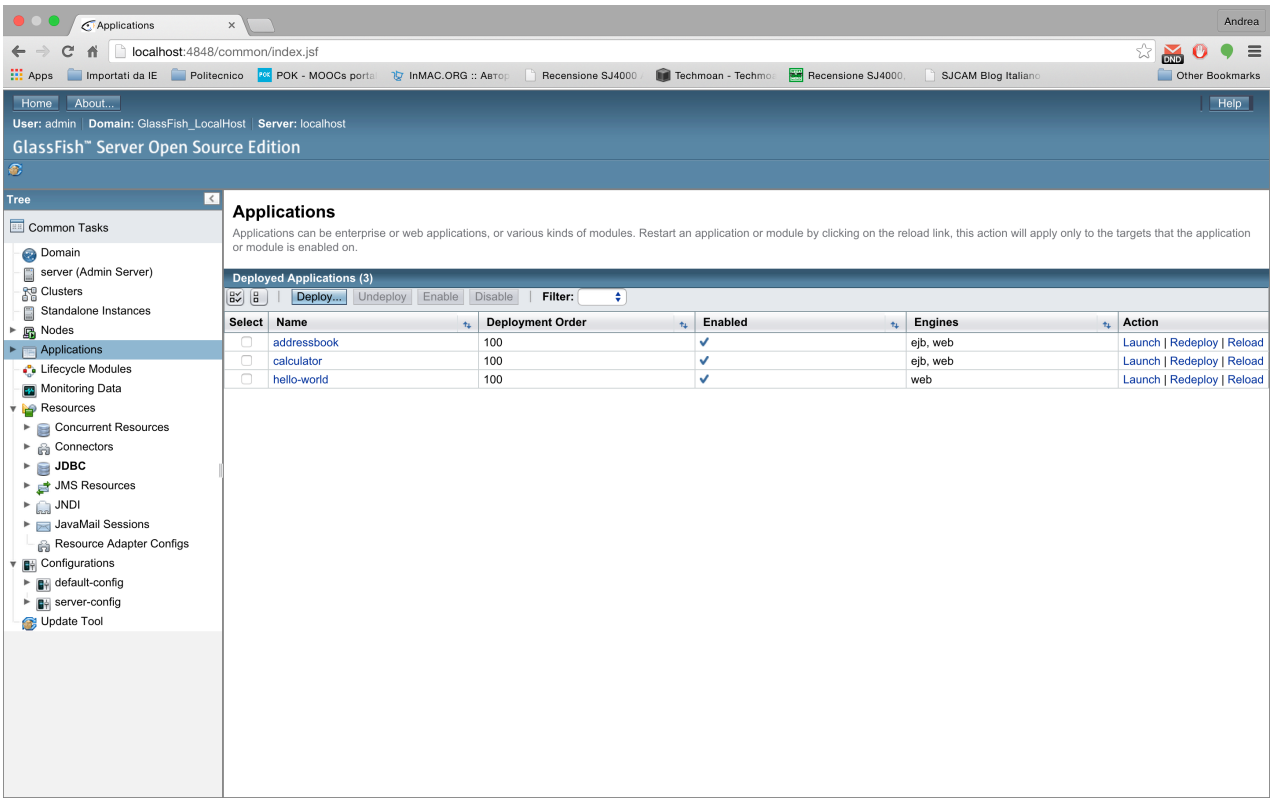
You can follow one of the following alternatives.

Autodeploy

Add the file downloaded in the at at *<GlassFish-Installation-Path>/domains/<Domain-Name>/autodeploy.*

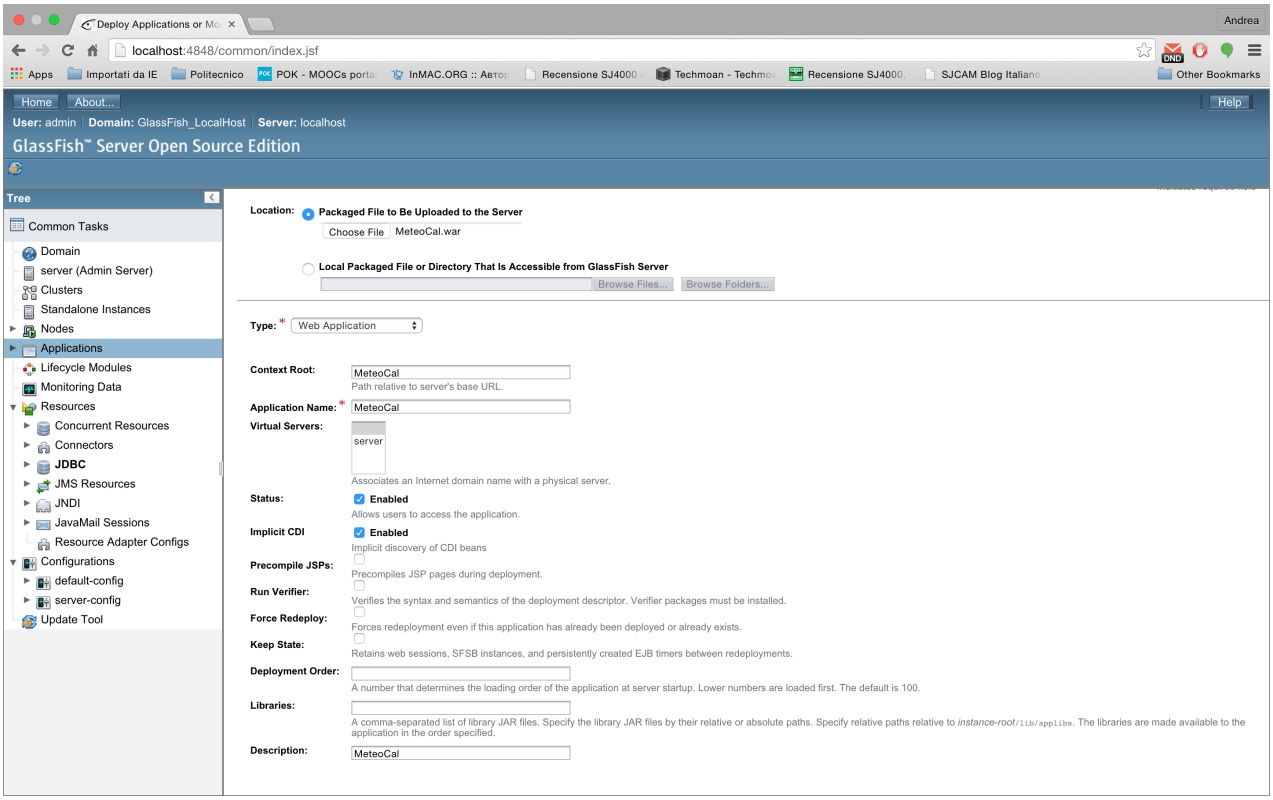
Manual deploy

Open the Glassfish Admin Console and go to Applications.

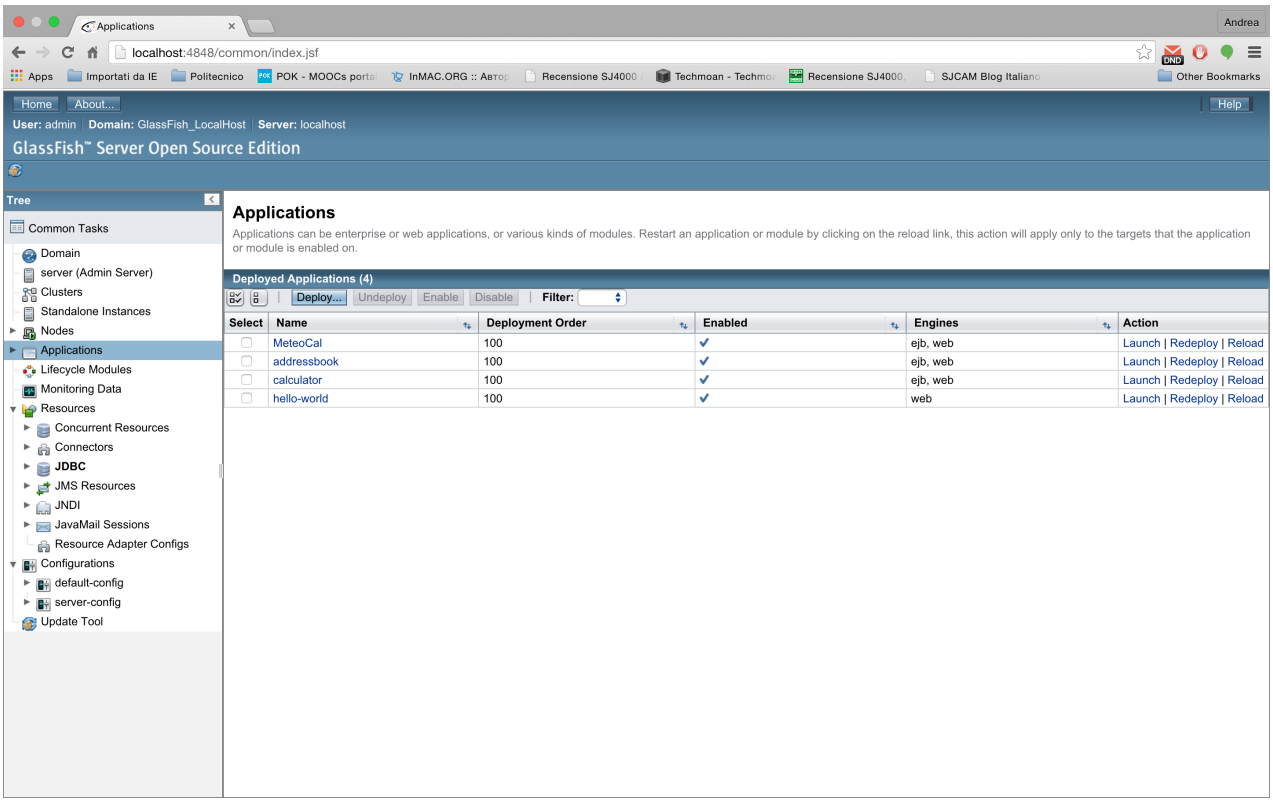


Select the *Deploy* button, to deploy a new application.

In the new window, select *Choose File* to open the downloaded file *MeteoCal.war*.
Add in the *description* field MeteoCal, and click ok to finish.



If all is done correctly, you should see the following window:



Executing the application

Congratulations, the configuration is done.

Now, to enjoy the app restart the server and go to:
http://localhost:8080/MeteoCal

Resolving issues

If you have some issues during the Glassfish server configuration, you can try to download our configuration for the server at:
https://code.google.com/p/meteocal-aturri-asalmoiraghi-friccardi/source/browse/Deliveries/glassfishConfig.zip

Extract the file domain.xml from the archive and put it at: *<GlassFish-Installation-Path>/domains/<Domain-Name>/config.*

NB: Remember to backup the original domain.xml, just rename it to domain.xml.old

Sample database

We provide a sample database that you can download at: https://code.google.com/p/meteocal-aturri-asalmoiraghi-friccardi/source/browse/Deliveries/sample.sql

You can import it into the “meteocal” database that you created previously. Please note that the database should be empty before importing the file that creates the tables and inserts some example data.

There are three users:

- atecwd+fab@gmail.com with password “meteocal10”
- atecwd+salmo@gmail.com with password “meteocal10”
- atecwd+sss@gmail.com with password “meteocal10”