



## Structured overlay Networks Application Platform

**SNAP**

Installation Manual

---

Carles Pairot Gavaldà - <[carles.pairot@urv.net](mailto:carles.pairot@urv.net)>  
Architecture and Telematic Services Research Group  
Universitat Rovira i Virgili  
Tarragona  
Spain

---



## Installing SNAP

When finishing downloading SNAP from its website (<http://snap.objectweb.org>) you may proceed decompressing the available ZIP file in any directory of your choice. Such directory will be SNAP's and Jetty webserver's main directory, and will be referred as `$SNAP_DIR`.

Before starting SNAP we must configure some basic settings which allow our machine to be able to help bootstrapping newcoming SNAP nodes. Such settings can be found on `$SNAP_DIR/snap/WEB-INF/dermi-config.xml` file.

A sample `dermi-config.xml` file is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
  <entry key="dermi.host">planetlab.urv.net</entry>
  <entry key="dermi.port">5009</entry>
  <entry key="dermi.sessionClass">dermi.session.pastry.Session</entry>
  <entry key="dermi.communityID">gojiita</entry>
  <entry key="dermi.underlyingDHT">pastry</entry>
  <entry key="dermi.classStore">c://class_store/</entry>
  <entry key="dermi.className">dermi.samples.simple.SimpleImpl</entry>
  <entry key="dermi.classID">d8ecf6c6-1370-474a-91d6-d26a66793023</entry>
  <entry key="BUNSHIN_STORAGE_MANAGER">bunshin.storage.DiskStorage</entry>
  <entry key="BUNSHIN_STORAGE_ROOT_DIR"></entry>
  <entry key="BUNSHIN_REPLICA_FACTOR">4</entry>
  <entry key="BUNSHIN_CACHE">FALSE</entry>
  <entry key="BUNSHIN_DEBUG">FALSE</entry>
  <entry key="BUNSHIN_ID_APPLICATION">bunshin</entry>
  <entry key="BUNSHIN_REFRESH_TIME_SEC">5</entry>
</properties>
```

Mainly, the most important parameters are the first two, which need to be pointed to the server which will act as the *bootstrap* of the SNAP network. The bootstrap is normally the first node in the network which will be activated. Naturally, if it is the first time, we will usually mark ourselves as bootstraps, and all other nodes will have this or another already joined node as bootstrap. Recall that all other nodes will join the network ring via the specified bootstrap. Therefore, in the *dermi.host* and *dermi.port* properties we need to indicate the SNAP network's bootstrap data.

Before starting SNAP, we should configure Jetty's HTTP port (if different from the default **33333**). To do so, we should edit `$SNAP_DIR/conf/jetty.xml` and change the bolded line, as seen below:



```
<!-- ===== -->
<!-- Configure the Request Listeners -->
<!-- ===== -->

<!-- - - - - - -->
<!-- Add and configure a HTTP listener to port 8080 -->
<!-- The default port can be changed using: java -Djetty.port=80 -->
<!-- - - - - - -->
<Call name="addListener">
  <Arg>
    <New class="org.mortbay.http.SocketListener">
      <Set name="Port">
        <SystemProperty name="jetty.port" default="33333"/>
      </Set>
      <Set name="PoolName">P1</Set>
      <Set name="MinThreads">10</Set>
      <Set name="MaxThreads">200</Set>
      <Set name="MaxIdleTimeMs">30000</Set>
      <Set name="lowResources">50</Set>
      <Set name="LowResourcePersistTimeMs">2000</Set>
      <Set name="acceptQueueSize">0</Set>
      <Set name="ConfidentialPort">33443</Set>
      <Set name="IntegralPort">33443</Set>
    </New>
  </Arg>
</Call>
```

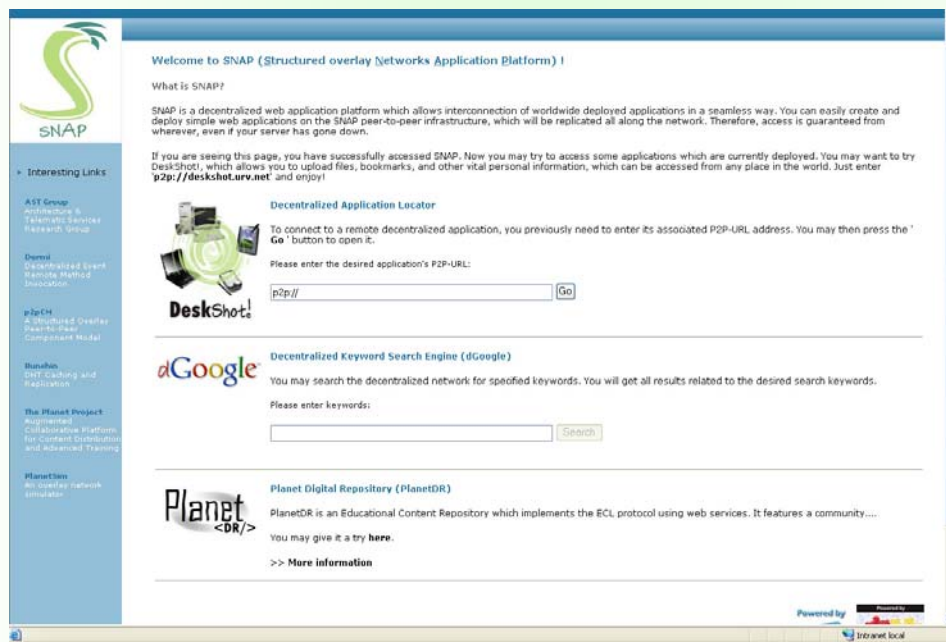
It is now a good moment to start the Jetty webserver, and try if SNAP works. To do so, we must invoke the **jetty.bat** or **jetty.sh** script, watching out that **java.exe** is on the system path. If not, the initialization script (jetty.\*) can be modified accordingly to match java.exe's current directory.

**IMPORTANT NOTE:** Java 1.5.0 SE onwards is required to run SNAP.

Once invoked, we can use the following URL scheme on the browser to access SNAP:

**http://hostname:hostport/**

If everything went successfully, a web page similar to this should be seen:



The most important section of SNAP's main webpage is the **Decentralized Application Locator**, which allows already deployed web application access in a decentralized way. The only previous requirement to access any application is that it has previously been signed and deployed onto SNAP. The first time SNAP is installed, no applications are currently deployed except SNAP's main webapp.

It is now time to learn how to develop and deploy your own web applications onto SNAP, so you may take a look at the "**Developer's Manual**".