AAL-2009-2-116

Deliverable D5.1 Appendix Elder-Spaces Application Installation Guide for CSCM

Deliverable Type: **(CO)***
Nature of the Deliverable: **(S)****
Date: **27/12/2013**
Distribution: **WP5**
Code: **D 5.1**
Editor: **ORIGO**
Contributors: **ORIGO**

Deliverable Type: *PU= Public, RE= Restricted to a group specified by the Consortium, PP= Restricted to other program participants (including the Commission services), CO= Confidential, only for members of the Consortium (including the Commission services)*

** ***Nature of the Deliverable***: *P= Prototype, R= Report, S= Specification, T= Tool, O= Other*

The ELDER-SPACES Consortium consists of:

BYTE	Project Coordinator	Greece
ORIGO	Partner	Hungary
FTB	Partner	Germany
e-Trikala	Partner	Greece
SEMMELWEIS	Partner	Hungary
SLG	Partner	Greece
CYBION	Partner	Italy

This page has been intentionally left blank.

Table of Contents

Table of Contents 3

1. Prerequisites 4
  1.1 Components 4
  1.2 Requirements 4
  1.3 Environment 4

2. Installation 4
  2.1 CSCM and configuration groups 4
  2.2 Installation 5
    2.2.1 Manually installed items 6
    2.2.2 Firewall 7

3. Maintenance 9

1. Prerequisites

1.1 Components

- PHP-based web application
- Java-based web service in Tomcat container

1.2 Requirements

- Ubuntu Linux (at least 10.04 LTS)
- Java Runtime Environment 1.6
- Tomcat Servlet Container (at least 6.0.35)
- PHP module, cgi, fastcgi, or fpm (at least 5.3.19 and we recommend fpm)
- http server application (we recommend nginx)
- Redis server (required version: 2.4.6)
- Mysql server (database server, we recommend MariaDB 5.5)
- Memcached
- Openmq (message queue broker and consumer)
- Wiwd (proprietary in-memory neural-network database server)
- Oracle database connection to iWiW system

These applications are used from CSCM repository.

1.3 Environment

Elder-Spaces designed to run on CSCM infrastructure (Centralized Software and Configuration Management for iWiW server environment). Also Elder-Spaces relies heavily on currently existing iWiW infrastructure. In many cases it uses iWiW services to serve requests (for example image and thumbnail storage, or the gadget services). We do not cover iWiW infrastructure setup guide in this document.

2. Installation

The following steps will allow you to run Elder-Spaces in a scalable HA environment. We create the configuration groups then we add servers to these groups. If we run out of resources on a server later we can add more to the required group in a few minutes.

In CSCM the it’s possible to merge groups (apply more config groups for one server) so we will use our existing configuration groups when it’s possible.

2.1 CSCM and configuration groups

In this chapter we use the currently existing config-iwiw configurations as base for our new groups. We will only describe the most important things you should be aware of and the deviations. The most important deviation is that we cloned config-iwiw-approval to config-Elder-Spaces to have distinct configurations what we can modify anytime without interfering iWiW systems.

Configuration groups contain “patches”. Patch means a specific version of an application and the related configuration templates for them. In somewhat simplified terms it’s an application & their configuration generated from templates. The most important parts of CSCM is the underlying NFS export and /sbin/v-syspatch and /sbin/v-sysconfig applications. v-sysconfig can generate configuration files from templates and v-syspatch manages attaching patches (mounting), calling v-sysconfig and starting the applications. Also v-syspatch can manage local or remote storage mounts for the patches.

These groups should not be changed now. We will use them as the base for our server then we install the required application and files.

We use the following configuration groups:

- config-hosting/hosting
  Base group for web application hosting.
- config-hosting/hosting-app
  Group for PHP web application hosting with nginx http server.
- config-Elder-Spaces/iwiw
  Base group for iWiW services.
- config-Elder-Spaces/iwiw-chat
  Group for XMPP chat service (Java, Tigase).
- config-Elder-Spaces/iwiw-api
  Group for web services api servlet (Java, Tomcat).
- config-Elder-Spaces/iwiw-openmq
  Group for message queue broker (Java, Openmq).
- config-Elder-Spaces/iwiw-listener
  Group for message queue consumer (Java, Openmq).
- config-Elder-Spaces/iwiw-wiwd
  Group for Wiwd server.
- config-Elder-Spaces/iwiw-memcache
  Group for Memcache in-memory key-value store.
- config-Elder-Spaces/iwiw-redis
  Group for Redis persistent key-value store and public/subscribe messaging server.
- config-Elder-Spaces/iwiw-mysql
  Group for the Mysql database server. In this case we use MariaDB 5.2 but also possible to use a newer version or Percona Server or Oracle Mysql Server.

2.2 Installation

First install Ubuntu Linux on the node. After it’s booted and it has it’s required networking configuration (connected to the internal network) copy CSCM applications to the server (/sbin/v-syspatch, /sbin/v-sysconfig) from cscm.iwiw.

Now the local part of CSCM is installed but it should be configured to use cscm.iwiw as it’s central configuration repository. Mount CSCM export to the following location: /srv/storage/cscm. Repository located at cscm.iwiw:/srv/storage/cscm/export. The required line in fstab:

```
cscm:/srv/storage/cscm/export /srv/storage/cscm nfs defaults,ro,nolock 0 0
```

Now we should configure what configuration groups to use on this server. This can be achieved by adding new options to the kernel command line in out boot loader. For GRUBv1 it can be achieved by modifying /boot/grub/menu.lst. The new parameters are: v-config=1, v-groups=…, v-confroot=/srv/storage/cscm/config-Elder-Spaces. A new configuration line in menu.lst looks like this:

```
kernel /boot/vmlinuz-2.6-xenU root=/dev/md1 ro v-config=1 v-groups=iwiw,iwiw-app v-confroot=/srv/storage/cscm/config-Elder-Spaces
```

Of course other parameters can vary from server to server like the root filesystem device, the kernel version or anything else. What we aware of now is the CSCM configuration parameters.

After everything is configured you should set up v-syspatch to start our patches on boot and stop them at shutdown by creating the required symbolic links to /etc/rc*.d:

```
ln -s /sbin/v-syspatch /etc/rc0.d/K10-vsyspatch
ln -s /sbin/v-syspatch /etc/rc1.d/K10-vsyspatch
ln -s /sbin/v-syspatch /etc/rc2.d/S99-vsyspatch
ln -s /sbin/v-syspatch /etc/rc3.d/S99-vsyspatch
ln -s /sbin/v-syspatch /etc/rc4.d/S99-vsyspatch
ln -s /sbin/v-syspatch /etc/rc5.d/S99-vsyspatch
ln -s /sbin/v-syspatch /etc/rc6.d/K10-vsyspatch
```

Let’s continue by doing our first boot to our CSCM system. At the first boot CSCM will create and initialise it’s required new filesystems, it deploys required applications and configurations then starts them.

2.2.1 Manually installed items

The Mysql database is initialised and you will have a working Mysql server but the database schema should be installed by hand. Create databases and users for your Elder-Spaces schemas, get the required schema files from http://svn.virgo.private/iwiw/branches/Elder-Spaces/database/ and load it to the new databases. For more information about this read the Mysql manual at http://dev.mysql.com/. The following schemas should be created and filled:

- activity, activity-sh1, activity-sh2
- api-commons-sh1, api-commons-sh2
- apiswitch
- appreg
- apps, appsdata-sh1, appsdata-sh2
- club-activity
- connnote
- events
- gamemail
- images-sh1, images-sh2, images-sh3
- kedvencek-sh1, kedvencek-sh2
- lifelong
- loginlog
- meme
- messages-sh1, messages-sh2, messages-sh3, messages-sh4
- notification-sh1, notification-sh2

- polls
- son
- st-cl1, st-cl2, st-pl, st-top
- travel
- usergroup

For sharded schemas you should install it’s init.sql file to all shards (sharded database names are suffixed by -sh*).

Database configuration in you web application is found at ${PATCH}/etc/*/Elder-Spaces/persistence-*.properties.

2.2.2 Firewall

After you have all your servers running you should configure your firewall to dispatch connections from internet to your application servers. We use haproxy for this. You will see that as we already wrote that some parts of the system use iWiW services. Configure it to bypass requests by the following schema:

```
frontend …
        acl Elder-Spaces.iwiw.hu-header-acl        hdr_end(Host) -i Elder-Spaces.iwiw.hu Elder-Spaces.com www.Elder-Spaces.com
        acl static.Elder-Spaces.iwiw.hu-header-acl hdr_end(Host) -i static.Elder-Spaces.iwiw.hu static.Elder-Spaces.com
        acl images.Elder-Spaces.iwiw.hu-header-acl hdr_end(Host) -i images.Elder-Spaces.iwiw.hu images.Elder-Spaces.com
        acl thn1.Elder-Spaces.iwiw.hu-header-acl   hdr_end(Host) -i thn1.Elder-Spaces.iwiw.hu thn1.Elder-Spaces.com
        acl api.Elder-Spaces.iwiw.hu-header-acl hdr_end(Host)    -i api.Elder-Spaces.iwiw.hu api.Elder-Spaces.com
        acl gadget.Elder-Spaces.iwiw.hu-header-acl hdr_end(Host) -i gadget.Elder-Spaces.iwiw.hu gadget.Elder-Spaces.com
        acl url-elder-go-to-java-acl                  path_beg /i/regisztracio /pages/user/registration.jsp /i/elfelejtettem-a-jelszavam /pages/user/forgottenpassword.jsp /pages/auth/authorize.jsp /pages/user/regconfirm.jsp /i/felhasznalasi-feltetelek /pages/misc/tos.jsp /i/profil/jelszo /pages/user/passwordchange.jsp /i/belepes /pages/user/login.jsp /pages/user/passwordcheck.jsp
        acl url-elder-go-to-nginx-acl                 path_beg /
        use_backend static.Elder-Spaces.iwiw.hu-bck if static.Elder-Spaces.iwiw.hu-header-acl
        use_backend images.Elder-Spaces.iwiw.hu-bck if images.Elder-Spaces.iwiw.hu-header-acl
        use_backend thn1.Elder-Spaces.iwiw.hu-bck if thn1.Elder-Spaces.iwiw.hu-header-acl
        use_backend api.Elder-Spaces.iwiw-bck if api.Elder-Spaces.iwiw.hu-header-acl
        use_backend gadget.Elder-Spaces.iwiw-bck if gadget.Elder-Spaces.iwiw.hu-header-acl
        use_backend Elder-Spaces.iwiw.hu-bck     if Elder-Spaces.iwiw.hu-header-acl url-elder-go-to-java-acl
        use_backend php.Elder-Spaces.iwiw.hu-bck if Elder-Spaces.iwiw.hu-header-acl url-elder-go-to-nginx-acl

backend Elder-Spaces.iwiw.hu-bck
```

```
        mode http
        timeout server 60s
        option httpclose
        option nolinger
        option http-server-close
        option http-pretend-keepalive
        option forceclose
        rspirep ^Server:\ (.*) Server:\ iWiW-Elder-Spaces
        option redispatch
        server  app1.Elder-Spaces  app1.Elder-Spaces.iwiw:80 maxconn 200 weight 50 check inter 10s fall 3 rise 1
        server  app2.Elder-Spaces  app2.Elder-Spaces.iwiw:80 maxconn 200 weight 50 check inter 10s fall 3 rise 1
backend php.Elder-Spaces.iwiw.hu-bck
        mode http
        timeout server 60s
        option httpclose
        option nolinger
        balance leastconn
        option redispatch
        option httpchk  GET /favicon.ico HTTP/1.1\r\nHost:\ Elder-Spaces.iwiw.hu
        server app1.origoapps.iwiw app1.origoapps.iwiw:94 maxconn 900 check inter 10000 weight 50 check inter 10s fall 3 rise 1
        server app2.origoapps.iwiw app2.origoapps.iwiw:94 maxconn 900 check inter 10000 weight 50 check inter 10s fall 3 rise 1
backend static.Elder-Spaces.iwiw.hu-bck
        mode http
        timeout server 60s
        option tcp-smart-connect
        option httpclose
        option nolinger
        rspirep ^Server:\ (.*) Server:\ iWiW-Elder-Spaces-static
        balance leastconn
        server  app1.Elder-Spaces  app1.Elder-Spaces.iwiw:8080 maxconn 200 weight 50 check inter 10s fall 3 rise 1
        server  app2.Elder-Spaces  app2.Elder-Spaces.iwiw:8080 maxconn 200 weight 50 check inter 10s fall 3 rise 1
backend gadget.Elder-Spaces.iwiw-bck
        mode http
        timeout server 60s
        option tcp-smart-connect
        option httpclose
        option nolinger
        rspirep ^Server:\ (.*) Server:\ iWiW-Elder-Spaces-gadget
        balance leastconn
        server  gadget.iwiw  gadget.iwiw:8080 maxconn 200 weight 50 check inter 10s fall 3 rise 1
backend images.Elder-Spaces.iwiw.hu-bck
        mode http
        timeout server 60s
        option tcp-smart-connect
        option httpclose
        option nolinger
        rspirep ^Server:\ (.*) Server:\ iWiW-Elder-Spaces-images
        balance leastconn
        server  img1.approval  img1.approval.iwiw.hu:8000 maxconn 200 weight 50 check inter 10s fall 3 rise 1
backend thn1.Elder-Spaces.iwiw.hu-bck
        mode http
        timeout server 60s
        option tcp-smart-connect
        option httpclose
        option nolinger
        rspirep ^Server:\ (.*) Server:\ iWiW-Elder-Spaces-thn
        balance leastconn
        server  thn1.approval  thn1.approval.iwiw:80 maxconn 200 weight 50 check inter 10s fall 3 rise 1
backend api.Elder-Spaces.iwiw-bck
        mode http
        timeout server 60s
        option tcp-smart-connect
        option httpclose
        option nolinger
        balance leastconn
        server  chat1.Elder-Spaces.iwiw  chat1.Elder-Spaces.iwiw:80 maxconn 200 weight 50 check inter 10s fall 3 rise 1 slowstart 60s
        server  chat2.Elder-Spaces.iwiw  chat2.Elder-Spaces.iwiw:80 maxconn 200 weight 50 check inter 10s fall 3 rise 1 slowstart 60s
```

Our firewall is part of the iWiW infrastructure so we do not cover it more detailed (it’s configuration, sysctl variables, etc.).

3. Maintenance

As Elder-Spaces runs on CSCM infrastructure you can use the default CSCM commands to manage your system. The most important commands:

- v-syspatch {attach|config|start|stop|detach|deploy} [$patch]
  With v-syspatch you can start or stop a patch. Also if the patch supports you can deploy a new application version with it. If you don’t provide patch name v-syspatch runs through all patches configured on your system (it knows by the configured groups).
- v-sysconfig generate $file
  With v-sysconfig you can generate configuration files from the templates in the CSCM repository.