



## Applicativo SBNWeb

### **Predisposizione di un server LINUX all'installazione dell'applicativo SBNWEB**

*Compilato : Alessandro Fusco*  
*Data : 24 novembre 2011*  
*Versione : 1.5*  
*Rivisto :*  
*Data :*  
*Distribuito a :*



## Applicativo SBNWeb

### **Predisposizione di un server LINUX all'installazione dell'applicativo SBNWEB**

*Versione* : 1.5  
*Data* : 24 novembre 2011  
*Distribuito a* : ICCU

## INDICE

1. PREMESSA.....	1
2. SISTEMA LINUX – AMBIENTE E CONFIGURAZIONE DI BASE.....	2
2.1. Configurazione del Sistema Operativo.....	2
2.2. Preparazione ambiente di base – Nomenclatura e dimensionamento delle strutture di lavoro .....	3
2.2.1. <i>Directory</i> .....	3
2.2.2. <i>Creazione degli utenti</i> .....	4
2.2.3. <i>Proprietari e permessi</i> .....	4
3. INSTALLAZIONE E CONFIGURAZIONE DEL SW DI BASE.....	6
3.1. Application Server JBOSS .....	6
3.1.1. <i>JBOSS</i> .....	6
3.1.2. <i>JDK (ambiente Java)</i> .....	7
3.2. Apache .....	8
3.3. RDBMS PostgreSQL .....	8
3.3.1. <i>Installazione di PostgreSQL</i> .....	8
3.3.2. <i>Definizione del database cluster</i> .....	9
3.3.3. <i>Configurazione dei file per l'accesso al database server</i> .....	9
3.3.4. <i>Avvio del database server</i> .....	10

## 1. PREMESSA

Questo documento descrive in dettaglio tutte le operazioni che occorre eseguire per la configurazione di base di un server Linux ai fini dell'installazione dell'applicazione SBNWeb.

La descrizione prevede quindi una prima parte relativa alla configurazione del sistema operativo e dell'ambiente di base del server (definizione dei file-system, spazio disco ecc.), e una seconda parte che riguarda più propriamente l'installazione del software di base; per l'installazione delle componenti applicative e la loro configurazione si rimanda al documento specifico.

L'ambiente software necessario sulla macchina Linux per ospitare l'applicazione in oggetto è costituito da:

- Sistema Operativo Linux RedHat 5.3 Enterprise Edition con LVM (in alternativa Scientific Linux SL 5.X)
- Compilatore GCC (solitamente all'interno del pacchetto di installazione del S.O.)
- Application server JBOSS versione 4.2.3 GA con JDK 1.5.X
- Server HTTP Apache Versione 2.2
- RDBMS PostgreSQL Versione 8.3.5

Il criterio adottato per la disposizione degli oggetti che costituiscono l'ambiente software é il seguente: tutti i prodotti di ambiente (motore del database, http server ecc.) vanno installati nei loro path di default (tipicamente /usr/local), mentre gli oggetti che realizzano l'applicativo e/o ne supportano la gestione, vanno installati sotto specifiche directory la cui creazione è demandata alla fase di predisposizione del server.

- l'Application server JBOSS, l'ambiente java e l'RDBMS PostgreSQL installati in **/usr/local**
- i dati e le relative strutture creati in **/db**
- l'ambiente operativo per l'export UNIMARC posizionato in **/export**
- l'ambiente di backup posizionato in **/backup**

## 2. SISTEMA LINUX – AMBIENTE E CONFIGURAZIONE DI BASE

L'applicativo SBNWEB prevede che il server sia equipaggiato con il **Sistema Operativo Linux RedHat 5.X Enterprise Edition**. La licenza di registrazione, pur non essendo di per sé necessaria, è però consigliabile in quanto permette di effettuare in maniera veloce l'eventuale installazione di pacchetti software aggiuntivi direttamente da internet. In alternativa si suggerisce la distribuzione **Scientific Linux SL 5.X** la quale, pur avendo le stesse caratteristiche della RedHat, non necessita di licenza per l'aggiornamento via internet di eventuali pacchetti software aggiuntivi.

Vediamo in dettaglio i principali prerequisiti per una corretta ed efficiente configurazione di base del sistema.

### 2.1. Configurazione del Sistema Operativo

Il S. O. deve essere installato con le seguenti caratteristiche:

- può essere installato a 64 bit se le caratteristiche della macchina lo consentono
- deve essere installato in lingua italiana
- deve essere effettuata una installazione completa di tutti i pacchetti software, in particolare LVM (Logical Volume Manager) per consentire una migliore e più efficace gestione dello spazio disco anche successivamente all'installazione, e il compilatore C nella versione GCC
- deve essere installato il manuale in linea (man) completo
- si suggerisce di valorizzare l'hostname, meglio se con un nome legato alla funzionalità del server (ad esempio Sbnweb-codpolo o qualcosa di simile).

È importante verificare con il comando

```
rpm -qa|grep glibc
```

che al termine dell'installazione del compilatore GCC sul server risultino presenti le seguenti librerie, necessarie per il corretto funzionamento delle procedure sviluppate per l'export UNIMARC:

Librerie C 'glibc'	glibc-headers-2.5-x glibc-2.5-x glibc-devel-2.5-x glibc-common-2.5-x
--------------------	---

Per quanto riguarda la definizione dello spazio disco, tenendo presente la dotazione standard di un server Linux (di solito due dischi di dimensioni ridotte per il Sistema Operativo, e due dischi di grandi dimensioni per l'applicativo e i dati), lo spazio disco deve essere così definito:

- la partizione di sistema, in RAID1, deve avere una dimensione di almeno 50 GByte;
- la partizione dedicata ai dati, in RAID1, deve avere una dimensione di almeno 100 GByte.

Nella partizione di sistema devono essere definiti i seguenti **file-system** con i relativi dimensionamenti:

/	1 Gbyte
/usr	5 Gbyte
/usr/local	8 Gbyte
/home	5 Gbyte
/var	1 Gbyte
/tmp	1 Gbyte
/opt	1 Gbyte

Nella partizione dei dati devono essere definiti i seguenti file system:

/db	40 Gbyte
/export	20 Gbyte
/backup	30 Gbyte

**Nota:** i dimensionamenti sopra riportati vanno considerati come minimali in fase di partenza, tenendo conto che lo strumento LVM consente di modificarli dinamicamente e in qualsiasi momento a seconda delle esigenze riscontrate.

## **2.2. Preparazione ambiente di base – Nomenclatura e dimensionamento delle strutture di lavoro**

In questo paragrafo affrontiamo la preparazione dell'ambiente di base, intendendo in particolare la definizione delle directory e sotto-directory, con relativi permessi e proprietari, necessarie ad accogliere i prodotti software e l'applicativo SBNWEB.

### **2.2.1. Directory**

Sotto **/db** creare le seguenti directory (per i proprietari e i permessi vedi par. 2.2.3)

**pgsql8.3.5** → contiene la directory con le strutture e i dati del databases;

**pgsql8.3.5/data** → contiene le strutture e i dati del databases;

**migrazione** → contiene l'ambiente di lavoro per l'eventuale migrazione del polo;

Sotto **/export** creare le seguenti sotto-directory (per i proprietari e i permessi vedi par. 2.2.3)

**Trasf** → contiene l'ambiente per la distribuzione degli aggiornamenti sw dell'applicativo

**Trasf/dep**

**Trasf/Storico**

**Trasf/logs**

Sotto **/backup** creare le seguenti directory (per i proprietari e i permessi vedi par. 2.2.3)

**DUMP\_DB** → contiene i file di backupi del databases;

**DUMPSYS** → contiene i file di backup dell'ambiente, cioè delle directory principali;

**logs** → destinata ad accogliere i file di log dell'esecuzione delle procedure di backup;

Sotto **/home** creare la seguente directory (per i proprietari e i permessi vedi par. 2.2.3)

**SCRIPTS** → contiene gli scripts di backup ed altri per la gestione del sistema;

### 2.2.2. Creazione degli utenti

Occorre creare i seguenti utenti:

**jboss** utente preposto all'utilizzo dell'applicazione;

**export** utente preposto esclusivamente alle operazioni di export UNIMARC (qualora le si voglia eseguire senza attivazione da interfaccia applicativa);

**postgres** utente preposto alla gestione dell'ambiente database.

Gli utenti sopra descritti vanno creati con le seguenti caratteristiche:

UTENTE	GRUPPO	HOME-DIR	PASSWORD ALLA CREAZIONE
jboss	jboss, export	/home/jboss	Jb9ss
export	export, jboss	/home/export	sbnwebExport
postgres	postgres	/db	postgresadm

### 2.2.3. Proprietari e permessi

Una volta create le directory e aver definito gli utenti, possiamo correttamente assegnare i proprietari e i permessi.

/db (drwxrwxrwx postgres root)

drwxr-xr-x postgres postgres **pgsql8.3.5**

drwx----- postgres postgres **pgsql8.3.5/data**

drwxrwxr-x jboss jboss **migrazione**

/export (drwxr-xr-x export jboss)

drwxrwxr-x export export Trsf

drwxrwxr-x export export Trsf/dep

drwxrwxr-x export export Trsf/Storico

drwxrwxr-x    export export    Trsf/logs

/backup        (drwxr-xr-x root root)

drwxr-xr-x     postgres root **DUMP\_DB**

drwxr-xr-x     root root **DUMPSYS**

drwxrwxrwx    root root **logs**

/home/SCRIPTS (drwxr-xr-x root root)



### 3. INSTALLAZIONE E CONFIGURAZIONE DEL SW DI BASE.

Esamineremo in questo capitolo le modalità di installazione e configurazione dei prodotti software di base che costituiscono l'ambiente operativo di SBNWEB.

#### 3.1. Application Server JBOSS

Trattiamo in questo paragrafo l'installazione del prodotto JBOSS v. 4.2.3. Qualora si decida l'installazione del prodotto nella versione 5.1, si faccia riferimento al documento "Migrazione dell'applicativo SbnWeb in ambiente Jboss 5.1"

L'installazione dell'Application Server JBOSS consta di due passi distinti che riguardano il prodotto JBOSS stesso e l'ambiente Java. Vediamoli in dettaglio.

##### 3.1.1. JBOSS

I passi da eseguire sono i seguenti (vanno eseguiti come utente root):

1. Aggiornare preliminarmente il file `/etc/hosts` aggiungendo alla riga 127.0.0.1 l'hostname del server
2. Scaricare da Internet il pacchetto JBOSS-4.2.3.GA.zip e posizionarlo nella directory `/usr/local`
3. Portarsi nella directory `/usr/local` e decomprimere il pacchetto.

L'ultimo comando crea una directory `jboss-4.2.3.GA` sotto la quale troverà locazione tutto l'ambiente riferibile all'Application Server.

Completata l'installazione del pacchetto **eliminare** dalla cartella `/usr/local/jboss-4.2.3.GA/server/default/lib` le seguenti librerie:

- commons-codec.jar
- commons-collections.jar
- commons-httpclient.jar
- hibernate3.jar
- quartz.jar

e sostituirle con le versioni più avanzate, reperibili su internet, le cui ulteriori funzionalità sono necessarie all'applicativo:

- commons-codec-1.3.jar
- commons-collections-3.2.jar
- commons-httpclient-3.1.jar
- hibernate-3.2.6.jar
- quartz-all-1.6.5.jar

aggiungere quindi le librerie:

- postgresql-8.3-603.jdbc2ee.jar

L'attività può considerarsi conclusa.

L'ultimo passo riguarda la personalizzazione di due file che contengono le informazioni per il corretto puntamento al DB; tale personalizzazione va però effettuata a conclusione delle attività

di creazione del DB e a valle dell'installazione dell'applicativo; per questa ragione se ne rimanda la trattazione al capitolo 2.2 Configurazione dell'applicativo del documento "Installazione dell'applicativo SBNWEB in ambiente Linux".

Per le analoghe attività relative alla versione 5.1 del prodotto, si rimanda la cap. 2 del citto documento "Migrazione dell'applicativo SbnWeb in ambiente Jboss 5.1"

### **3.1.2. JDK (ambiente Java)**

Per l'installazione di questa componente, seguire i seguenti passi come utente root

Da Internet scaricare il prodotto JDK 1.5.X per piattaforme LINUX e in formato *.bin* autoestraente; quindi posizionarlo nella directory */usr/local*.

Posizionarsi nella */usr/local* ed eseguire il file *.bin* (*./<nome-del-file>.bin*)

Verrà eseguita l'installazione dell'ambiente java in una directory creata appositamente con il nome e la versione del prodotto (ad esempio: *jdk1.5.0\_12*)

Per comodità di lavoro, creare sotto */usr/local* il link *jdk -> jdk1.5.0\_12/* con il comando

```
ln -s jdk1.5.0_12/ jdk
```

### 3.2. Apache

Solitamente il prodotto APACHE 2.2 viene installato insieme al Sistema Operativo nella directory standard */etc/httpd*.

Pertanto in questo paragrafo esamineremo in dettaglio le personalizzazioni da effettuare nel file di configurazione **proxy\_ajp.conf**.

Il file in questione si trova nella cartella */etc/httpd/conf.d* e va modificato come utente root:

Inserire in fondo al file **proxy\_ajp.conf** le seguenti righe

```
ProxyPass /sbn ajp://localhost:8009/sbn/  
ProxyPass /SbnMarcWeb ajp://localhost:8009/SbnMarcWeb/  
ProxyPass /servizi ajp://localhost:8009/servizi/  
ProxyPass /web-console ajp://localhost:8009/web-console  
ProxyPass /jmx-console ajp://localhost:8009/jmx-console
```

Effettuate le personalizzazioni sopra descritte, bisogna restartare il servizio ed inserire in maniera permanente lo stop/start automatico alla ripartenza del sistema. I passi sono (sempre come root):

fermare il servizio con il comando **/etc /init.d /httpd stop**

far ripartire il servizio con il comando **/etc/init.d/httpd start**

eseguire come root il comando **chkconfig httpd on**

### 3.3. RDBMS PostgreSQL

L'installazione del prodotto richiede che alcuni passi siano fatti come utente root (installazione), mentre la configurazione deve essere fatta come utente *postgres*, per la creazione dell'utenza *postgres* si rimanda al par. 2.2.2.

Naturalmente in questa sede viene esaminato solo l'aspetto di installazione e configurazione del prodotto, rimandando ad altro documento tutto ciò che concerne la creazione del database, delle relative strutture dati e il loro popolamento.

#### 3.3.1. Installazione di PostgreSQL

I passi, da effettuare come utente root, sono i seguenti:

Scaricare da internet il pacchetto *postgresql-8.3.5.tar.gz* e posizionarlo sotto una directory di lavoro; in ambiente di collaudo è stata scelta ***/usr/local/src***

Spostarsi nella directory di lavoro e decomprimere il pacchetto con il comando

```
tar -xzf postgresql-8.3.5.tar.gz
```

Verrà quindi creata una cartella, chiamata di default come la versione di PostgreSQL che stiamo utilizzando, a cui dovremo imporre come proprietario l'utente *Postgres* tramite il comando

**chown -R postgres:postgres postgresql-8.3.5**

Ora i sorgenti dovranno essere configurati per la compilazione tramite la seguente serie di istruzioni:

```
./configure --prefix=/usr/local/pgsql8.3.5 (directory dove verrà installato il motore)
make (compilazione, può richiedere un certo tempo)
make check (per controllare l'esito della compilazione)
make install (installazione)
```

Infine, settare correttamente la variabile PATH nel profile dell'utente postgres (che, lo ricordiamo, si trova nella home-directory /db): nel file .bash\_profile aggiungere in fondo alla definizione della variabile PATH la directory /usr/local/pgsql8.3.5/bin (PATH=\$PATH:/usr/local/pgsql8.3.5/bin).

**3.3.2. Definizione del database cluster**

Terminata la fase di installazione, passiamo alla creazione del database cluster.

Usando l'utente root posizionarsi nella cartella /db/pgsql8.3.5 e creare la directory **data** quindi assegnarle postgres:postgres come proprietario

```
cd /db/pgsql8.3.5
mkdir data
chown -R postgres:postgres data
```

Passare a utente postgres con il comando `su - postgres`

Creare in PostgreSQL un nuovo database cluster con il comando

```
/usr/local/pgsql8.3.5/bin/initdb --locale=C --lc-messages=it_IT.UTF-8 -E UNICODE -D /db/pgsql8.3.5/data
```

**3.3.3. Configurazione dei file per l'accesso al database server**

L'ultimo passo riguarda la personalizzazione di due importanti file di configurazione per consentire il corretto accesso al database server.

I file si trovano sotto la directory /db/pgsql8.3.5/data e sono:

```
postgresql.conf
pg_hba.conf
```

Il file **postgresql.conf** va modificato per consentire che il database server si ponga in ascolto per qualsiasi indirizzo (il default è 'localhost').

Nella sezione "CONNECTIONS AND AUTHENTICATION", togliere il cancelletto di commento alla riga

```
#listen_addresses = 'localhost'
```

e sostituire al termine 'localhost' il termine '\*' (mantenendo gli apici).

Sempre nel file ***postgresql.conf***, nella sezione "WRITE AHEAD LOG", togliere il cancelletto di commento alla riga

```
#checkpoint_segments = 3
```

e sostituire il valore **10** al valore 3.

Infine, sempre nello stesso file, nella sezione "CLIENT CONNECTION DEFAULTS", alla variabile `default_text_search_config` modificare il valore di default *pg\_catalog.english* con il valore

***pg\_catalog.italian***

Nel file ***pg\_hba.conf*** vanno inserite le informazioni necessarie ad abilitare uno o più indirizzi IP alla connessione col database server.

Gli indirizzi da abilitare vanno indicati in CIDR notation nell'apposita sezione: "# IPv4 local connections:" o "# IPv6 local connections:" dipendentemente dalla tipologia di rete in cui è inserito il server.

Per ogni sottorete o indirizzo puntuale da cui provengono le richieste di connessione al database server va inserita una riga del tipo:

```
host all all <indirizzo IP>/32 password
```

Si riportano, a solo titolo di esempio, due righe da inserire in "# IPv4 local connections:" la prima per abilitare uno specifico indirizzo (193.206.221.2), la seconda per abilitare una sottorete (da 192.168.10.0 a 192.168.10.255), in ambedue i casi l'accesso sarà condizionato alla verifica della password.

```
host all all 193.206.221.2/32 password
```

```
host all all 192.168.10.0/24 password
```

### **3.3.4. Avvio del database server**

A questo punto le operazioni sono terminate e si può far partire il database server (sempre come utente *postgres*).

Per avviare il database server usare il comando:

```
/usr/local/pgsql8.3.5/bin/pg_ctl -D /db/pgsql8.3.5/data -l logfile start
```

Per fermare il database server usare il comando:

```
/usr/local/pgsql8.3.5/bin/pg_ctl -D /db/pgsql8.3.5/data stop
```

Infine, una volta avviato postgres, occorre modificare la password dell'utente DB postgres con i seguenti comandi (sempre come utente *postgres*)

```
psql
```

```
alter user postgres with password 'postgresadm';
```

```
\q
```

Nella fase di installazione dell'applicativo verranno forniti degli appositi files per l'attivazione automatica del data server all'avvio della macchina.

La loro descrizione e le istruzioni per il loro posizionamento e personalizzazione verranno forniti nel documento di installazione delle componenti applicative.