



Installazione Oracle

Versioni

- Oracle 10-11 ufficialmente supportate
- Postgress 9.1/2/3 supportate ma Bonita consiglia la 9.3 per la versione 6.4
- Bonita 6.4
- Java Oracle 1.7 per Bonita 6.4
- Java Oracle 1.8 per tomcat 8.5
- Tomcat 8.5
- eXtraWay EE

Preparazione ambiente

Disabilitare selinux e firewall.

Aggiungere gruppi:

```
groupadd -g 501 oinstall
groupadd -g 502 dba
groupadd -g 503 oper
```

Aggiungere utente:

```
useradd -u 502 -g oinstall -G dba,oper oracle
```

Impostare password:

```
passwd oracle
```

vim /etc/sysctl.conf ed aggiungere:

```
fs.aio-max-nr = 1048576
fs.file-max = 6815744
kernel.shmall = 2097152
kernel.shmmax = 4294967295
kernel.shmmni = 4096
kernel.sem = 250 32000 100 128
net.ipv4.ip_local_port_range = 9000 65500
net.core.rmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_default = 262144
net.core.wmem_max = 1048576
```

Eeguire:

```
sysctl -p
```

vim /etc/security/limits.conf ed aggiungere:

```
oracle soft nproc 131072
oracle hard nproc 131072
oracle soft nofile 131072
oracle hard nofile 131072
oracle soft core unlimited
oracle hard core unlimited
oracle soft memlock 50000000
oracle hard memlock 50000000
```

vim /etc/pam.d/login ed aggiungere:

```
session required pam_limits.so
```

Vim /etc/hosts ed aggiungere ip nome macchina completo e nome macchina. Esempio:

```
10.17.61.190 auac.bo.priv auac
```

Modificare .bash_profile per utente oracle nella sua home /home/oracle



```
# Oracle Settings
export TMP=/tmp

export ORACLE_HOSTNAME=auac.bo.priv
export ORACLE_UNQNAME=orcl
export ORACLE_BASE=/ora01/app/oracle
export ORACLE_HOME=$ORACLE_BASE/product/11.2.0/db_1
export ORACLE_SID=orcl

PATH=/usr/sbin:$PATH:$ORACLE_HOME/bin

export LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib;
export CLASSPATH=$ORACLE_HOME/jlib:$ORACLE_HOME/rdbms/jlib;

alias cdob='cd $ORACLE_BASE'
alias cdoh='cd $ORACLE_HOME'
alias tns='cd $ORACLE_HOME/network/admin'
alias envo='env | grep ORACLE'

umask 022
```

Installare :

```
yum install binutils -y
yum install compat-libstdc++-33 -y
yum install compat-libstdc++-33.i686 -y
yum install gcc -y
yum install gcc-c++ -y
yum install glibc -y
yum install glibc.i686 -y
yum install glibc-devel -y
yum install glibc-devel.i686 -y
yum install ksh -y
yum install libgcc -y
yum install libgcc.i686 -y
yum install libstdc++ -y
yum install libstdc++.i686 -y
yum install libstdc++-devel -y
yum install libstdc++-devel.i686 -y
yum install libaio -y
yum install libaio.i686 -y
yum install libaio-devel -y
yum install libaio-devel.i686 -y
yum install libXext -y
yum install libXext.i686 -y
yum install libXtst -y
yum install libXtst.i686 -y
yum install libX11 -y
yum install libX11.i686 -y
yum install libXau -y
yum install libXau.i686 -y
yum install libxcb -y
yum install libxcb.i686 -y
yum install libXi -y
yum install libXi.i686 -y
yum install make -y
yum install sysstat -y
yum install unixODBC -y
yum install unixODBC-devel -y
yum install zlib-devel -y
yum install elfutils-libelf-devel -y
yum install xorg-x11-server-Xorg xorg-x11-xauth xorg-x11-apps xorg-x11-server-utils xdpinfo -y
```

Creare struttura cartelle:



```
mkdir -p /ora01/app
chown oracle:oinstall /ora01/app
chmod 775 /ora01/app
```

Creare ORACLE_BASE directory per oracle:

```
mkdir -p /ora01/app/oracle
chown oracle:oinstall /ora01/app/oracle
chmod 775 /ora01/app/oracle
```

Creare ORACLE_HOME directory per oracle:

```
mkdir -p /ora01/app/oracle/product/11.2.0/db_1
chown oracle:oinstall -R /ora01/app/oracle
```

In Centos Linux 7 /tmp per convertire il tmp in storage, eseguire il comando e fare riavvio.

```
systemctl mask tmp.mount
```

Setup di Oracle tramite console

Entrare come utente oracle:

```
su - oracle
```

Scaricare il file dalla cartella in rete RESOURCES/ORACLE/11g sulla home /home/oracle e fare unzip i due file.

Modificare vim /etc/ssh/sshd_config in:

```
X11Forwarding yes
```

```
systemctl restart sshd
```

Come root eseguire:

```
xhost +<nome macchina>
```

Come oracle eseguire:

```
DISPLAY=<machine-name>:0.0; export DISPLAY
```

Installare Xming sulla tua macchina Windows. Avvia Xming Launcher. Scegli tutto default e disabilita solo ACl. Quando fai il login con putty prima abilita x11 forwarding sulle opzioni del SSH.

Entrare come utente oracle e far partire il setup ./runInstaller nella cartella database che abbiamo estratto prima.

Disabilita security update nella prima schermata. Ignora il messaggio della mail. Scegliere versione Enterprise. E non Creare nuovo DB se non si sa il nome db. Quando fa il check, potrebbero essere dei problemi con le versioni pacchetti ed un campo kernel che non lo riconosce anche se è giusto. Ignorare tutto e proseguire.

Quando appare l'errore "install":

```
vim /ora01/app/oracle/product/11.2.0/db_1/ctx/lib/ins_ctx.mk
```

Da:

```
ctxhx: $(CTXHXOBJ)
$(LINK_CTXHX) $(CTXHXOBJ) $(INSO_LINK)
```

a:

```
ctxhx: $(CTXHXOBJ)
-static $(LINK_CTXHX) $(CTXHXOBJ) $(INSO_LINK) /usr/lib/gcc/x86_64-redhat-linux/4.8.2/libstdc++.a
```

Poi prova "Retry"

Quando appare l'errore "agent nmhs" :

Editare \$ORACLE_HOME/sysman/lib/ins_emagent.mk, cerca:

```
$(MK_EMAGENT_NMECTL)
```

e sostituire con:



```
$(MK_EMAGENT_NMECTL) -lnnz11
```

Riprova “Retry”

Alla schermata degli script, eseguirli come root dalla console quelli indicati.

Per come sysdba. Fare login come oracle e poi eseguire :

sqlplus / as sysdba

Creare, cancellare e modificare DB tramite console

Creare DB, CDB e PDB tramite la console con la connessione **X11 forwarding** come sopra indicato. Come utente oracle eseguire:

```
dbca
```

e seguire i passi come preferenze. Nel caso di PDB scegliere anche l'opzione del container e mettere il nome che si preferisce.

Usare il comando **dbca** anche per cancellare o modificare i DB.

Per aprire il **pdb** connesso con utente sys si lancia il comando

```
ALTER DATABASE OPEN;
```

per controllare lo stato

```
select status from v$instance;
```

Post Installation

Editare “/etc/oratab” file setting ed aggiungere il flag come nell'esempio, 'Y'. La riga esisterà quando sarà creato un DB.

```
auac:/ora01/app/oracle/product/11.2.0/db_1:Y
orcl:/ora01/app/oracle/product/11.2.0/db_1:Y
```

Editare “dbstart” & “dbshut” file, in \$ORACLE_HOME/bin per non avere il messaggio di ORACLE_HOME_LISTNER. Cambiare da:

```
ORACLE_HOME_LISTNER=$1
```

a:

```
ORACLE_HOME_LISTNER=$ORACLE_HOME
```

Abilitare web console

Per la console EMC dopo aver acceso il servizio oracle eseguire:

```
emca -config dbcontrol db -repos create
```

Seguire i passi.

Aggiungere oracle come servizio in systemd

Creare cartella lock:

```
mkdir /home/oracle/lock
```

Creare file vim /etc/init.d/oracle con il contenuto:

```
#!/bin/bash
#
# /etc/init.d/oracledb
#
# Run-level Startup script for the Oracle Instance, Listener, and Web Interface

export ORACLE_HOME=/ora01/app/oracle/product/11.2.0/db_1
export PATH=$PATH:$ORACLE_HOME/bin
export ORACLE_UNQNAME=orcl
export ORACLE_HOME_LISTNER="$ORACLE_HOME"

ORA_OWNER="oracle"

if [ x"${whoami}" != x"$ORA_OWNER" ]
```



```
then
    echo "Switching to user $ORA_OWNER"
    su "$ORA_OWNER" -c "$0 $@"
    exit 0
fi

# if the executables do not exist -- display error

if [ ! -f $ORACLE_HOME/bin/dbstart -o ! -d $ORACLE_HOME ]
then
    echo "Oracle startup: cannot start"
    exit 1
fi

case "$1" in
    start)
        # Oracle listener and instance startup
        echo -n "Starting Oracle: "
        $ORACLE_HOME/bin/lsnrctl start
        $ORACLE_HOME/bin/dbstart
        touch /home/oracle/lock/oracle

        $ORACLE_HOME/bin/emctl start dbconsole
        echo "OK"
        ;;
    stop)
        # Oracle listener and instance shutdown
        echo -n "Shutdown Oracle: "
        $ORACLE_HOME/bin/lsnrctl stop
        $ORACLE_HOME/bin/dbshut
        rm -f /home/oracle/lock/oracle

        $ORACLE_HOME/bin/emctl stop dbconsole
        echo "OK"
        ;;
    reload|restart)
        $0 stop
        $0 start
        ;;
    *)
        echo "Usage: `basename $0` start|stop|restart|reload"
        exit 1
esac

exit 0
```

Renderlo eseguibile con:

```
chmod +x /etc/init.d/oracle
```

Creare file /etc/systemd/system/oracle.service con contenuto:

```
[Unit]
Description=Servizio di oracle
After=network.target remote-fs.target nss-lookup.target

[Service]
Type=forking
ExecStart=/etc/init.d/oracle start
ExecStop=/etc/init.d/oracle stop

SuccessExitStatus=0 143
RestartSec=10
Restart=always
```



```
User=oracle
Group=oinstall

[Install]
WantedBy=multi-user.target
```

Eeguire i seguenti comandi:

```
systemctl daemon-reload
systemctl enable oracle.service
systemctl start oracle.service
```

Export/import Oracle DB

Export DB

Creare cartella bolzano in \$ORACLE_HOME/admin/orcl/dpdump/

Enter nella console di Oracle:

```
sqlplus / as sysdba
```

```
CREATE DIRECTORY bolzano AS '/opt/oracle/Oracle11gee/admin/orcl/dpdump/bolzano';
```

```
GRANT READ, WRITE ON DIRECTORY bolzano TO system;
```

Exit console. Eeguire:

Oracle inferiori a 19g:

```
expdp AUAC_USR_EXT/auac_pwd_test@ORCL SCHEMAS=AUAC_USR_EXT DIRECTORY=bolzano
```

Oracle 19g o superiori:

```
expdp USERID='auac_usr_ext/auac_pwd_19@localhost:1521/auacpdb\' DIRECTORY=marche
DUMPFILE=auacdumpall_for_marche.dmp LOGFILE=auacdumpall_for_marche.log SCHEMAS=AUAC_USR_EXT
CONTENT=ALL Copiare il file .dmp nel nuovo server nella cartella dpdump/bolzano =====Import DB=====
Creare cartella bolzano in $ORACLE_HOME/admin/orcl/dpdump/ Entrare nella console del Oracle:
sqlplus / as sysdba <code xml> CREATE DIRECTORY bolzano AS
'/ora01/app/oracle/admin/orcl/dpdump/bolzano'; GRANT READ, WRITE ON DIRECTORY bolzano TO system;
CREATE BIGFILE TABLESPACE TSD_AUACUSR_1 DATAFILE 'tbs_tsd_auac.dat' SIZE 4G AUTOEXTEND ON; CREATE
BIGFILE TABLESPACE AUAC_USR_EXT DATAFILE 'tbs_usr_auac.dat' SIZE 4G AUTOEXTEND ON; CREATE BIGFILE
TABLESPACE STABLE DATAFILE 'tbs_stable.dat' SIZE 1G AUTOEXTEND ON; CREATE BIGFILE TABLESPACE
TSD_MRAAUACUSR_1 DATAFILE 'tbs_tds_mraauacusr_1.dat' SIZE 1G AUTOEXTEND ON; - oracle 19g CREATE
BIGFILE TABLESPACE AUAC_TBS_I DATAFILE 'AUAC_TBS_I' SIZE 1G AUTOEXTEND ON; CREATE BIGFILE
TABLESPACE AUAC_TBS_L DATAFILE 'AUAC_TBS_L' SIZE 1G AUTOEXTEND ON; CREATE BIGFILE TABLESPACE
AUAC_TBS_T DATAFILE 'AUAC_TBS_T' SIZE 1G AUTOEXTEND ON; - end oracle 19g CREATE USER AUAC_USR_EXT
identified by auac_pwd_test default tablespace TSD_AUACUSR_1; GRANT CREATE SESSION,CONNECT to
AUAC_USR_EXT; GRANT DBA to AUAC_USR_EXT; - oracle 19g ALTER USER AUAC_USR_EXT QUOTA UNLIMITED ON
AUAC_TBS_I; ALTER USER AUAC_USR_EXT QUOTA UNLIMITED ON AUAC_TBS_L; ALTER USER AUAC_USR_EXT QUOTA
UNLIMITED ON AUAC_TBS_T; ALTER USER AUAC_USR_EXT QUOTA UNLIMITED ON USERS; - end oracle 19g </code>
Exit console. Eeguire: Oracle inferiore a 19g: impdp AUAC_USR_EXT SCHEMAS=AUAC_USR_EXT
DIRECTORY=bolzano Oracle 19g o superiori: impdp USERID='/' as sysdba\' DIRECTORY=marche
DUMPFILE=auacdumpall_for_marche.dmp LOGFILE=auacdumpall_for_marche.log SCHEMAS=AUAC_USR_EXT
CONTENT=ALL =====Installazione postgresql===== Eeguire: <code xml> yum install postgresql-server
postgresql-contrib postgresql-setup initdb systemctl enable postgresql </code> Editare il file
/var/lib/pgsql/data/pg_hba.conf (in Ubuntu /etc/postgresql/versione_postgre/main)cambiare tutti i
ident in md5 e aggiungere una con host 0.0.0.0/0 per accettare le connessioni da tutti: (come
nell'esempio qui sotto) <code xml> host all all 127.0.0.1/32 md5 host all all 0.0.0.0/0 md5 </code>
Editare il file /var/lib/pgsql/data/postgresql.conf ed aggiungere o modificare in: <code xml>
listen_addresses = '*' </code> Riavviare il servizio postgres. =====Operazioni su postgres===== In
Linux di default, si crea un utente postgres dopo l'installazione. Puoi cambiare la sua password
eseguendo: <code xml> passwd postgres </code> L'utente postgres è senza password di default. Dopo,
puoi entrare nel prompt di PostgreSQL e cambiare la password per PostgreSQL postgres user eseguendo:
<code xml> su - postgres </code> Se ricevi un errore, puoi impostare una shell valida con il
seguente comando: <code xml> su -shell /bin/bash postgres </code> Dopodiché, eseguire: <code xml>
su - postgres </code> Per cambiare la password, usa il seguente comando: <code xml> psql -d
template1 -c "ALTER USER postgres WITH PASSWORD 'NewPassword';" </code> Puoi entrare nel PostgreSQL
client shell usando: <code xml> psql postgres </code> Per uscire scrivere \q. Per creare un DB
eseguire: <code xml> createdb testDB </code> Puoi creare un ruolo come nell'esempio: <code xml>
createuser samplerole -pwprompt </code> Ti verrà chiesta una password per l'utente. Altrimenti puoi
```



assegnare ownership del nuovo DB ad un specifico utente o ruolo. Si può realizzare eseguendo :
`<code xml> createdb testDB -O samplerole </code>` Nell'esempio qui sopra, specificare il ruolo che si desidera. Ti puoi collegare al nuovo DB eseguendo: `<code xml> psql testDB </code>` Se si vuol entrare con un utente o ruolo specifico, eseguire: `<code xml> psql testDB -U samplerole </code>` In caso che dia errore, provare ad aggiungere l'host: `<code xml> psql testDB -U samplerole -h host </code>` Puoi cancellare un DB usando il comando dropdb: `<code xml> dropdb testDB </code>` ===== Creare o cancellare DB bonita ===== Entrare nel psql console ed eseguire: `<code xml> postgres=#CREATE USER bonita WITH PASSWORD 'bonita'; postgres=#CREATE DATABASE bonita WITH ENCODING 'UTF8' OWNER bonita; </code>` Vedere la lista DB eseguire: \l Cancellare DB. Entrare nel psql console ed eseguire: `<code xml> postgres=#DROP DATABASE bonita;' </code>` ===== Backup per export ed importare DB ===== Per backup eseguire come root o utente postgres: `<code xml> pg_dump -h localhost -U bonita bonita > db_bonita_exp.pgsql </code>` Per restore eseguire: `<code xml> cat db_bonita_exp.pgsql |psql -U bonita -d bonita -h localhost </code>` ===== Installazione Bonita 6.4 ===== Esiste un pacchetto già pronto, solo da estrarre con unzip, qui: `<code>smb:storage.bo.priv/resources/Bonita/BonitaBPMCommunity-6.4.2-Tomcat-7.0.55.zip</code>` Richiede solamente **JDK 7** ===== Creazione utente e db ===== Dopo aver fatto login come utente postgres aggiungiamo l'utente bonita e creiamo i due database necessari al corretto funzionamento; `<code> $psql postgres=# CREATE USER bonita WITH PASSWORD 'bonita'; postgres=# CREATE DATABASE bonita WITH ENCODING 'UTF8' OWNER bonita; postgres=# \q </code>` ===== Driver JDBC ===== Scarica i Driver JDBC per postgres [qui](#) e mettiamoli nella cartella lib/bonita/. ===== Bonita ===== 1) bin/setenv.sh Modificare il vendor da "h2" a "postgres" 2) conf/Catalina/localhost/bonita.xml Configurazione database postgres 3) webapps/bonita/WEB-INF/classes/bonita-connectors.properties (nella versione 6 non esiste) Configurazione 3diws e database oracle 4) conf/bitronix-resources.properties Configurazione database postgres 5) copiare il driver jdbc nella cartella lib/bonita ===== BonitaView ===== 1) webapps/bonitaview/WEB-INF/classes/bonitaView.properties Configurazione bonita ===== File di config per le app ===== ===== AuAc ===== 1) webapps/AuAc/WEB-INF/web.xml Configurazione CAS 2) webapps/AuAc/WEB-INF/spring-servlet.xml Configurazione CAS 3) webapps/AuAc/WEB-INF/classes/auac.properties Configurazione 3diws, SMTP, database oracle, CAS (url logout), bonitaview, bonita ===== SanitaRegioneVeneto ===== 1) webapps/SanitaRegioneVeneto/WEB-INF/web.xml Configurazione CAS 2) webapps/SanitaRegioneVeneto/WEB-INF/classes/auac.properties Configurazione SMTP, database oracle ===== AREA ===== 1) webapps/area/WEB-INF/classes/application-venetoprod.properties Configurazione database oracle e CAS