Getting Started with the Raspberry Pi

[GettingStartedRaspberryPi.htm - uqlpayne 30 Oct 2013]

Contents

- Requirements
- Install PostgreSQL
- Install GlassFish
- Relay Emails Through GMail
- IlabServiceBroker Deployment

Requirements

- Raspberry Pi *Raspbian* Image which contains the *Java Platform (JDK)*
  http://www.raspberrypi.org/downloads

  It is assumed that this image is running on the Raspberry Pi and that the network has been configured for access to the internet.

- GlassFish Server Open Source Edition 4.0 (Linux) [glassfish-4.0.zip (97.0 MB)]
  https://glassfish.java.net/download.html

Install PostgreSQL

- Use *PuTTY* to log into the Raspberri Pi.
- An update must be performed first so that *apt-get* knows that new versions of packages are available.

  ```
  sudo apt-get update
  ```

- Install the newest versions of all packages currently installed on the system.

  ```
  sudo apt-get upgrade
  ```

- Install *PostgreSQL*. At time of writing, this will install version 9.1 of PostgreSQL.

  ```
  sudo apt-get install postgresql
  ```

- Administration of the PostgreSQL databases is best done by using the GUI tool *pgAdmin III* rather than using the command line. Since *pgAdmin III* cannot be run on the Raspberry Pi, remote access is required.

  Modify the PostgreSQL configuration files so that administration of PostgreSQL can be carried out remotely by *pgAdmin III*.

  ```
  cd /etc/postgersql/9.1/main/
  sudo cp postgresql.conf postgresql.conf.orig
  sudo nano postgresql.conf
  ```

  Change the line:

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

  to:

  ```
  listen_addresses = '*'
  ```

  Save the file.

  ```
  sudo cp pg_hba.conf pg_hba.conf.orig
  ```

sudo nano pg_hba.conf

Add the line:

host all all .domain md5

Replace *.domain* with the domain of the network that you are using and save the file.

- Restart the PostgreSQL service.

  sudo service postgresql restart

- Change the PostgreSQL *admin* password.

  sudo -u postgres psql postgres

  At the *postges=#* prompt, enter:
  \password postgres

  At the *Enter new password:* prompt, enter the new password.

  At the *Enter it again:* prompt, enter the new password again.

  At the *postges=#* prompt, enter:
  \q

- Run *pgAdmin III* on another computer on the same network that has it installed.

  Add the PostgreSQL server running on the Raspberry Pi as follows:

  - From the menu, select *File->Add Server...*
  - For *Name:*, enter a name for the server, e.g. *ilab-pi*
  - For *Host:*, enter the full hostname of the Raspberry Pi
  - For *Username:*, enter **postgres**
  - For *Password:*, enter a password
  - Uncheck *Store password*
  - Click *OK*

Install GlassFish

- Use *WinSCP* to transfer *glassfish-4.0.zip* to */home/pi/* on the Raspberri Pi.
- Use *PuTTY* to log into the Raspberri Pi.
- Unzip *glassfish-4.0.zip* to */opt/* on the Raspberri Pi.

  sudo unzip glassfish-4.0.zip -d /opt/

- Start GlassFish running on the Raspberry Pi:

  cd /opt/glassfish4/bin/
  sudo ./asadmin start-domain domain1

- Change the GlassFish *admin* password.

  sudo ./asadmin change-admin-password

  At the *Enter admin user name [default: admin]*> prompt, hit *Enter*.

  At the *Enter the admin password*> prompt, hit *Enter*.

  At the *Enter the new admin password*> prompt, enter the new password.

  At the *Enter the new admin password again*> prompt, enter the new password again.

  The following message should display:

```
Command change-admin-password executed successfully.
```

- Enable remote access to the GlassFish administration console.

  ```
  sudo ./asadmin enable-secure-admin
  ```

  At the *Enter admin user name*> prompt, enter:
  admin

  At the *Enter admin password for user "admin"*> prompt, enter admin's password.

  The following message should display:
  ```
  You must restart all running servers for the change in secure admin to take effect.
  Command enable-secure-admin executed successfully.
  ```

- Stop GlassFish and restart:

  ```
  sudo ./asadmin stop-domain domain1
  sudo ./asadmin start-domain domain1
  ```

  View the GlassFish server log:

  ```
  more /opt/glassfish4/glassfish/domains/domain1/logs/server.log
  ```

- Test GlassFish:

  Use *WinSCP* to transfer pitest_index.jsp to */home/pi/* on the Raspberri Pi.

  ```
  cd /opt/glassfish4/glassfish/domains/domain1/autodeploy
  sudo mkdir test
  sudo cp /home/pi/pitest_index.jsp test/index.jsp
  ```

  Open a web browser on another computer on the same network as the Raspberry Pi and enter the url:

  ```
  http://host.domain:8080/test/
  ```

  Replace *host.domain* with the full hostname of the Raspberry Pi that you are using.

  The web page should display something like:

  ```
  PiTest
  App Server : GlassFish Server Open Source Edition 4.0
  JVM : 1.7.0_40 - Oracle Corporation
  Java home : /usr/lib/jvm/jdk-7-oracle-armhf/jre
  OS : Linux - 3.6.11+
  Architecture : arm
  ```

Relay Emails Through GMail

- Install *postfix* and mail utilities.

  ```
  sudo apt-get install postfix mailutils mpack
  ```

- Configure *postfix*.

  ```
  cd /etc/postfix sudo mv main.cf main.cf.orig
  sudo nano main.cf
  ```

  Add the following to *main.cf* then save and exit *nano*:

  ```
  relayhost = [smtp.gmail.com]:587
  smtp_sasl_auth_enable = yes
  smtp_sasl_password_maps = hash:/etc/postfix/sasl/password
  smtp_sasl_security_options = noanonymous
  smtp_tls_CAfile = /etc/postfix/cacert.pem
  ```

```
smtp_use_tls = yes
inet_protocols = ipv4
```

- Create the password file for *postfix*.

  ```
  cd /etc/postfix/sasl
  sudo nano password
  ```

  Add the following to *password* then save and exit *nano*:

  ```
  [smtp.gmail.com]:587 your_username@gmail.com:your_password
  ```

  Replace *your_username* and *your_password* with your real username and password.

  Secure the new file *password* and make it usable for Postfix only. It must be owned by *root* and no one else should have read access to that file:

  ```
  sudo chown root:root password
  sudo chmod 600 password
  ```

  Now convert */etc/postfix/sasl/password* into a format that Postfix can read:

  ```
  sudo postmap password
  ```

  Run the command:

  ```
  ls -l
  ```

  Verify that the results look similar to the following:

  ```
  total 12
  -rw------- 1 root root    49 Oct 25 17:08 password
  -rw------- 1 root root 12288 Oct 25 17:09 password.db
  ```

- Copy CA root certificates to postfix directory:

  ```
  cd /etc/postfix
  sudo cp /etc/ssl/certs/ca-certificates.crt /etc/postfix/cacert.pem
  ```

- Restart Postfix:

  ```
  sudo /etc/init.d/postfix restart
  ```

- Test the postfix server by sending an email to another account:

  ```
  echo "Some sample text." | mail -s "Test" username@gmail.com
  ```

- Test the postfix server by sending an email attachment to another account:

  ```
  mpack -s "test" /home/pi/pitest_index.jsp username@gmail.com
  ```

IlabServiceBroker Deployment

Deployment of the IlabServiceBroker project using the WAR file builds requires that:

- The *UQ-iLab-ServiceBroker-Java* GitHub repository has been downloaded as a ZIP file from:

  https://github.com/uqlpayne/UQ-iLab-ServiceBroker-Java

The following tutorials need to be completed:

1. The *IlabServiceBroker Database* tutorial (see IlabServiceBrokerDatabase.pdf).
2. The *ExperimentStorage Database* tutorial (see ExperimentStorageDatabase.pdf).
3. The *IlabServiceBroker Deployment* tutorial (see IlabServiceBrokerDeployment.pdf).

   Run the *GlassFish Server Administration Console* by entering the following URL in a web

browser.

`http://host.domain:4848/`

Replace *host.domain* with the full hostname of the Raspberry Pi that you are using.

For each deployment, in the *Deploy Applications or Modules* panel, select *Packaged File to Be Uploaded to the Server* and browse for the desired *.war* file. This will deploy the applications to the */opt/glassfish4/glassfish/domains/domain1/applications* directory.
