Install & Configure Solr on Ubuntu, the quickest way

Note: I used the wiki page Ubuntu-10.04-lts-server as basis of this tutorial.
More infos on the general installation at :http://wiki.apache.org/solr/SolrTomcat

One of the most efficient way to deploy a Solr server is to encapsulate it in a Java servlet, the Apache Foundation (the provider of Solr) brought to us Tomcat, a powerfull http server written in Java.

The 2 steps procedure (could be 7 if you want to customize everything)

Install Java (if not already installed)

Because tomcat and solr are Java based softwares we need the Java environment (As it is advised in the Solr wiki : prefere a full JDK to a simple JRE.)

```
$ sudo apt-get install openjdk-6-jdk
```

Note : Working on Amazon EC2 instance, I had to upgrade temporarly my “micro” instance to “medium”, because of special memory needs during java’s installation.

Install Tomcat & Solr (You can’t avoid this one)

For ubuntu (tested with 10.4 LTS), there is an unique package to install, it will take care of installing both Tomcat & Solr, plus the integration of Solr in Tomcat.

```
$ sudo apt-get install solr-tomcat
```

then let’s start the server

```
$ sudo service tomcat6 start
```

Ok it is done !

- Solr server responds at `http://localhost:8080/solr`
- and the config & index will be stored at : `/usr/share/solr/`

In order to use it with Sunspot you'll have to copy `conf/schema.xml` & `conf/solrconfig.xml` from your application to `/usr/share/solr/` or `/usr/share/solr/conf` (Ubuntu 12.04).
But, indeed, you should read the tutorial further to understand why you shouldn't use Solr directly.

For now your data are stored in a folder of your system not dedicated to a specific user or application. It is quite common to read that a public server should use a dedicated user account with constraint privileges. (internet is full of it)

If like me your not totaly happy with it, follow the next steps to customize the whole thing. I won't explain you how to start Tomcat as another user (Google it), but how to choose your Solr home directory

Install tomcat management utilities (not required)

These utilities could be usefull later, it cost nothing to install them.

```
$ sudo apt-get install tomcat6-admin
```

Once installed, you'll have to grant some user

```
$ sudo nano /etc/tomcat6/tomcat-users.xml
```

and paste these lines (change username and password as you whish)

```
<role rolename="manager"/>
<role rolename="admin"/>
<user username="tomcat" password="tomcat" roles="manager,admin"/>
```

A Quick test (not required)

At this point Solr should already be running smoothly. Try Starting tomcat server.

```
$ sudo service tomcat6 start
```

As it is installed as a service, it is quite cool to manage (you can use start|stop|restart verbs)

- Now, reach it with your web browser : `http://localhost:8080/`
  Tomcat should say : **It works !**
- And Solr should be accessible at : `http://localhost:8080/solr`
  if it is ok, take a look to the "Solr Admin" section.

- If you installed the management package, take a look at : `http://localhost:8080/manager/html` and `http://localhost:8080/host-manager/html`

Configure Solr (not required)

The context

Solr is a Java software, when the server starts it create an instance of Solr, which take place in a home directory and get an single uri to access it. let's configure them.

In the Solr web admin interface you could check on the third line the default home directory `SolrHome=/usr/share/solr/`
let's edit the config file :

```
$ sudo nano /var/lib/tomcat6/conf/Catalina/localhost/solr.xml
```

and change the fragment below :

```
<Environment name="solr/home" ..... value="/my-directory/solr" ....>
```

Tomcat use the filename (minus ".xml") as URI to access the instance. If you want to change the uri of Solr just rename the previous file.
If you want several instances of Solr on the same server (several apps, or environments), create a new copy of the previous file
`solr-dev.xml` can be reached at `http://localhost:8080/solr-dev/`
`solr-prod.xml` can be reached at `http://localhost:8080/solr-prod/`
Will give you two Solr indexes (one index in each solr/home) at two differents uri.
This is cool !

The home directory

We said to tomcat to point to an unexisting folder, let's create it by copying the default solr folder.

```
$ sudo cp -prL /usr/share/solr/ /my-directory/
```

copy options are : r = recursive, p = preserve (ownership, timespamps...), L= copy symlink as real folders & files
If you have a specific user running tomcat :

```
sudo chown -R tomcat:tomcat /my-directory
```

The directory you just copied look a bit like our solr/ folder in our app. let’s copy our application configuration files (beware of preserving the R&W access, do a copy/paste of the content if you are not sure) `conf/schema.xml` & `conf/solrconfig.xml` in it.
Note : when I copy `conf/solrconfig.xml` to the new Solr instance it fails to start. But I successfuly use the default file.

The application.

It could be something like :

```
production:
  solr:
    hostname: solrserver
    port: 8080 #tomcat defaults to port 8080
    path: '/solr-prod' # or '/solr' (ubuntu 12.04)
    log_level: WARNING
development:
  solr:
    hostname: localhost
    port: 8982
    log_level: INFO
test:
  solr:
    hostname: localhost
    port: 8981
    log_level: WARNING
```

Enjoy !

for now your index is empty, don’t forget to reindex everything.
After indexing, you’ll be able to see some infos and stats of your index at `http://localhost:8080/solr/admin/schema.jsp`