

1 Installation guide for tablet 2.x

1.1 Installation for linux like OS server

1.1.1 (Red Hat based distributions (RHEL/Centos/Fedora))

Preparation First step is to enable epel-repository with this command in the terminal :

```
yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
```

Before installing « Apache couchdb 1.6.1 » on the server, some packages have to be present on it. These are :

1. GNU Make
2. GNU Compiler Collection
3. libcurl
4. help2man
5. Python (≥ 2.7)
6. Python Sphinx ($\geq 1.1.3$)
7. ICU
8. OpenSSL
9. Erlang OTP ($\geq R14B01$, $\leq R17$)
10. Mozilla SpiderMonkey (1.8.5)

In the command-line type :

```
yum install autoconf autoconf-archive automake curl-devel erlang-asn1
erlang-erts erlang-eunit erlang-os_mon erlang-xmerl
gcc-c++ help2man js-devel libicu-devel libtool perl-Test-Harness
```

As 2 dependencies require a specific version of packages, you have to check the version of « Mozilla SpiderMonkey » and « Erlang OTP ».

```
#Mozilla SpiderMonkey version
yum --showduplicates list js-devel
```

```
#Erlang version
erl
```

If the « automatic way » installed too recent package-versions, it's better first to remove them and then to install them from sources.

First compile « Erlang OTP » package from source like this :

```
#it's better to run these commands from Download directory
wget http://www.erlang.org/download/otp_src_R16B03.tar.gz
tar -xvf otp_src_R16B03.tar.gz
cd otp_src_R16B03
#connect with root
./configure
make && make install
```

Then compile « Mozilla SpiderMonkey » :

```
#it's better to run these commands from Download directory
wget http://ftp.mozilla.org/pub/mozilla.org/js/js185-1.0.0.tar.gz
tar -xvf wget js185-1.0.0.tar.gz
cd js185-1.0.0/js/src/
#connect with root
./configure
make && make install
```

Install couchdb Download source « apache-couchdb-1.6.1.tar.gz » from <https://cwiki.apache.org/confluence/display/COUCHDB/Current+Releases> and install :

```
tar -xvf apache-couchdb-1.6.1.tar.gz
cd apache-couchdb-1.6.1
#connect with root
./configure --with-erlang=/usr/lib64/erlang/usr/include
make && make install
```

Configure couchdb service Add user and proper file ownership and permission :

```
adduser -r --home /usr/local/var/lib/couchdb -M
--shell /bin/bash --comment "CouchDB Administrator" couchdb
chown -R couchdb:couchdb /usr/local/etc/couchdb
chown -R couchdb:couchdb /usr/local/var/lib/couchdb
chown -R couchdb:couchdb /usr/local/var/log/couchdb
chown -R couchdb:couchdb /usr/local/var/run/couchdb
chmod 0770 /usr/local/etc/couchdb
chmod 0770 /usr/local/var/lib/couchdb
chmod 0770 /usr/local/var/log/couchdb
chmod 0770 /usr/local/var/run/couchdb
```

Add symbolic link in the startup directory and fix the run levels :

```
ln -s /usr/local/etc/rc.d/couchdb /etc/init.d/couchdb
chkconfig --add couchdb
chkconfig --level 345 couchdb on
```

Add your user to group couchdb :

```
sudo adduser <username> <group>
#or
sudo vim /etc/group
```

Configure firewall and start service Get a list of active zones for the firewall :

```
sudo firewall-cmd --get-active-zones
```

Open port 5984 for firewall and reload it :

```
sudo firewall-cmd --zone=public --add-port=5984/tcp --permanent
firewall-cmd --reload
```

Start couchdb service :

```
sudo /usr/local/etc/rc.d/couchdb start
#or
sudo /etc/init.d/couchdb start
#or
sudo service couchdb start
```

Check that the service is running :

```
sudo service couchdb status

sudo netstat -plantu
```

In the produced list with the last line, you should see the ip-address of your server with :5984.

1.1.2 Debian based distributions (debian, ubuntu)

Preparation As Red Hat, we have to update and install some prerequisites for « erlang package » :

```
sudo apt-get update
sudo apt-get install fop
sudo apt-get install libncurses5-dev
sudo apt-get install openjdk-6-jdk
sudo apt-get install unixodbc-dev
sudo apt-get install g++
sudo apt-get install libssl-dev
sudo apt-get install libqt4-opengl-dev
sudo apt-get install libgtk2.0-dev
sudo apt-get install libxslt*
sudo apt-get install libxsltproc
sudo apt-get install flex

sudo apt-get install build-essential erlang-base-hipe erlang-dev erlang-manpages
erlang-eunit erlang-nox libicu-dev libmozjs185-dev libcurl4-openssl-dev
```

First compile « Erlang OTP » package from source like this :

```
#it's better to run these commands from Download directory
wget http://www.erlang.org/download/otp_src_R16B03.tar.gz
tar -xvf otp_src_R16B03.tar.gz
cd otp_src_R16B03
#fix new password for root and unlock
sudo passwd root
sudo passwd -u root
./configure
make && make install
```

```
#lock root
sudo passwd -l root
#test erlang
erl
```

The second installation step tries to install « Mozilla SpiderMonkey » (`libmozjs185-dev`), so we just check if installation succeeded :

```
sudo apt-cache policy libmozjs185-dev
```

If the package isn't installed, proceed to compile as for Red Hat :

```
#it's better to run these commands from Download directory
wget http://ftp.mozilla.org/pub/mozilla.org/js/js185-1.0.0.tar.gz
tar -xvf wget js185-1.0.0.tar.gz
cd js185-1.0.0/js/src/
#connect with root
sudo passwd -u root
./configure
make && make install
sudo passwd -l root
```

Install couchdb Unpack `apache-couchdb-1.6.1.tar.gz`, install :

```
tar -xzf apache-couchdb-1.6.1.tar.gz
cd apache-couchdb-1.6.1
#unlock root
sudo passwd -u root
sudo ./configure
sudo make && make install
#lock root
sudo passwd -l root
```

Configure couchdb service Configure service :

```
sudo adduser couchdb
sudo chown -R couchdb:couchdb /usr/local/var/lib/couchdb/ /usr/local/
var/log/couchdb/ /usr/local/var/run/couchdb/ /usr/local/etc/couchdb
sudo ln -s /usr/local/etc/init.d/couchdb /etc/init.d/couchdb
sudo update-rc.d couchdb defaults
```

Add your user to group couchdb :

```
sudo adduser <username> <group>
#or
sudo vim /etc/group
```

Start couchdb service :

```
sudo /etc/init.d/couchdb start
sudo /etc/init.d/couchdb status
```

Configure firewall Enable firewall and open port 5984 :

```
sudo ufw enable
sudo ufw allow 5984
```

Check that the service is running :

```
sudo service couchdb status
```

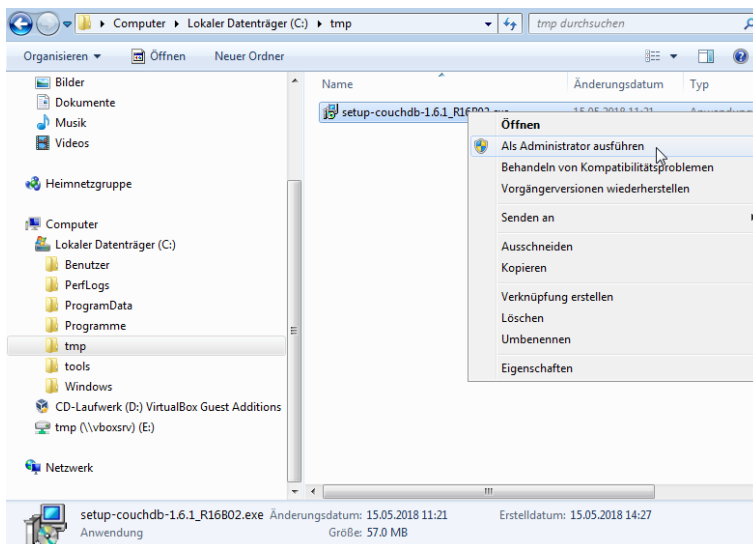
```
sudo netstat -plantu
```

In the produced list with the last line, you should see the ip-address of your server with :5984.

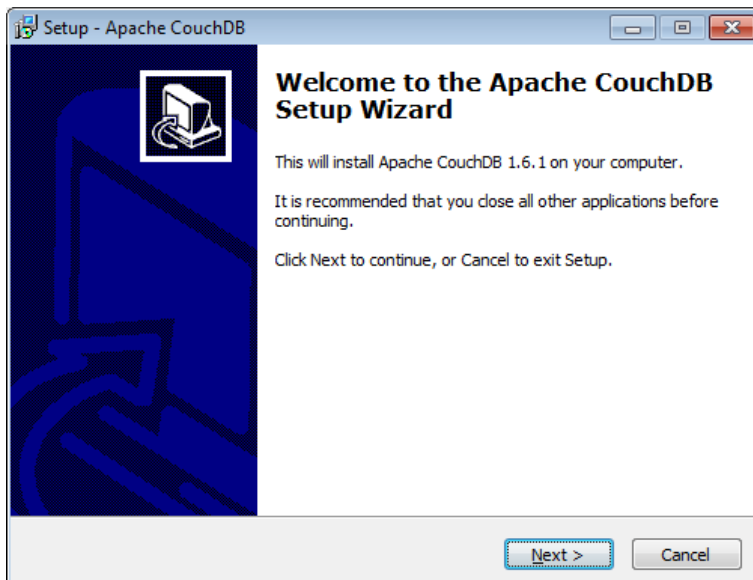
1.2 Installation for Windows

1.2.1 Installation from binaries

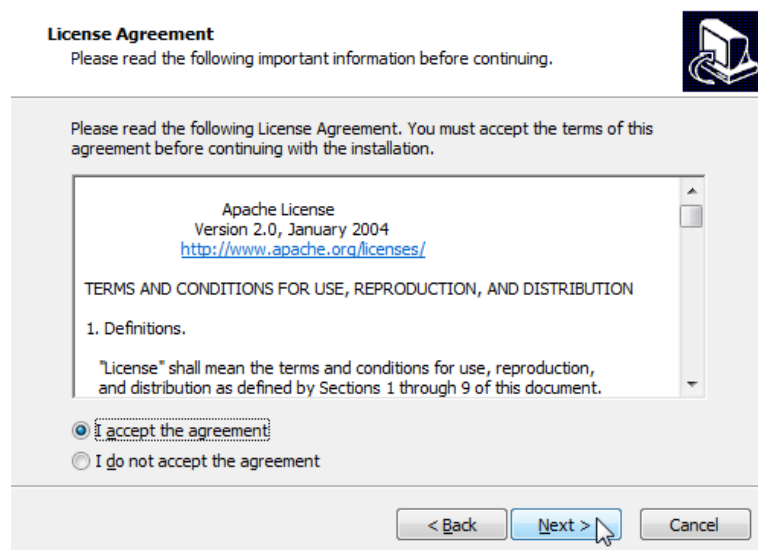
Download binary `setup-couchdb-1.6.1_R16B02.exe` from internet site <http://archive.apache.org/dist/couchdb/binary/win/1.6.1/>, and execute as administrator :



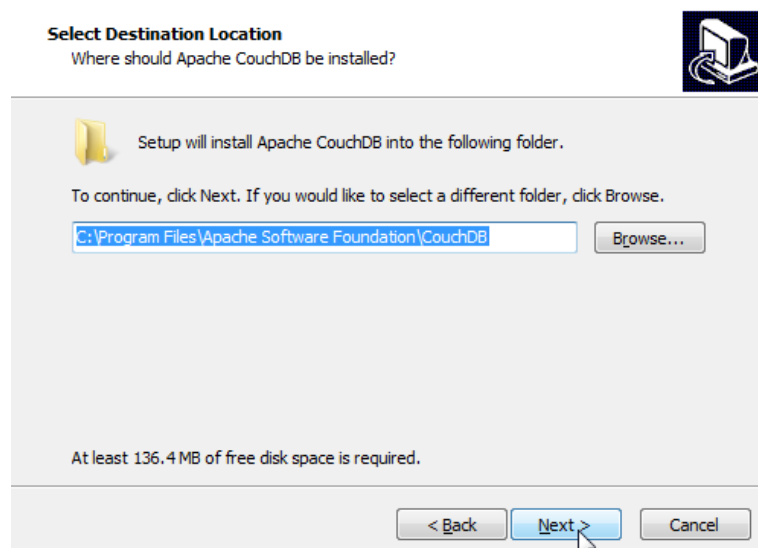
The Wizard is starting :



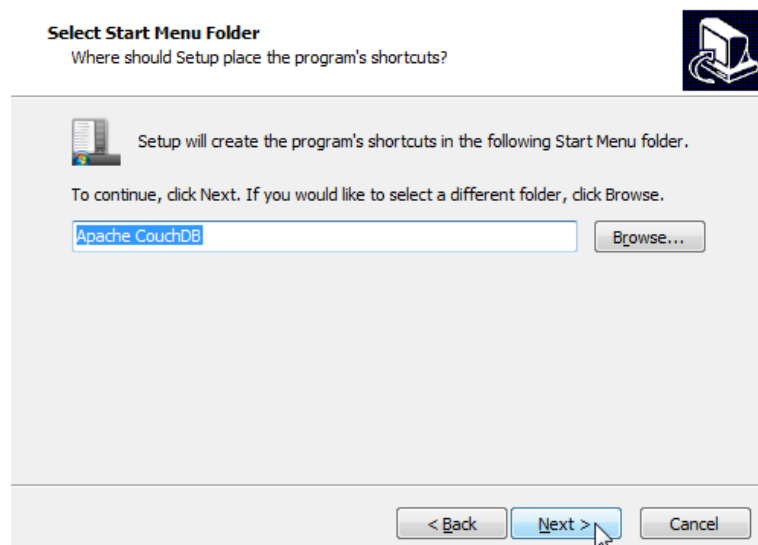
Accept license conditions :



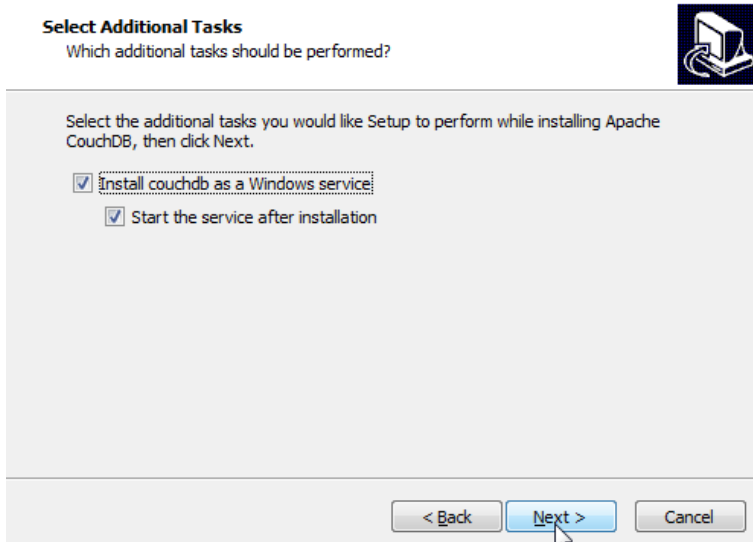
Choose the directory for the installation :



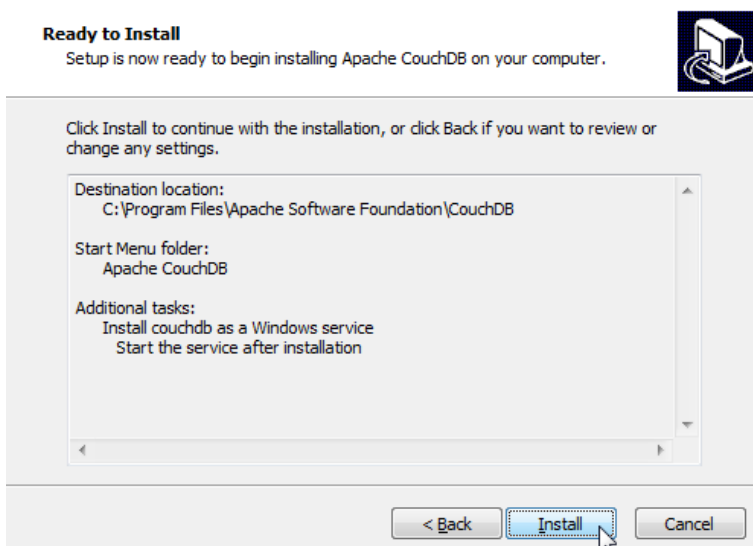
Select the name for the start menu shortcuts :



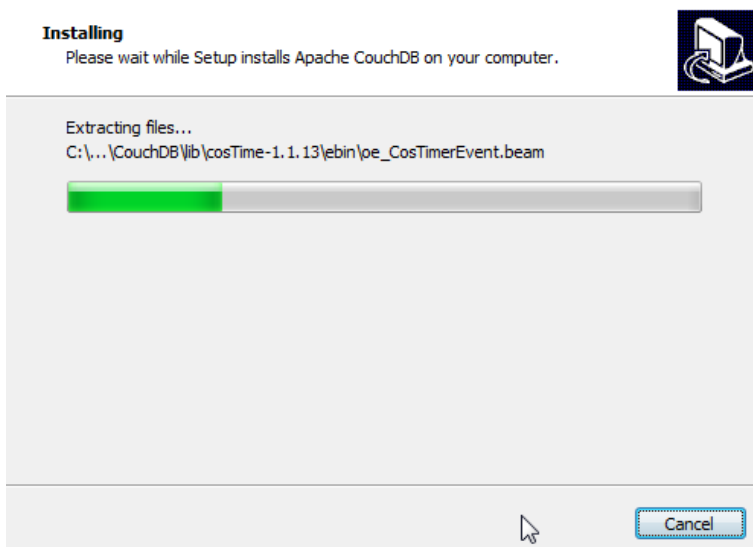
Activate both flags, so that couchdb runs as a service and starts automatically after booting :

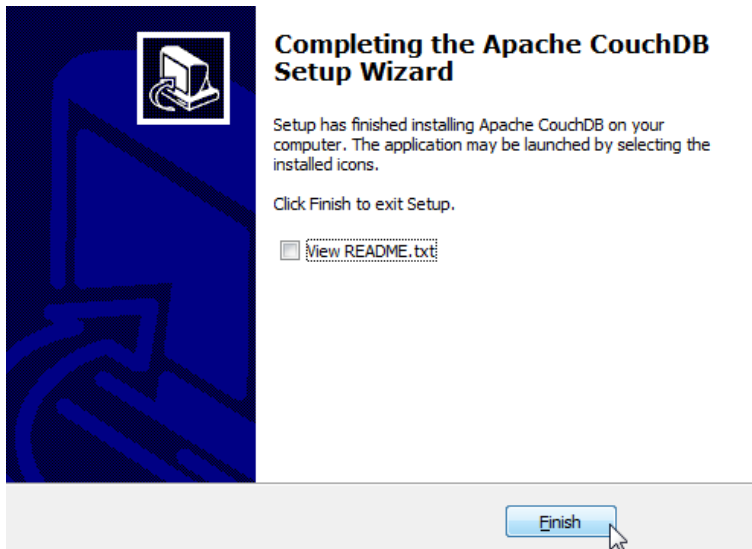


Confirm the installation with the chosen parameters :



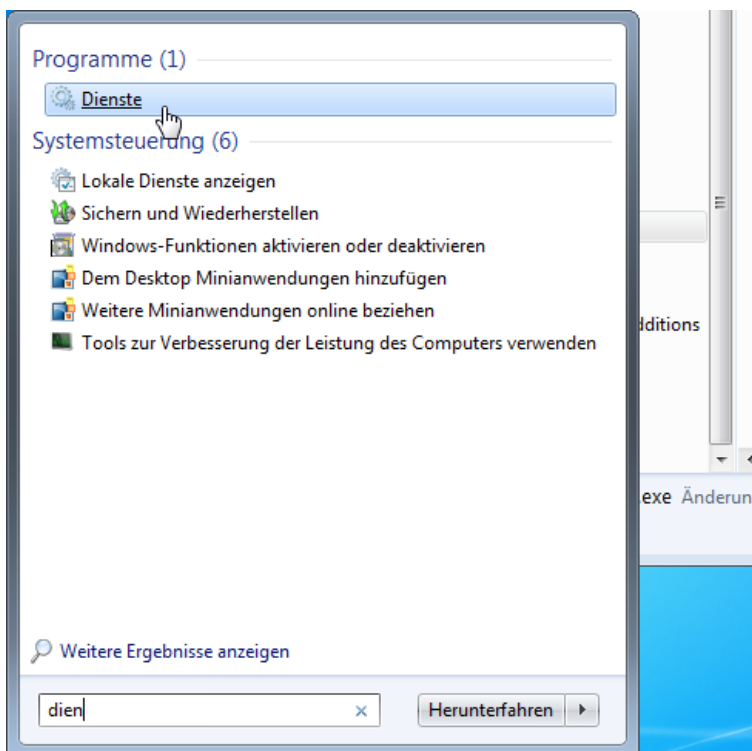
Wait till the installation process is complete and finish it :



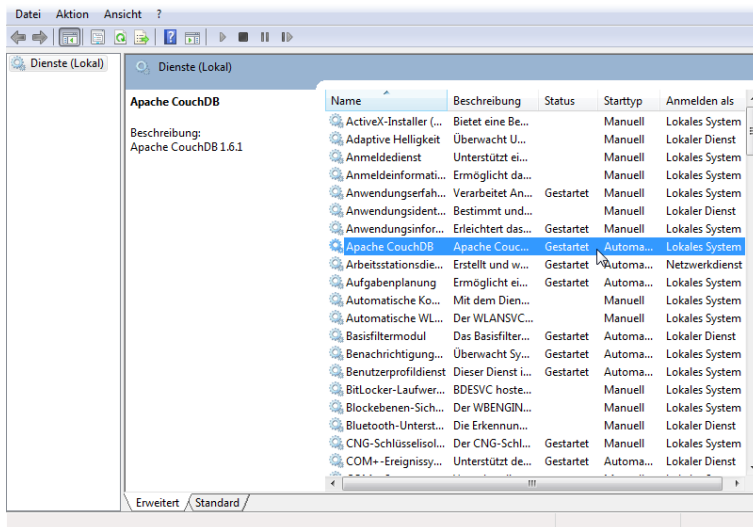


1.2.2 Checking the service

Now to check if the service is running, type « services » in the search from the start menu :

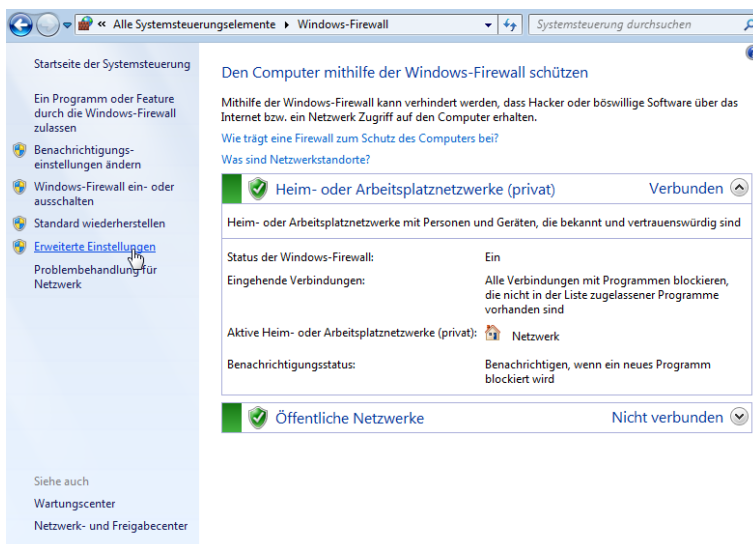


The gearwheel appears and verify that « Apache CouchDB » is in the list and starts automatically :

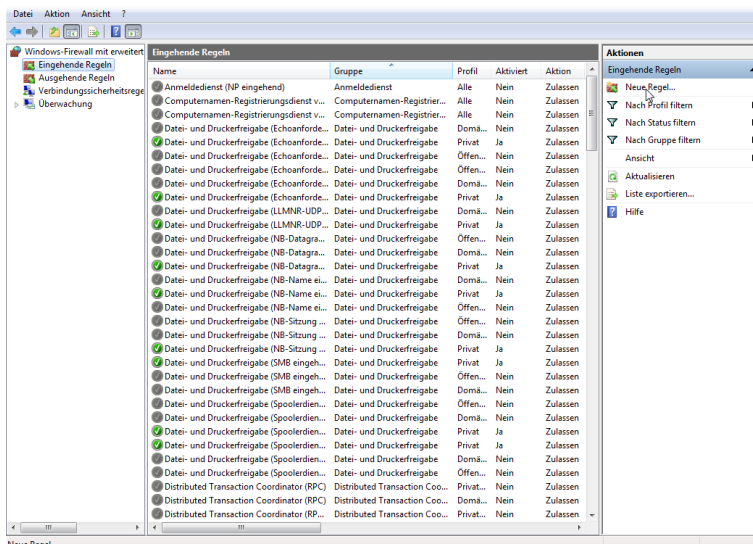


1.2.3 Configure firewall

In the configuration panel, open the Windows firewall and choose « advanced settings » :



Click « Inbound Rules » and « New Rule » in the right panel :



Choose « Port » as rule type :

Regeltyp
Wählen Sie den Typ der zu erstellenden Firewallregel aus.

Schritte:

- Regeltyp
- Protokoll und Ports
- Aktion
- Profil
- Name

Welchen Regeltyp möchten Sie erstellen?

☐ **Programm**
Regel, die die Verbindungen für ein Programm steuert.

☒ **Port**
Regel, die die Verbindungen für einen TCP- oder UDP-Port steuert.

☐ **Vordefiniert:**

Regel, die die Verbindungen für einen Windows-Vorgang steuert.

☐ **Benutzerdefiniert**
Benutzerdefinierte Regel

[Weitere Informationen über Regeltypen](#)

< Zurück Weiter > Abbrechen

Let activate « TCP » on the top of the window and in « specific local port », type 5984 :

Protokoll und Ports
Geben Sie die Protokolle und Ports an, für die diese Regel gilt.

Schritte:

- Regeltyp
- Protokoll und Ports
- Aktion
- Profil
- Name

Betrifft diese Regel TCP oder UDP?

☒ **TCP**

☐ **UDP**

Gilt diese Regel für alle lokalen Ports oder für bestimmte lokale Ports?

☐ **Alle lokalen Ports**

☒ **Bestimmte lokale Ports:**
Beispiel: 80, 443, 5000-5010

[Weitere Informationen über Protokolle und Ports](#)

< Zurück Weiter > Abbrechen

Allow connection for all domains :

Aktion

Legen Sie die Aktion fest, die ausgeführt werden soll, wenn eine Verbindung die in der Regel angegebenen Bedingungen erfüllt.

Schritte:

- Regeltyp
- Protokoll und Ports
- Aktion**
- Profil
- Name

Welche Aktion soll durchgeführt werden, wenn eine Verbindung die angegebenen Bedingungen erfüllt?

☒ **Verbindung zulassen**
Dies umfasst sowohl mit IPsec geschützte als auch nicht mit IPsec geschützte Verbindungen.

☐ **Verbindung zulassen, wenn sie sicher ist**
Dies umfasst nur mithilfe von IPsec authentifizierte Verbindungen. Die Verbindungen werden mit den Einstellungen in den IPsec-Eigenschaften und -regeln im Knoten "Verbindungssicherheitsregel" gesichert.
[Anpassen...](#)

☐ **Verbindung blockieren**

[Weitere Informationen über Aktionen](#)

[< Zurück](#) [Weiter >](#) [Abbrechen](#)

Profil

Geben Sie die Profile an, für die diese Regel zutrifft.

Schritte:

- Regeltyp
- Protokoll und Ports
- Aktion
- Profil**
- Name

Wann wird diese Regel angewendet?

☒ **Domäne**
Wird angewendet, wenn ein Computer eine Verbindung mit der Firmendomäne hat.

☒ **Privat**
Wird angewendet, wenn ein Computer eine Verbindung mit einem privaten Netzwerk hat.

☒ **Öffentlich**
Wird angewendet, wenn ein Computer eine Verbindung mit einem öffentlichen Netzwerk hat.

[Weitere Informationen zu Profilen](#)

[< Zurück](#) [Weiter >](#) [Abbrechen](#)

And finally choose a rule's name and verify that the rule is active :

Name

Geben Sie den Namen und die Beschreibung dieser Regel an.

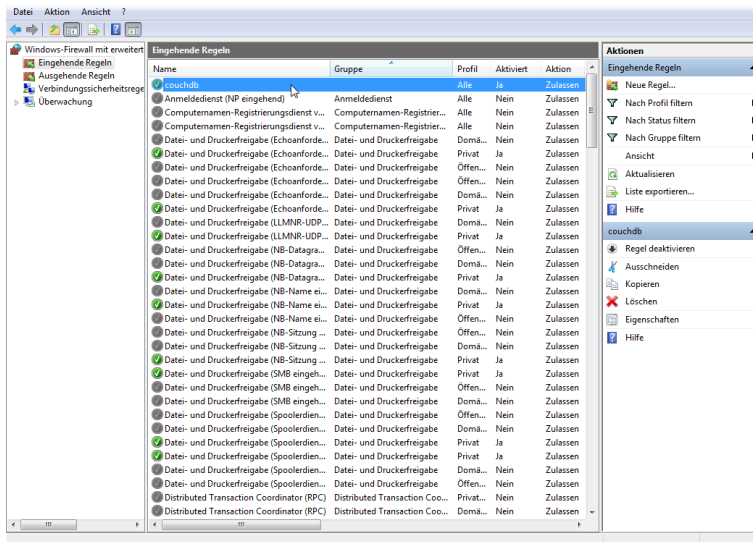
Schritte:

- Regeltyp
- Protokoll und Ports
- Aktion
- Profil
- Name**

Name:

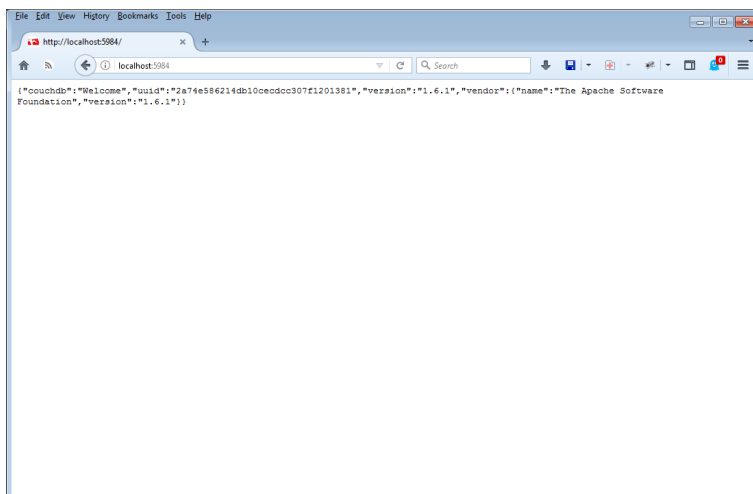
Beschreibung (optional):

[< Zurück](#) [Fertig stellen](#) [Abbrechen](#)

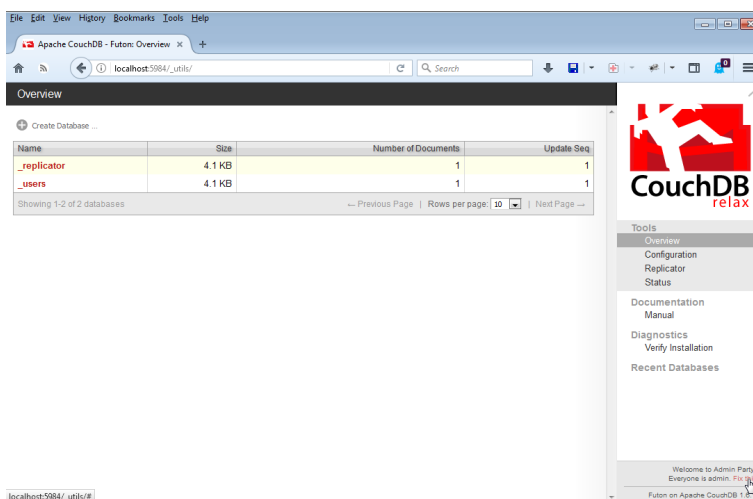


1.3 couchdb configuration for Preisadmin (aka AdminApp)

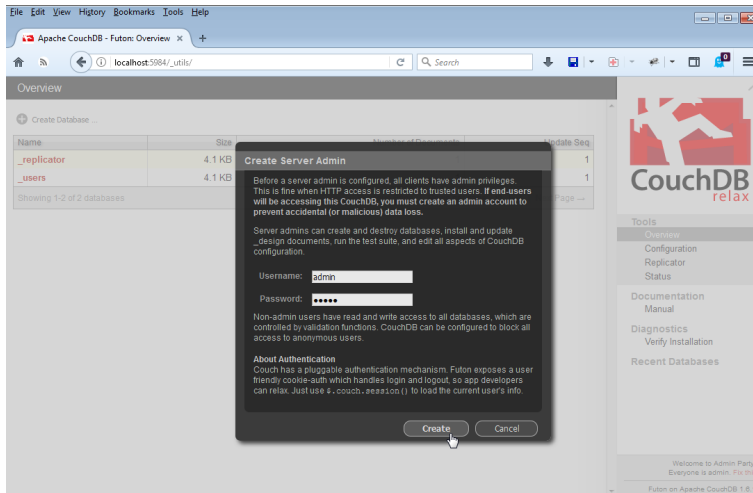
At this point, the service is running and the server is ready to receive the Preisadmin. The application requires some more specific configuration. But first, if your installation is correct, you can check it by opening a browser, typing in the address bar <http://localhost:5984> and the following message should appear :



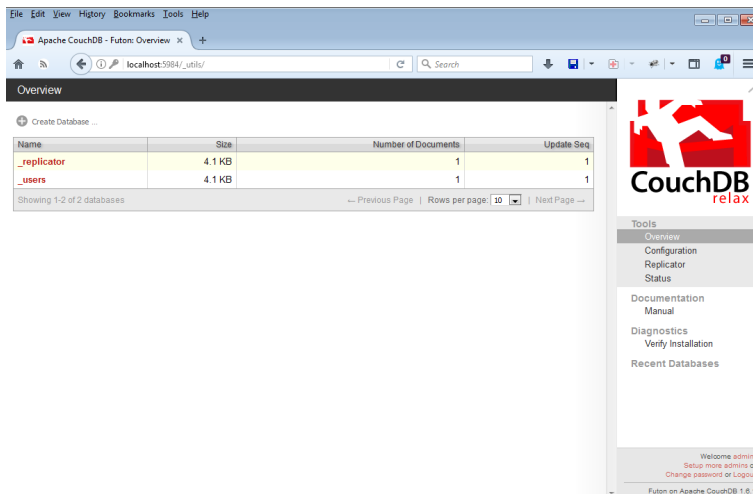
To start the couchdb specific configuration, type in the address bar http://localhost:5984/_utils :



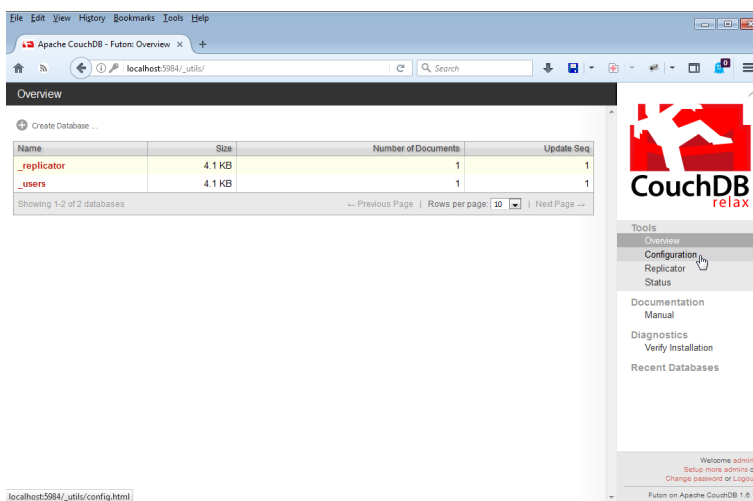
Define an administrator for the couchdb server by clicking on « Fix this » at the right bottom of the window :



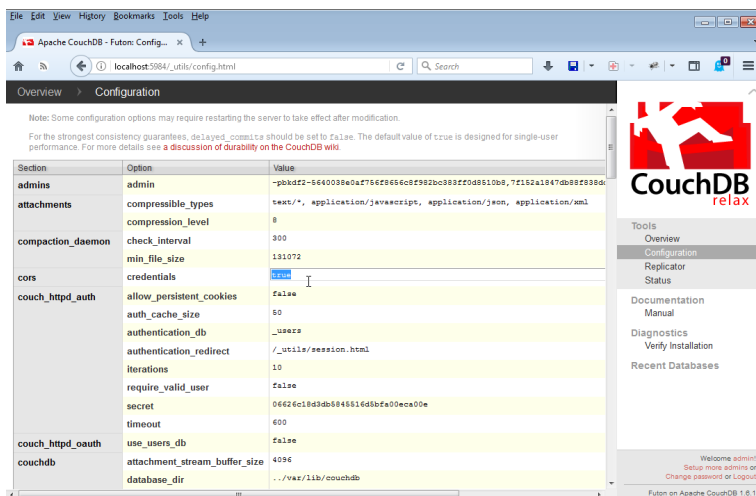
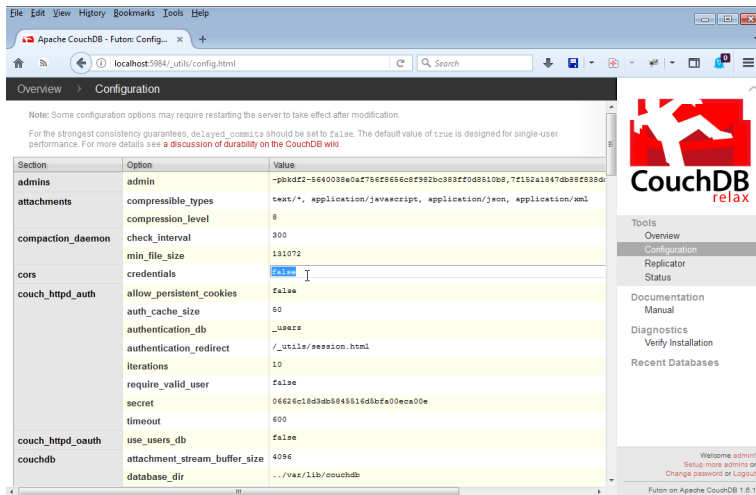
Now you're logged as server's administrator as you can see at the right bottom.



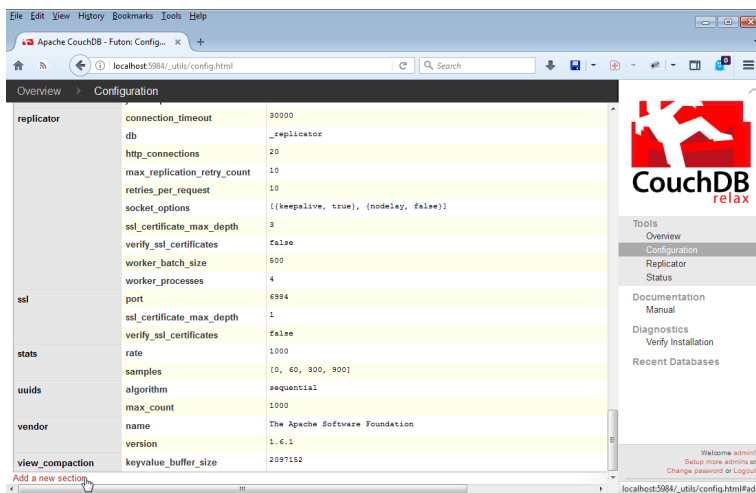
Then click on « Configuration » :



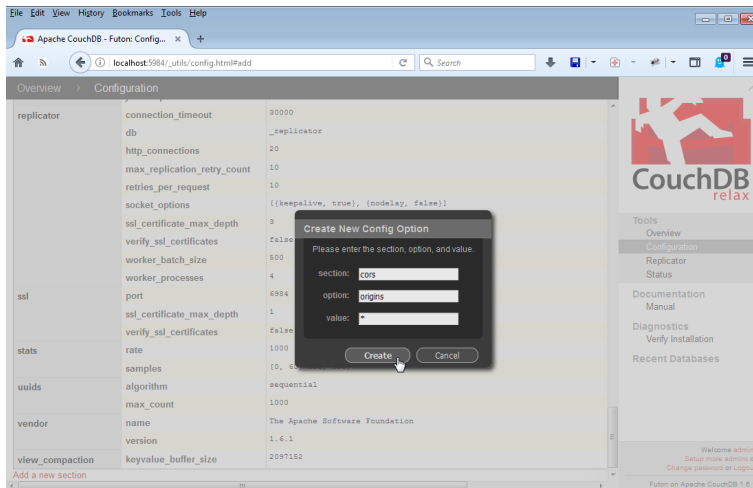
In the « cors » section, set « credentials » value to « true » by double clicking on the « Value » field :



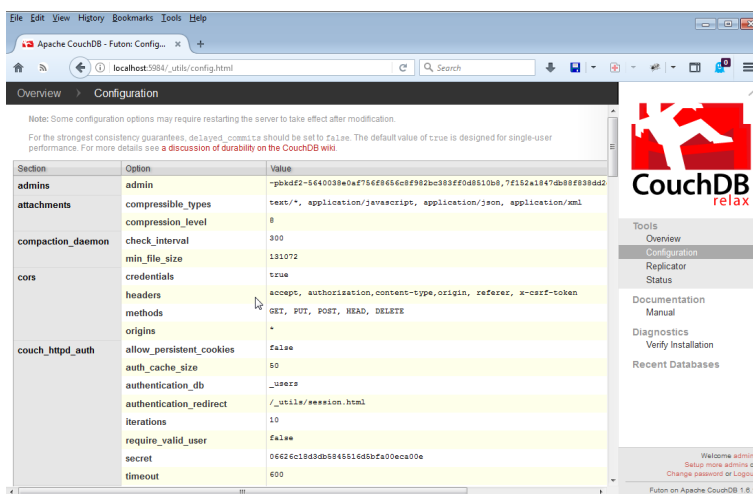
Then scroll till the end of the configuration's page and click « Add a new section » :



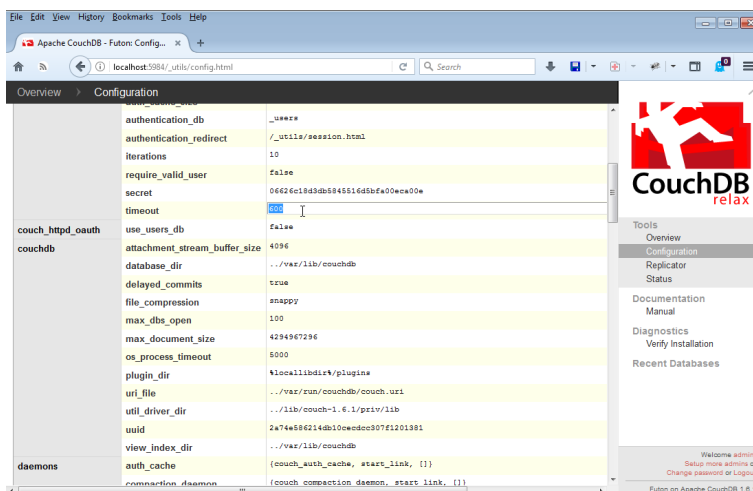
A pop-up window opens and fill it like mentioned below :



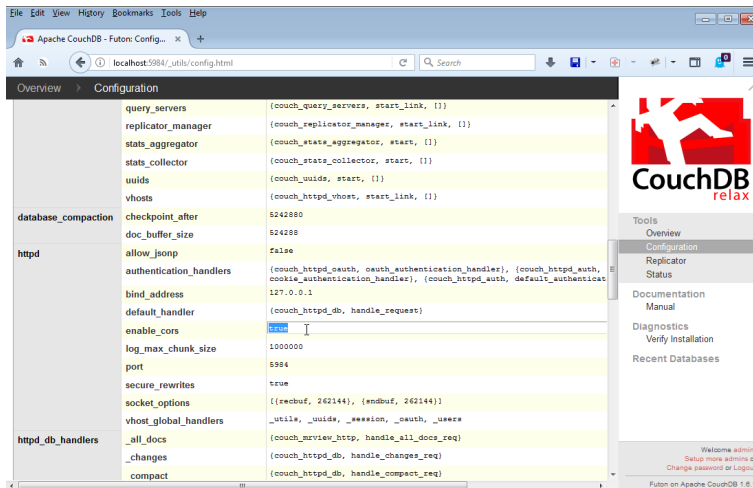
Repeat this step to have the same configuration as below for the « cors » section :



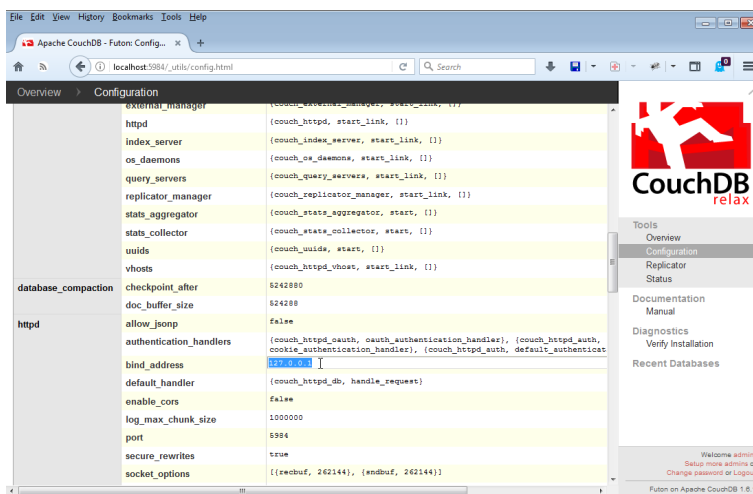
Increase the timeout value of section « », so that the connection stays open longer :



Set value true for « enable_cors » of section « httpd » :



Finally, modify « bind_address » with your personal IP address in the section « httpd » :



To access the configuration page of the server, you must now enter your IP address in the address bar :

