

RDFストア環境構築(Virtuoso編)3

この記事は1年以上前に書かれました。
内容が古くなっている可能性がありますのでご注意ください。

Virtuoso編の第3回では、マイクロソフトのクラウドプラットフォームAzureに環境を構築します。

はじめに、CentOSの仮想マシンを作成します。

仮想マシンの作成

イメージの選択

すべて

- MICROSOFT
- WINDOWS SERVER
- SHAREPOINT
- SQL SERVER
- BIZTALK SERVER
- VISUAL STUDIO
- DYNAMICS
- UBUNTU
- CENTOS**
- SUSE
- ORACLE
- PUPPET LABS

マイイメージ
マイディスク

MSDN
 サポート提供可能イメージ

おすすめ

OpenLogic

OpenLogic

This distribution of Linux is based on CentOS version 6.5 and is provided by OpenLogic. It contains an installation of the Basic Server packages.

OS ファミリー Linux
発行者 OpenLogic
ディスクの数 1
場所 East Asia;Southeast Asia;Brazil South;North Europe;West Europe;Japan East;Japan West;East US;North Central US;South Central US;West US

料金情報
料金は、仮想マシンをプロビジョニングするために選択したサブスクリプションによって異なります。

エンドポイントは、

- リモートで作業するためのSSH (ポート22)
- SPARQLエンドポイントを公開するためのHTTP (ポート80)
- Virtuoso管理画面用のポート (ポート8890)

を設定します。

名前	↑	プロトコル	パブリック ポート	プライベート ポート	負荷分散セット名
HTTP		TCP	80	80	-
SSH		TCP	22	22	-
virtuoso		TCP	8890	8890	-

ここからはSSHで作業します。

root権限の必要な作業をする際に毎回sudoするのは面倒ですので、はじめにrootのパスワードを設定してsuできるようにします。

```
$ sudo passwd root
```

このままでは、SSHでrootにログインできますので、/etc/ssh/sshd_configファイルの

```
#PermitRootLogin yes
```

となっている箇所を

```
PermitRootLogin no
```

に変更し、rootのログインを不許可にして

```
# service sshd restart
```

sshdを再起動します。

azureのCentOSにはVirtuosoのビルドに必要な開発環境が入っていないため、yumでインストールします。

```
# yum install gcc gmake autoconf automake libtool flex bison gperf gawk m4 make openssl-devel readline-devel git
```

次に、GitHubからソースコードを入手します。

```
$ git clone git://github.com/openlink/virtuoso-opensource.git
```

あとは、お決まりの手順となります。

```
$ cd virtuoso-opensource
$ ./autogen.sh
$ ./configure --prefix=/usr/local/ --with-readline
$ make
# make install
```

ビルドが終了したら、以下のコマンドでVirtuosoを起動します。

```
# virtuoso-t +configfile /usr/local/var/lib/virtuoso/db/virtuoso.ini
```

Virtuosoを起動したら、ブラウザでポート8890に接続します。



左上の「Conductor」のリンクをクリックすると、アカウントとパスワードの入力を求められます。デフォルトでdbaとdavの2つのアカウントが用意されており、パスワードはアカウント名と同じです。

ログインしたら、【System Admin】【User Accounts】タブ画面でdbaとdavのパスワードを変更し、不要なアカウントは無効にしておきます。

最後に、SPARQLエンドポイント公開の設定をします。
このままでは、SPARQLエンドポイントのURLは

```
http://サーバ名:8890/sparql
```

ですので、apacheのリバースプロキシを使用して

```
http://サーバ名/sparql
```

で接続できるようにします。

まず、apacheをインストールし、自動起動を有効にします。

```
# yum install httpd
# chkconfig httpd on
```

次に、SELinuxを無効化します。

```
# setenforce 0
```

さらに/etc/selinux/configファイルの

```
SELINUX=enforcing
```

となっている箇所を

```
SELINUX=disabled
```

と変更し、再起動後もSELinuxが無効になるようにしておきます。

最後に、mod_proxyを設定します。

/etc/httpd/conf.d/mod_proxy.confファイルを開き、以下の記述をします。

```
ProxyPass /sparql http://localhost:8890/sparql retry=5
ProxyPassReverse /sparql http://localhost:8890/sparql
```

apacheを起動したら完了です。

```
# service httpd start
```

以上で、

```
http://サーバ名/sparql
```

でSPARQLエンドポイントに接続できるようになりました。

RDFストア環境構築 [1](#) [2](#) [3](#)

カテゴリー: オープンデータ | タグ: LOD, SPARQL | 投稿日: 2014年6月9日

[<https://midoriit.com/2014/06/rdf%e3%82%b9%e3%83%88%e3%82%a2%e7%92%b0%e5%a2%83%e6%a7%8b%e7%af%89virtuoso%e7%b7%a83.html>] | 投稿者: 小池隆
