BaseX for dummies part 2

A short introduction to BaseX in general, and to the BaseX / PHP combination in particular

Introduction

Our goal is to have a working BaseX application so that we can create, manage and query XML databases, from anywhere on this planet.
Therefore we need to deploy one of the available BaseX distributions. I chose for the WAR distribution since that is the format that can be deployed by Elastic Beanstalk out of the box. (It may quite well be possible to convert the JAR distribution to a WAR variant, but that would have meant yet another hurdle to take, and as you will see there already are enough hurdles to be taken. However, using the WAR distribution comes with a price: you will not get the BaseX GUI to play with.)

Recapitulation of part 1

In part 1 of this introduction we looked at BaseX in general, at XQuery in particular, and at accessing BaseX databases and queries with a PHP client in detail.
We made a generic PHP script that can run any XQuery (inside an .xq file) and deliver the results.
Everything was located on localhost though. In this second part we will deploy BaseX in the cloud, on an Amazon web server. Once installed and configured the BaseX application can then be accessed from anywhere.

Selecting a provider

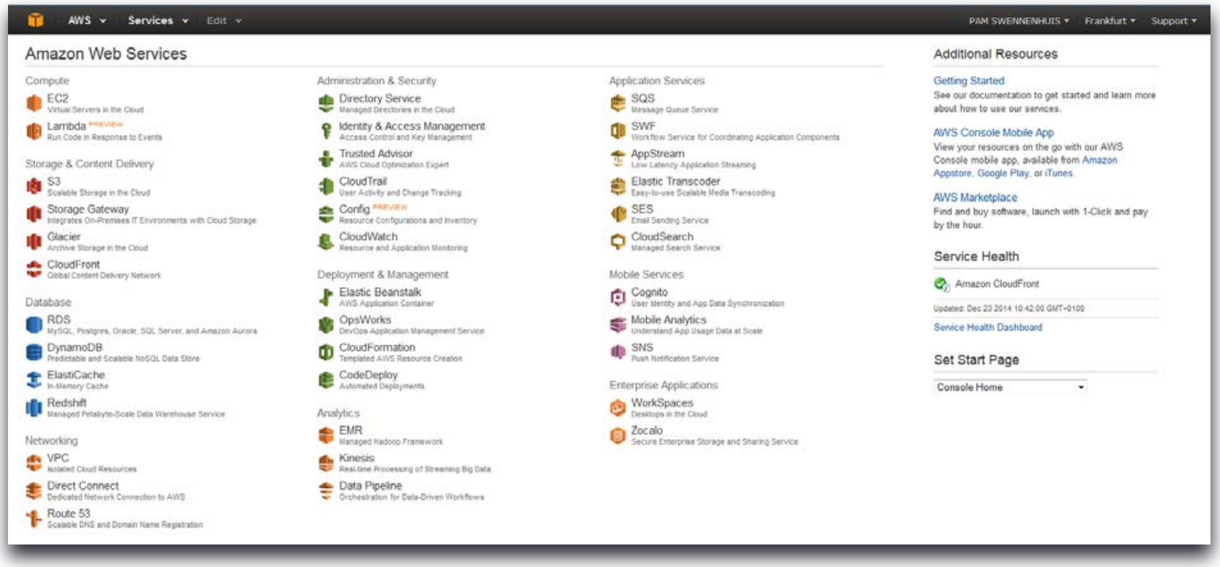
I will be short on this subject. There are lots of platforms available where you can setup a (virtual) server and deploy an application. They come with varying features and price tags. I first tried OpenShift but found it not to be very user-friendly. After consulting a friend I tried AWS (Amazon Web Services) and that turned out to be a much better choice.

AWS and EC2 setup

Amazon Web Services (AWS)

The Amazon Web Services is a huge services and applications facility. It can host your databases, web services, applications and more, and it comes with some nice features to keep your managing activities - such as monitoring application health, deploying your (web)apps and application scaling - down to a minimum.
To make use of AWS you need an account. To do so, go to the AWS main page at http://aws.amazon.com and select 'Sign up' at the upper right corner.
Please note that AWS services are not free, though some of them are.
Once you have signed up, I would advise you to read the Getting Started section at http://aws.amazon.com/documentation/gettingstarted/, especially the "Getting started with AWS" section. The platform is loaded with features and is quite intimidating, so you can easily get lost.

The starting point in AWS is the AWS Management Console
at https://eu-central-1.console.aws.amazon.com/console/home



Amazon Web Services

Compute

- EC2 – Virtual Servers in the Cloud
- Lambda PREVIEW – Run Code in Response to Events

Storage & Content Delivery

- S3 – Scalable Storage in the Cloud
- Storage Gateway – Integrates On-Premises IT Environments with Cloud Storage
- Glacier – Archive Storage in the Cloud
- CloudFront – Global Content Delivery Network

Database

- RDS – MySQL, Postgres, Oracle, SQL Server, and Amazon Aurora
- DynamoDB – Predictable and Scalable NoSQL Data Store
- ElastiCache – In-Memory Cache
- Redshift – Managed Petabyte-Scale Data Warehouse Service

Networking

- VPC – Isolated Cloud Resources
- Direct Connect – Dedicated Network Connection to AWS
- Route 53 – Scalable DNS and Domain Name Registration

Administration & Security

- Directory Service – Managed Directories in the Cloud
- Identity & Access Management – Access Control and Key Management
- Trusted Advisor – AWS Cloud Optimization Expert
- CloudTrail – User Activity and Change Tracking
- Config PREVIEW – Resource Configurations and Inventory
- CloudWatch – Resource and Application Monitoring

Deployment & Management

- Elastic Beanstalk – AWS Application Container
- OpsWorks – DevOps Application Management Service
- CloudFormation – Templated AWS Resource Creation
- CodeDeploy – Automated Deployments

Analytics

- EMR – Managed Hadoop Framework
- Kinesis – Real-time Processing of Streaming Big Data
- Data Pipeline – Orchestration for Data-Driven Workflows

Application Services

- SQS – Message Queue Service
- SWF – Workflow Service for Coordinating Application Components
- AppStream – Low Latency Application Streaming
- Elastic Transcoder – Easy-to-use Scalable Media Transcoding
- SES – Email Sending Service
- CloudSearch – Managed Search Service

Mobile Services

- Cognito – User Identity and App Data Synchronization
- Mobile Analytics – Understand App Usage Data at Scale
- SNS – Push Notification Service

Enterprise Applications

- WorkSpaces – Desktops in the Cloud
- Zocalo – Secure Enterprise Storage and Sharing Service

Additional Resources

Getting Started
See our documentation to get started and learn more about how to use our services.

AWS Console Mobile App
View your resources on the go with our AWS Console mobile app, available from Amazon Appstore, Google Play, or iTunes.

AWS Marketplace
Find and buy software, launch with 1-Click and pay by the hour.

Service Health

Amazon CloudFront

Updated: Dec 23 2014 10:42:00 GMT+0100

Service Health Dashboard

Set Start Page

Console Home

(Well, actually, the URL can be different if you selected another region than "eu-central'. More on regions later.)

On this page, we are going to use just two of the available services:

1) EC2 (virtual servers in the cloud)
2) Elastic Beanstalk (AWS Application Container)

With EC2 you can create the virtual server that will host your BaseX application, and with Elastic Beanstalk you can deploy and manage the application itself.

In addition you may want to have a look at the billing pages at https://console.aws.amazon.com/billing/home where your AWS charges are being tracked.

EC2

So let us instantiate a virtual server.
To do so, follow these steps:

1. Select region

First, make sure you are in the region of your choice. The virtual server instance you will be creating will be run in one of the Amazon regions. **Regions have limited availability!** That is, your virtual server instance is not automatically reachable from every location in the world. So make sure to select a region that will be reachable from the computers that you expect your BaseX instance to be approached from.
Also, the key/value pairs that provide SSH access to your instance – more on that later – will work only in the region where you create them. This is very important! It took me quite some time before I figured out why I could not remotely access my instance. It appeared to be a mismatch of the region where my instance was running – I selected EU Frankfurt – with the region where the key/value pair was registered (US East).

You can select a region in the upper right corner of the EC2 Dashboard.

2. Go to EC2 Dashboard

On the AWS Management Console select *EC2 (Virtual Servers in the Cloud)*.
You will be lead to the EC2 Dashboard

3. Launch instance

In the middle of the EC2 Dashboard click the button *Launch Instance.*
This will lead you to the page where you can select what type of virtual server you want the instance to be run on.

4. Select instance platform

Select the Amazon Linux platform by clicking the *Select* button on the right of the Amazon Linux entry

5. Select instance type

Now you have to select an instance configuration, i.e. the CPU/memory configuration for the Amazon Linux instance. Since only the *t2.micro* type is free, select that one.
If you expect your app to consume a lot of CPU and/or memory, this configuration may not be sufficient and you will have to select one that better suits your needs, at additional costs.

6. Review

Unless you want to change the instance details – and know what you are doing – you are now ready to launch the virtual server. Select *Review and Launch* at the bottom of the page to do so.
Accept the presented detail settings without modifying them.
At the end of the process the instance will be created for you. You can see its progress real-time.
If all goes well – and it should – at the end you will have a running virtual server instance.

SSH Access

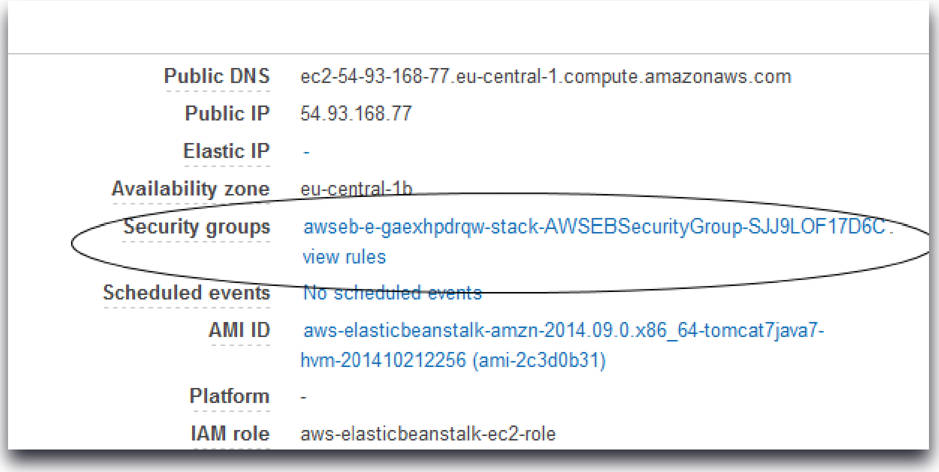
Your virtual server instance is running. Now you will probably want to remotely access it with SSH. To enable this you will have to create a public/private key pair.

Public key and private key

To obtain a public/private key pair, go to the EC2 main page at https://eu-central-1.console.aws.amazon.com/ec2/ (the URL is different for other regions), **make sure you are in the right region** (look at the top right of the page to find out what region you are in), then, on the left, click *Key pairs* and next *Create key pair*. Give it a name you can easily remember and click *Create*.
Next you can save the private part, a *.pem* file, on your computer. You will need this file for SSH connections.
Please remember that the keys will work only for the selected region!

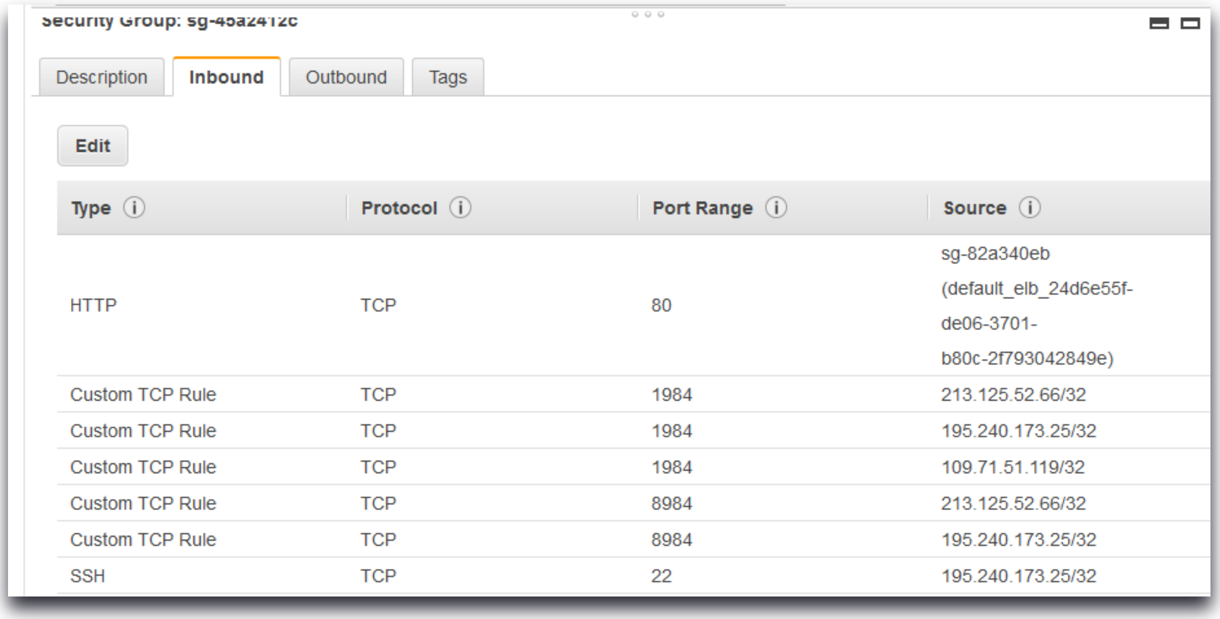
Security groups

Another important issue to be aware of when using EC2 is that your instances have so-called *security groups*. In the security groups you can – and should – limit access to your EC2 instance based on IP-address and protocol.
For example, you can limit SSH access to just your (local) computer.
Please note that, as long as you have not edited the security profile yet, **all** inbound SSH traffic is disabled.
To edit the security group, from the AWS console go to the EC2 Dashboard.
Then select the EC2 instance and, in the lower half of the screen, click the entry to the right of *Security groups*.



Public DNS	ec2-54-93-168-77.eu-central-1.compute.amazonaws.com
Public IP	54.93.168.77
Elastic IP	-
Availability zone	eu-central-1b
Security groups	awseb-e-gaexhpdrqw-stack-AWSEBSecurityGroup-SJJ9LOF17D6C . view rules
Scheduled events	No scheduled events
AMI ID	aws-elasticbeanstalk-amzn-2014.09.0.x86_64-tomcat7java7-hvm-201410212256 (ami-2c3d0b31)
Platform	-
IAM role	aws-elasticbeanstalk-ec2-role

On the security group page click the *Inbound* tab



Security Group: sg-45a2412c

Type	Protocol	Port Range	Source
HTTP	TCP	80	sg-82a340eb (default_elb_24d6e55f-de06-3701-b80c-2f793042849e)
Custom TCP Rule	TCP	1984	213.125.52.66/32
Custom TCP Rule	TCP	1984	195.240.173.25/32
Custom TCP Rule	TCP	1984	109.71.51.119/32
Custom TCP Rule	TCP	8984	213.125.52.66/32
Custom TCP Rule	TCP	8984	195.240.173.25/32
SSH	TCP	22	195.240.173.25/32

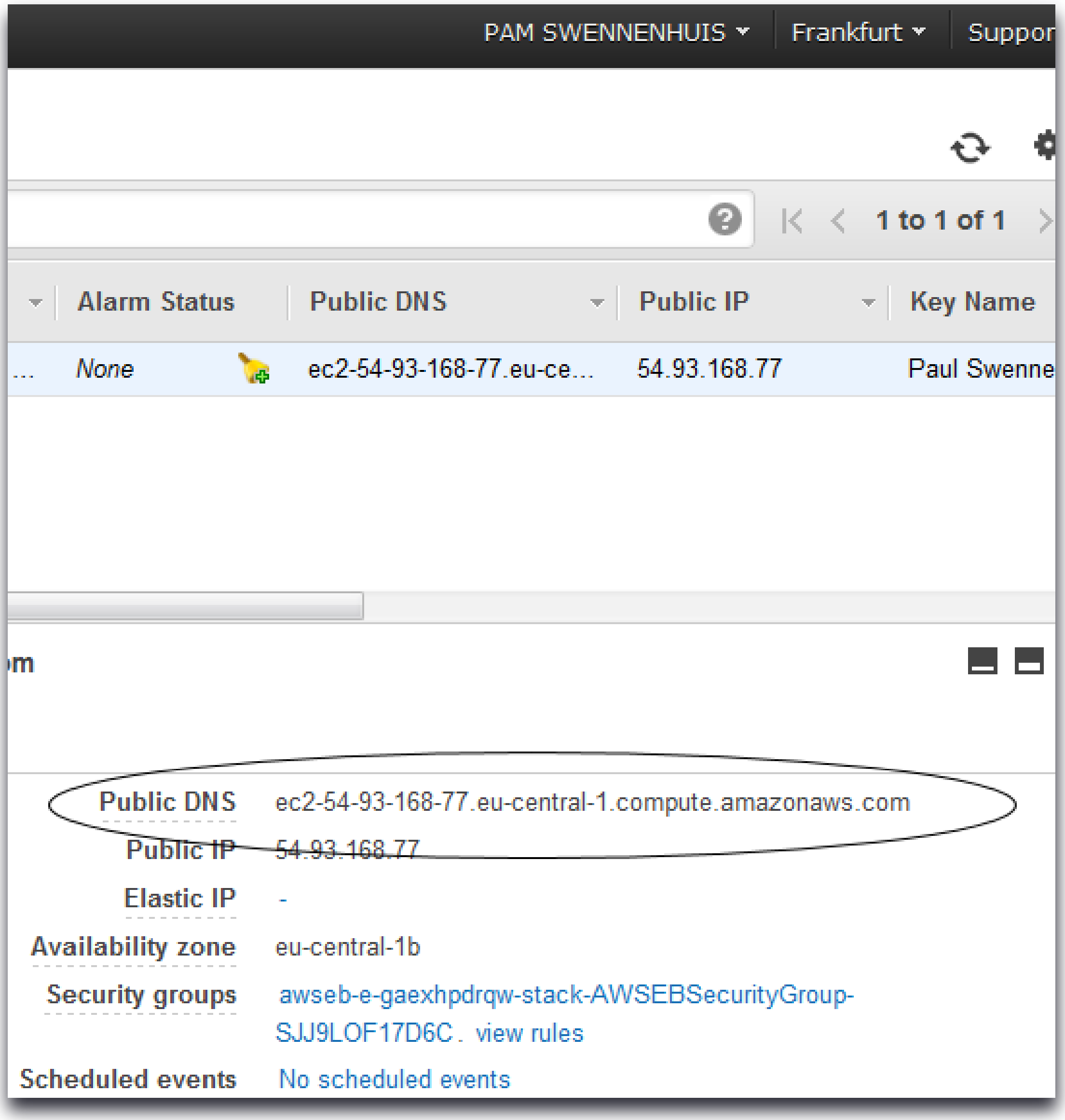
Click *Edit* to change, add or delete rules.
For SSH access you should provide a SSH rule with your IP address.
For BaseX database access from PHP scripts you should provide a *Custom TCP rule* for port 1984 and the IP address of the computer the PHP script is running on.
The edit process is quite straightforward. Just make sure not to forget the /32 or /64 postfix after the source IP. Without it the rule is invalid.

Connecting with Putty

Before you proceed: **make sure you allowed SSH access to your EC2 instance for the computer you are working on**! See the previous topic.

For SSH connections I use Putty. Of course you can select a tool of your own choice.
To connect to the EC2 instance using Putty, take the following steps:

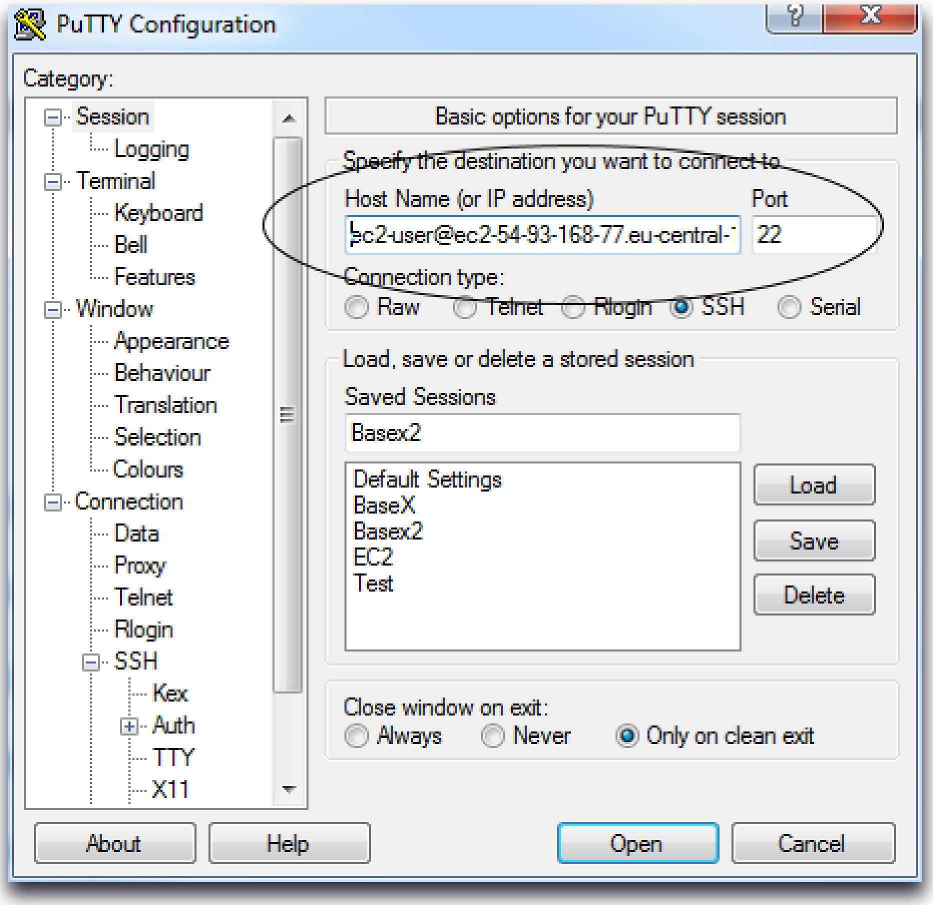
1. If not already done so, download Putty and install it onto your computer.
2. First we are going to convert the OpenSSH *.pem* key that we downloaded earlier to Putty’s own format.
   Start PuttyGen (Windows Start menu > Putty > Run PuttyGen)
3. Select File > Load private key and locate the *.pem* file you saved earlier
4. Click *Save private key* and save the Putty key (*.ppk* extension). Use the same name as the *.pem* file so you know they belong to each other.
5. You can now exit PuttyGen
6. Start Putty
7. From the EC2 instance page, copy the Public DNS string:



	Alarm Status	Public DNS	Public IP	Key Name
...	*None*	ec2-54-93-168-77.eu-ce...	54.93.168.77	Paul Swenne

Public DNS	ec2-54-93-168-77.eu-central-1.compute.amazonaws.com
Public IP	54.93.168.77
Elastic IP	-
Availability zone	eu-central-1b
Security groups	awseb-e-gaexhpdrqw-stack-AWSEBSecurityGroup-SJJ9LOF17D6C . view rules
Scheduled events	No scheduled events

8. Paste that string in Putty’s *Host name* input box, and precede it with “ec2-user@”:



Host Name (or IP address): ec2-user@ec2-54-93-168-77.eu-central-

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Saved Sessions: Basex2

Default Settings
BaseX
Basex2
EC2
Test

Close window on exit: ○ Always ○ Never ● Only on clean exit

(Note: "ec2-user" is the default user name that AWS creates for EC2 instances.)

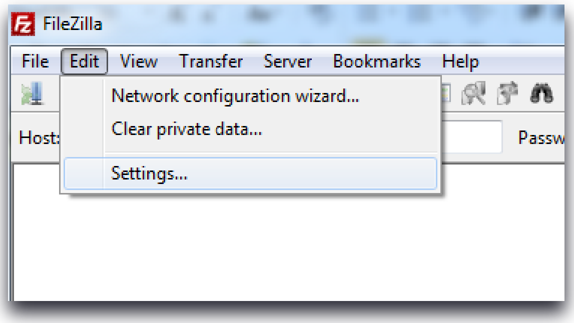
9. Save the session by entering a name in the *Saved sessions* box and pressing *Save*
10. On the left, go to *SSH > Auth*
11. Using the Browse button, locate the private *.ppk* key file that you created with PuttyGen
12. On the left, go back to *Session*, and click *Save* again
13. Now double-click the saved session entry.
    If all went well a SSH console is opened and you are connected to your EC2 instance as "ec2-user@<your EC2 instance>"
    For now, you can just exit the SSH connection.

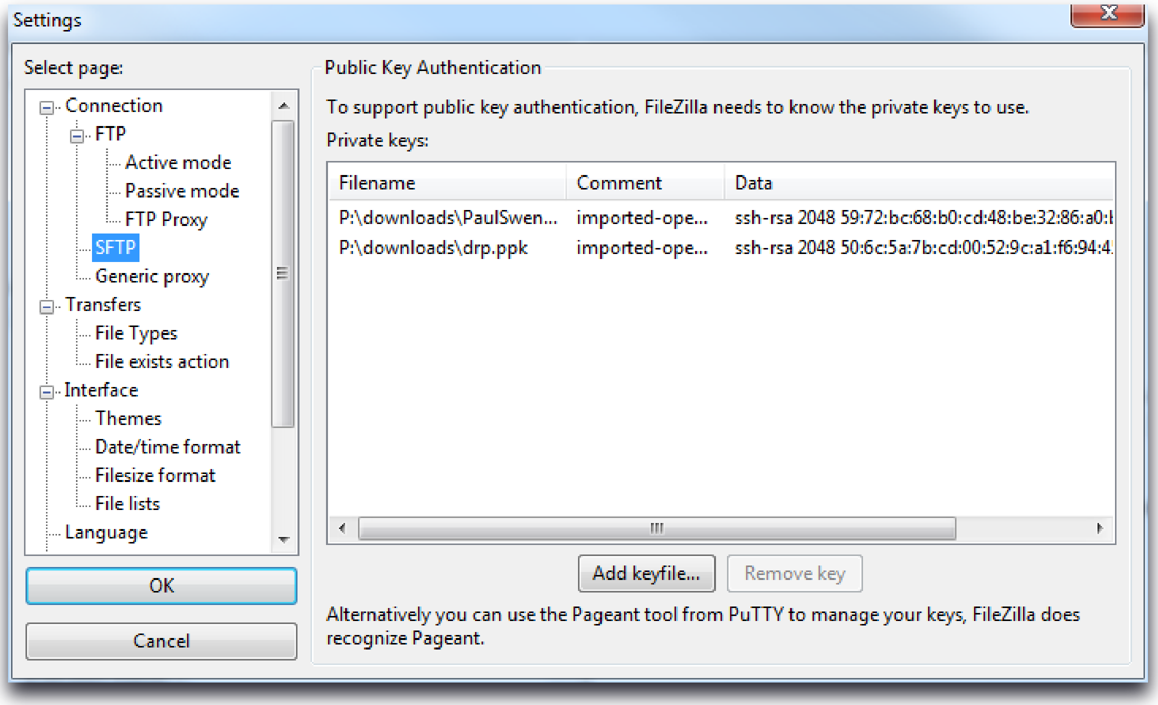
File administration with FileZilla

Next to being able to open an SSH connection to the EC2 instance, you will also want to be able to browse through the folders and files, and transfer files – for example **.xq* files.
FileZilla can do that for you.
As with SSH, there are of course others options available.

So let's see how to set up things in FileZilla:

1. If not already done so, download FileZilla and install it onto your computer.
2. Tell FileZilla what SSH key to use:
   Go to *Edit > Settings*, then *SFTP*





Public Key Authentication

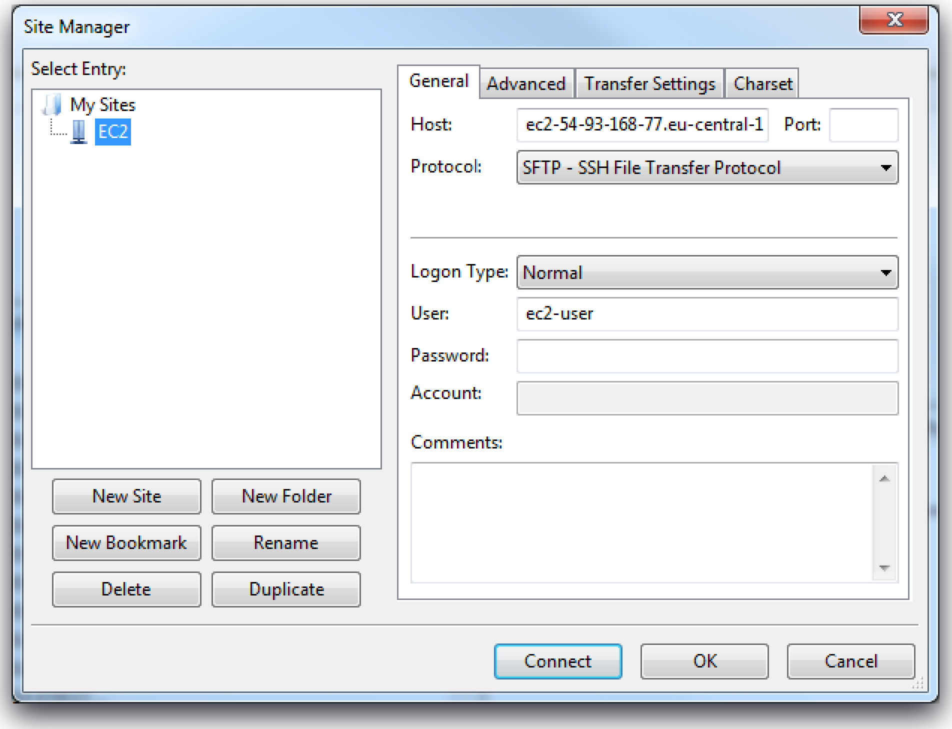
To support public key authentication, FileZilla needs to know the private keys to use.

Private keys:

Filename	Comment	Data
P:\downloads\PaulSwen...	imported-ope...	ssh-rsa 2048 59:72:bc:68:b0:cd:48:be:32:86:a0:
P:\downloads\drp.ppk	imported-ope...	ssh-rsa 2048 50:6c:5a:7b:cd:00:52:9c:a1:f6:94:4

Alternatively you can use the Pageant tool from PuTTY to manage your keys, FileZilla does recognize Pageant.

3. Click *Add key file* and point to the previously saved *.ppk* file
4. Go to *File* > *Site manager* and add a new site by clicking *New site*
5. Provide a name and press *Enter*
6. On the right, provide the Host, i.e. the Public DNS string from the EC2 page (**without** the “ec2-user@” part), enter SFTP as *Protocol*, ‘Normal’ as *Logon Type* and ‘ec2-user’ as *User*.
   Do **not** enter a password.



Site Manager

Select Entry:

My Sites
EC2

Host: ec2-54-93-168-77.eu-central-1
Port:
Protocol: SFTP - SSH File Transfer Protocol
Logon Type: Normal
User: ec2-user
Password:
Account:
Comments:

7. When done, click *Connect*.
   You should be connected to your EC2 instance and be able to view and browse the files and folders there.

Installing the BaseX WAR distribution

If everything went ok so far, we can proceed and deploy the WAR distribution of BaseX. This is a simple straightforward process.

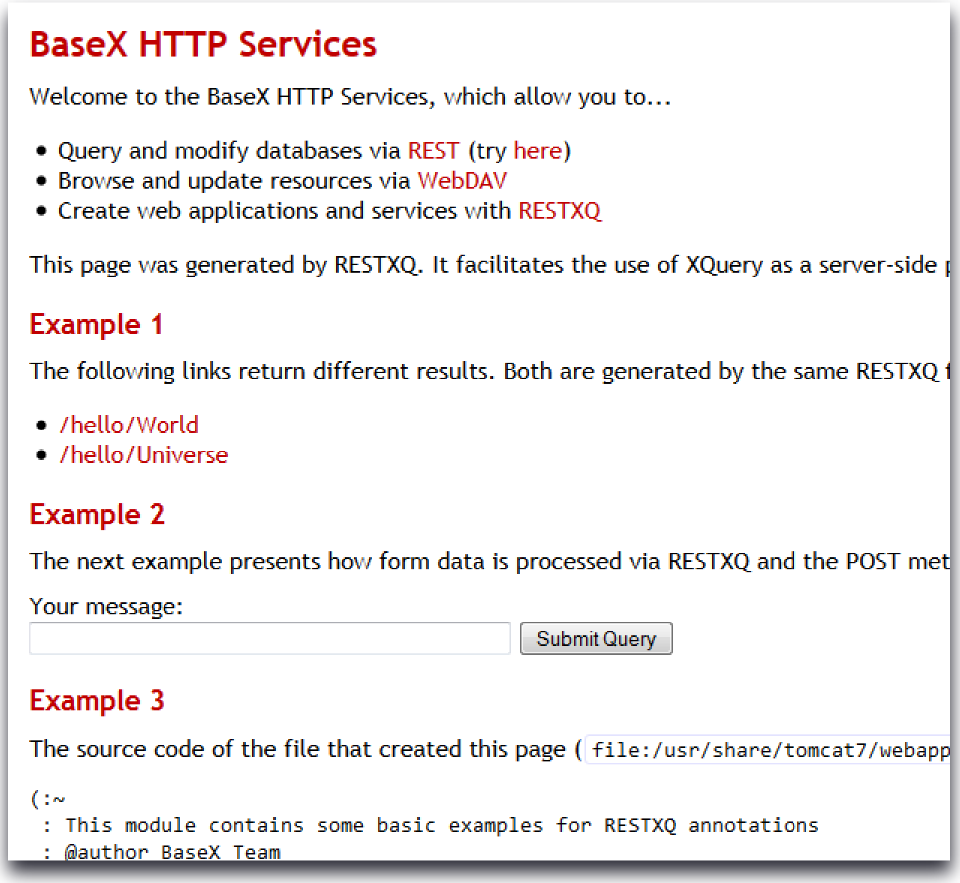
1. Download the BaseX WAR distribution from http://basex.org/products/download/all-downloads/ (BaseX 7.9.war)
2. From the AWS console got to *Elastic Beanstalk* (under *Deployment and Management*)
3. At the right of the screen click *Create New Application*

4. On the Application Information page, enter a name for the app, for example “EC2BaseX”
5. On the Environment Type page, select *Web Server* for the environment tier, select *Tomcat* for the predefined configuration and *Load balancing, autoscaling* for Environment type. Click *Next*.
6. On the Application Version page, select *Upload your own*, click *Browse*, and then select the BaseX war package. Click *Next*.
7. On the Environment Information page, enter a unique environment name and environment URL. Click *Check Availability* to ensure that your URL is available. If the URL is available, it is displayed in green; otherwise, you'll see an error message. When you have a unique URL, click *Next*.
8. On the Additional Resources page, make sure both check boxes are clear, and then click *Next*.
9. On the Configuration Details page, under *EC2 key pair* select the pair that you created earlier, in the section *Public and private key*.
   Also, under *Email address*, enter an email address that you read on a regular basis. You can leave the rest unchanged. Click *Next*.
10. On the Environment Tags page, click *Next*.
11. On the Review page, verify your settings, and then click *Launch*. On the AWS Elastic Beanstalk dashboard, you can watch in real time as Elastic Beanstalk creates an environment and deploys the app. This process takes a few minutes.
    But then…..

Test if BaseX is running

So you made it up to here? And both your EC2 instance and BaseX app are running in good health? Let us see …

In your browser, enter *<url of the BaseX application>*, for example
http://drpbasex2.elasticbeanstalk.com
You should see a page starting with “BaseX HTTP Services”:



BaseX HTTP Services

Welcome to the BaseX HTTP Services, which allow you to…

- Query and modify databases via REST (try here)
- Browse and update resources via WebDAV
- Create web applications and services with RESTXQ

This page was generated by RESTXQ. It facilitates the use of XQuery as a server-side p

Example 1

The following links return different results. Both are generated by the same RESTXQ f

- /hello/World
- /hello/Universe

Example 2

The next example presents how form data is processed via RESTXQ and the POST met

Your message:

Submit Query

Example 3

The source code of the file that created this page (`file:/usr/share/tomcat7/webapp`

```
(:~
 : This module contains some basic examples for RESTXQ annotations
 : @author BaseX Team
```

If you see this page: congratulations! You just installed BaseX on AWS! Now you can go on and configure the app in the next section.

If not: please review every step in the sections above. Were you able to SSH connect? Could you see files and folders with FileZilla? Did you enter the correct environment and configuration parameters? Did you copy the URL correctly? What folders and files does FileZilla show? You should have a *tomcat7* folder in /var/lib, and in the tomcat7 folder there should be a webapps/ROOT folder containing the BaseX files.

Where to put .xq files, libraries, databases

OK, so BaseX is running. But…. there is nothing to work with, yet.
You can test this for yourself by entering *<url of Basex app>/rest/,* for example http://drpbasex2.elasticbeanstalk.com/rest/
This will show the available databases, and at this moment there is none yet. So how do we get the databases at the right location? And where to put the query files *(*.xq*) and libraries *(*.xqy*)?
BaseX, by default, will search for databases in the *data* folder. The data folder is located in */var/lib/tomcat7/webapps/ROOT*
However, before you try to transfer your local databases to that *data* folder, please note that

a) you must have write permission on the *data* folder. After deploying BaseX the default user (ec2-user) will not have write permission on the *data* folder so you have to set it.
b) the databases must be owned by *tomcat*, otherwise BaseX will not be able to read them.

To set writing permissions for ec2-user on the *data* folder, proceed as follows:

1. SSH connect to the EC2 instance
   Use Putty to open a connection to the EC2 instance where you deployed BaseX
2. Switch user
   Make ec2-user a superuser by typing *sudo su*
3. Make the data folder writable
   a. cd to */var/lib/tomcat7/webapps/ROOT*
   b. type *chmod 777 data*
4. Transfer your databases to the data folder
   Use FileZilla to transfer the databases on your local computer to the data folder
5. Reset the rights on data to 644 by typing *chmod 644 data* in Putty

To make tomcat the owner of the databases, proceed as follows:

1. SSH connect to the EC2 instance
   Use Putty to open a connection to the EC2 instance where you deployed BaseX
2. Switch user
   Make ec2-user a superuser by typing 'sudo su'
3. Make tomcat owner
   a. cd to /var/lib/tomcat7/webapps/ROOT/data
   b. type *chown -R tomcat:tomcat **
      This will recursively set *tomcat* as owner of all databases in *data*

Now try the URL *<url of Basex app>/rest/* again. It should now list all the available databases.

Queries

For he query files *(*.xq*) you can create a folder just below the ROOT.
I created an *xq* folder and put all query files there.
Set the rights on this folder to 644. The folder does not have to be owned by tomcat.

Test the queries by issuing a command like this:
http://drpbasex2.elasticbeanstalk.com/rest/?run=xq/allresults.xq
Substitute the URL part with the URL of your domain. And use the appropriate folder and query name.

Libraries

By default, BaseX searches for library files (imported modules) in this folder:
usr/share/tomcat7
If you are using imported modules in your XQueries make sure to place them in that folder.

Users and rights

You may have wondered – I did! – why you did not have to enter credentials, in the form of basic authentication, when you tried to access *<your EC2 instance url>/rest* (e.g. http://drpbasex2.elasticbeanstalk.com/rest .
That is because, after installation, there is no password for the admin user.
That is a situation we better avoid, so let us do two things now:

a) set a password for the admin user
b) create a custom BaseX user along with a password

We will use the custom user for the PHP scripts.

Database commands with the REST API

We can use the REST API – in the browser – to do some basic administration.
For example, we can list the known users by entering *…./rest/?command=show users*

Create user

To create a user, first use some md5 hash tool like http://md5-hash-online.waraxe.us/ to get the md5 version of a password you are going to use. Then enter
…./rest/?command = create user <username> <md5hash>
For example *…./rest/command=create user johndoe 123456789abcdef123456789abcdef*

Grant rights

To give this user admin rights enter
…./rest/command=grant admin to johndoe

Change password

To change the password of user admin
…./rest/command=password <newpassword md5 hash>
For the last command it is assumed that you are logged in as admin. By default this is the case. That is, if you did not to enter a username/password combination for one of the commands above, you were logged in as admin.
After changing the password for user admin you **will** have to login.
Test this by restarting your browser and then enter *<url of the BaseX application>/rest/*

Connecting via BaseX client

Do you remember that, in part 1 of this tutorial, we wrote a PHP script to query a BaseX database and process the results?
We can re-use that script now. Just make a copy of it and then edit it to accommodate for the new situation: the EC2 instance running the BaseX application.
All you have to do is change the *URL*, the *port*, the *username* and the *password* in the script.
The port, by default, is 1984. The rest of the parameters have been set by yourself.
Make sure though that you added the IP address of the server where the PHP script is running as a Custom TCP rule of the Inbound tab in the security group of the instance.

Test the script.

That's it!

That concludes this BaseX introduction. Our goal was to have a working BaseX application so that we can create, manage and query XML databases, from anywhere on this planet.
And that's exactly what we have achieved..

I hope this introduction to BaseX was useful for you, and wish you happy coding.
If you encounter any error or unclarity in this introduction please let me know at paul@swennenhuis.nl