AdWhirl Open Source Server
Setup Instructions

AdWhirl Server Setup Instructions

The server runs in Amazon's web cloud. To set up the server, you need an Amazon Web Services (AWS) account and the AMI (Amazon Machine Image) for the AdWhirl server, which contains all the software, including the operating system and associated configuration settings, applications, libraries, and so on.

Setting up the AdWhirl Server

1. To set up the environment:
    a. You must have a version 1.5.0 compatible Java Runtime installation, and the `JAVA_HOME` environment variable must be correctly set.
    b. You must have an active Amazon Web Services (AWS) account, and have signed up for both Amazon S3 and Amazon EC2.
    c. You must create a directory called `.ec2` in your home directory, which contains your X.509 certificate and private key.
    d. The `EC2_HOME` environment variable must be correctly set.
    e. The `EC2_CERT` and `EC2_PRIVATE_KEY` environment variables must be correctly set.
    f. You must download and unzip the Amazon EC2 command line tools. These tools are written in Java and include shell scripts for Windows 2000/XP and Linux/Unix/Mac OSX.

    See Amazon's Getting Started Guide for more information at: http://docs.amazonwebservices.com/AmazonEC2/gsg/2006-10-01/.

2. Get the AMI ID of the latest version of the server software from the Google Code project site at http://code.google.com/p/adwhirl.

3. You will be running an instance of the public AMI for the server. Because it has no password, you need a public/private key pair to log in to the instance. To generate a key pair to log in to the AMI instance, enter the following, where *`adwhirl-keypair`* represents the key pair you are creating:

    ```
    ec2-add-keypair adwhirl-keypair
    ```

4. Save the private key that that is generated to a local file, including the `--BEGIN--` and `--END--` lines.

5. Launch an instance of the currently running AMI by substituting the AMI ID in the following command:

    ```
    ec2-run-instances ami-XXXXXXXX -k adwhirl-keypair
    ```

    The output looks like this:

    ```
    INSTANCE   i-YYYYYYYY  ami-XXXXXXXX   pending   adwhirl-keypair  0
    ```

The instance ID is the second field in the output (`i-YYYYYYYY`).

Note: After you do this, Amazon starts billing you for the computing and network resources that the instance consumes.

It takes a few minutes for the instance to launch. When the instance state in the field just before the key pair name reads "running," the instance has started booting. It may still be a short time before it is accessible over the network, however.

6. To check the status of your instance, enter your instance name in this command:

   ```
   ec2-describe-instances i-YYYYYYYY
   ```

   This command also gives you the DNS name of the instance, which we'll refer to as *domU-AA-AA-AA-AA-AA-AA.ZZZZ.compute.amazonaws.com.*

7. To authorize SSH and HTTP traffic for the instance, enter:

   ```
   ec2-authorize default -p 22
   ec2-authorize default -p 80
   ```

   The first command authorizes network access to instances in your default group on the standard SSH port (22), and the second command opens up the standard HTTP port (80).

8. To control the instance, you must log in as the root user:

   ```
   ssh -i keyfile
      root@domU-AA-AA-AA-AA-AA-AA.ZZZZ.compute.amazonaws.com
   ```

   After you are logged in, you can edit the AdWhirl AMI if you so desire.

   Note: It is highly suggested that you run the GNU Screen and work inside of it.

9. The bundle supplied with the SDK has been tested, but you can still update it, if desired, from the SVN (subversion) branch:

   ```
   svn up adwhirl
   ```

10. If you update the bundle, you must edit the utility class to include your AWS keys and any specific configuration you desire. Emacs, vim, nano, ed, and so on are all installed on the instance; feel free to use your editor of choice. For example:

    ```
    emacs adwhirl/src/util/AdWhirlUtil.java
    ```

11. Build the distribution.

    ```
    ant dist
    ```

12. Now you have a jar that you can use to start the services.

    ```
    java -jar adwhirl/dist/adwhirl.jar
    ```

    You're up and running!

Re-Bundling Your Instance

If you want to save any changes to the current instance, you must re-bundle it. To re-bundle your instance, follow these steps.

1. You must copy your private key and X.509 certificate to the instance from your local machine:

   ```
   scp -i keyfile pk-XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.pem
   cert-YYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYY.pem
   root@domU-AA-AA-AA-AA-AA-AA.ZZZZ.compute.amazonaws.com
   ```

2. After you make your changes, a system snapshot must be created and packaged into an AMI. Replace *ZZZZZZZZZZZZ* with your AWS Account ID (not your AWS key).

   ```
   ec2-bundle-vol -d /mnt -k ~root/ pk-
   XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.pem
   -c cert-YYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYY.pem -u ZZZZZZZZZZZZ
   ```

   This command may take several minutes to complete.

3. Confirm that you now have an AMI, its associated manifest file, and image parts:

   ```
   ls -l /mnt/image.*
   ```

4. Because all AMIs are loaded from Amazon S3 storage, you must upload your newly bundled AMI to your account on Amazon S3. S3 stores data objects in buckets, which are similar to directories. You must specify a bucket name in the command below for **<***your-s3-bucket***>**.

   ```
   ec2-upload-bundle -b <your-s3-bucket> -m /mnt/image.manifest.xml -a
   <aws-access-key-id> -s <aws-secret-access-key>
   ```

5. In order for Amazon EC2 to locate your AMI, you must register it:

   ```
   ec2-register <your-s3-bucket>/image.manifest.xml
   ```

   Now you can create instances of your new AMI.

Terminating Your Instance

At some point, you will want to terminate your instance and stop being billed. You can terminate it using this command:

```
ec2-terminate-instances i-YYYYYYYY
```

Alternatively, if you are already logged in to the AMI instance, you can shut it down with your ssh tool:

```
shutdown -h now
```

Using Elastic Load Balancing

Elastic Load Balancing is a web service that distributes application loads between two or more EC2 instances. To use the Amazon Elastic Load Balancing API Tools, follow these instructions.

1. Get the Elastic Load Balancing API Tools from: http://developer.amazonwebservices.com/connect/entry.jspa?externalID=2536&categoryID=88.

You can find documentation on the syntax of the Elastic Load Balancing tools at: http://docs.amazonwebservices.com/ElasticLoadBalancing/2009-05-15/DeveloperGuide/

2. Create a new load balancer. Make sure your availability zones correspond with the zones your instances are in.

```
elb-create-lb  AdWhirlLB --headers --listener "lb-port=80,
instance-port=80,protocol=HTTP" --availability-zones us-east-1a
```

This returns the DNS name of your load balancer. You'll want to add this as a target to a CNAME entry in your DNS record.

3. To make sure that your instances are up and running correctly, add a health checker:

```
elb-configure-healthcheck AdWhirlLB  --headers --target
"HTTP:8080/ping" --interval 30 --timeout 3 --unhealthy-threshold 2 --
healthy-threshold 2
```

4. Register your instances with the load balancer:

```
elb-register-instances-with-lb  AdWhirlLB  --headers --instances
i-XXXXXXXX,i-YYYYYYYY
```

Traffic should now be automatically split between your running instances.