PUSH TECHNOLOGY

The Diffusion™ Amazon EC2 Starter Guide

PUSH TECHNOLOGY LIMITED
3RD FLOOR HOLLAND HOUSE
1-4 BURY STREET LONDON EC3A 5AW
Telephone: +44 (0)203 588 0900

340 MADISON AVE, NY, NY 10173
Telephone: +1 201-978-5574

Twitter: @push_technology
www.pushtechnology.com

Contents

1. Diffusion Starter Guide ... 3
2. Who is Push Technology? ... 3
3. What is Diffusion™? ... 3
4. What does it all mean? ... 4
4.1. What shall I Build? ... 5
5. Getting Started ... 6
6. Creating a Diffusion "Cloud" Instance ... 6
6.1. Finding Diffusion AMI (Amazon Machine Image) ... 6
6.2. Creating an Instance ... 7
6.3. Launching the Instance ... 11
6.4. Assign an IP Address ... 13
6.5. Confirm Successful Launch ... 15
7. Build your Diffusion application ... 15
7.1. Download Remote Control ... 15
7.2. Create a standard Java project ... 16
7.3. Reference the Diffusion Remote jar file ... 19
7.4. Build a Diffusion Remote Service Listener ... 20
7.5. Build a Data Source Provider ... 20
7.6. Build the Diffusion Publisher ... 21
7.7. Build a Java Executable File ... 21
7.8. Start the Diffusion Remote Publisher ... 24
7.9. Subscribing to Data ... 26
8. Conclusion ... 28
9. Support ... 29
Appendix 1 - Further Information ... 29
Appendix 2 - AMI Definition Table ... 30

1. Diffusion Starter Guide

This starter guide will provide you with all the development knowledge required to get a Diffusion application up and running using Diffusion within Amazon EC2. Diffusion is a data distribution platform and once you have been through this guide, you will have a data distribution application that can accept a data feed locally, publish messages to an instance of Diffusion in the cloud and then publish the data to any clients that are connected and have shown an interest in specific data. This guide is designed to provide you with an insight into the power of Diffusion and give you an understanding on how Diffusion can be integrated with the Amazon EC2 cloud offering.
You will need to be familiar with the Eclipse development environment and understand the concept around code compilation and deployment of Java applications.

There will be no cost charged to you for using Diffusion in the cloud. However there may still be a charge to your Amazon account based on the pricing schedule for usage of the Amazon EC2 provision.

2. Who is Push Technology?

Founded in 2006 and headquartered in London, Push Technology Limited is an innovative technology and solutions specialist that focuses on pushing data beyond the edge. We are passionate about technology and its application to business, and critically, how it can be cleverly applied, at minimum impact, to realise massive gains in business performance.

We are not interested in delivering what everyone else can deliver. We are changing the conversation and concept of what is possible, and we have invested hundreds of man years developing what is an exciting, cutting-edge solution able to deliver bi-directional, dynamic and relevant data faster, more securely, more efficiently, and at a lower cost than ever before.

Uniquely capable of delivering Data on Demand, Push Technology's robust and scalable communication platform – Diffusion™ – enables large-scale smart data distribution that unlocks the true potential of today's multi-channel, connected world to help organisations realise meaningful, valuable and personalised exchanges.

3. What is Diffusion™?

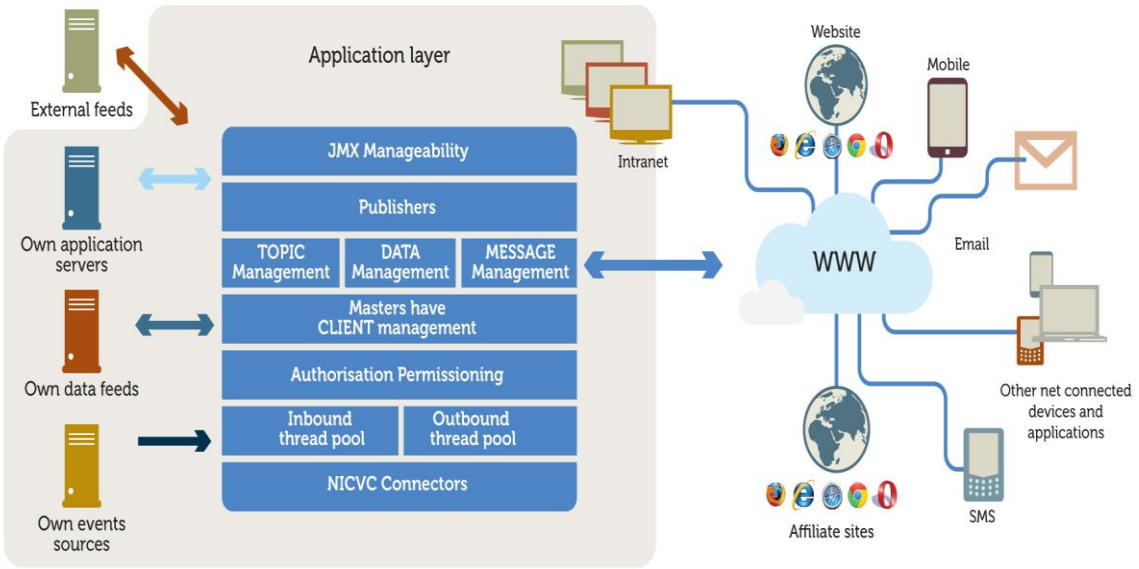
Diffusion is Push Technology's leading enterprise messaging and communications platform that is both scalable and suitable for rapid deployment, providing exceptional performance over the Internet. Diffusion enables organisations to gain critical momentum in their markets while realising immediate ROI on their simplified delivery architecture. Diffusion streams volatile data efficiently and with minimal overhead across client networks, conserving bandwidth while increasing application performance with near-zero latency.

With outstanding flexibility, Diffusion will multiplex data streams and push them out on multiple protocols to a range of different Internet-facing clients ensuring the largest available marketplace. Diffusion is the smart performance choice in a growing number of data-intensive market sectors, delivering the following functional advantages:

- Real-time / volatile data dissemination over the Internet.
- Gathering, collating and publishing data from multiple sources before distributing in a single format over the net.
- Real-time, dynamic media applications. Online applications that operate beyond traditional Web browsers.
- Highly scalable low-latency applications where enormous amounts of data move back and forth.
- Delivery of the right information to the right person at the most critical time is paramount - no matter what the end-device.

4. What does it all mean?

Diffusion is primarily an intelligent data distribution engine. Some of the main components are introduced in the diagram below:-

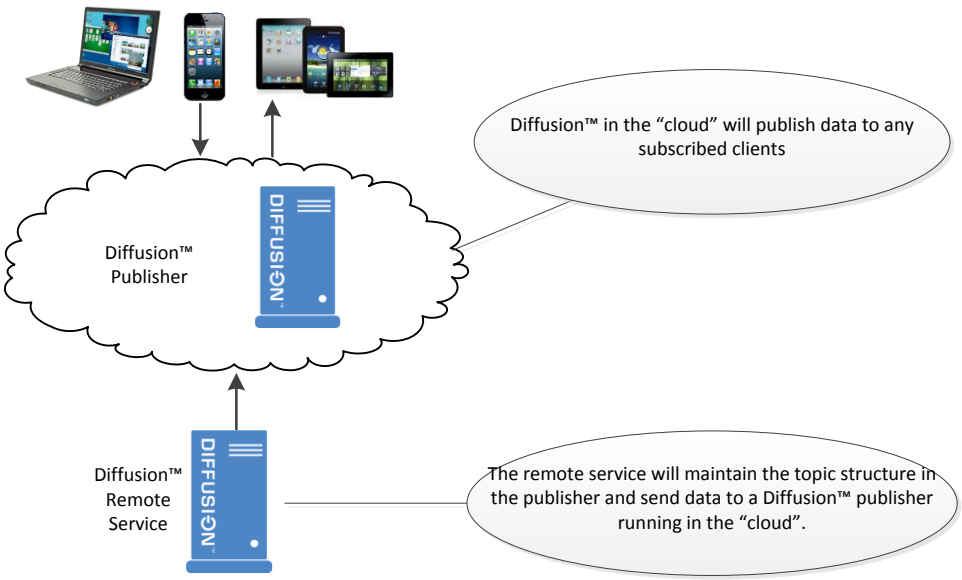


Many of the components shown are optional. Only a publisher and connected clients are essential for a working Diffusion solution. Clients, both internal and external can connect to a Diffusion server. Once they have made a connection, they have the opportunity to subscribe, by way of a client API, to any source of information they are interested in.

The server, once it has received the subscription request, will then publish the data that the client has shown an interest in by using a Diffusion publisher. The publisher will publish messages to all the clients that are interested in that information. The communication is bi-directional and thus clients may also send messages to publishers.

4.1. What shall I Build?

The diagram below provides a graphical representation of the application you are going to build. Push Technology will provide you with an instance of Diffusion that can be used to create your own image in the cloud. This cloud image will provide you with a standard implementation of a Diffusion publisher which enables to you build and execute your own Diffusion application. The Remote Service will provide the data source that feeds into the Diffusion instance in the cloud and the instance in the cloud will publish all the messages to any connected clients that have subscribed to specific topics.



5. Getting Started

This section will provide a detailed description on how to build and run your own Diffusion application. It will guide you through building your application, connecting to the image of Diffusion in the "cloud" and finally running an instance of Diffusion locally to publish messages to the Diffusion instance in the cloud.

Before you get started you need to know what resources are available to you. The reference materials that can be utilised during this development process include:

- Online API documentation that is located via the following URL:

  http://docs.pushtechnology.com/

  These documents include the complete Diffusion developer's manual and API documentation for all the client libraries.

- Sample source code that will provide guidance on how to build a Diffusion application including the publisher, remote control and the visual GUI. The sample code can be found via the following URL:

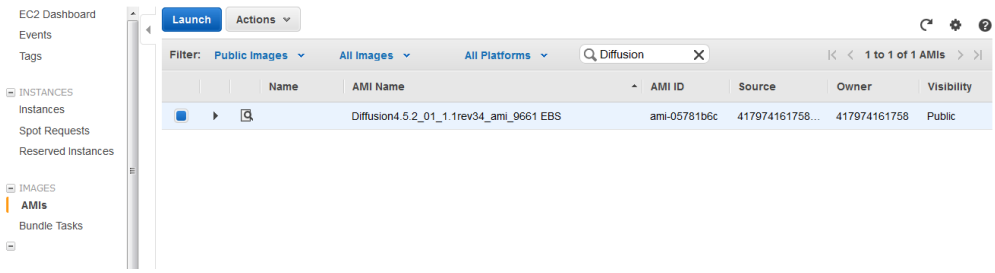
  https://github.com/pushtechnology/cloud

6. Creating a Diffusion "Cloud" Instance

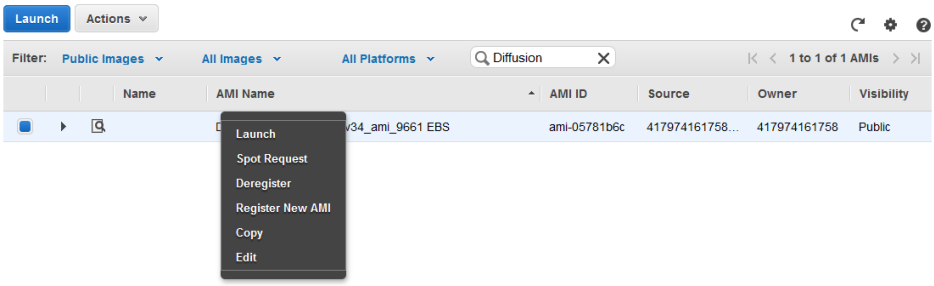
This section will provide guidance on how to create and maintain an instance of Diffusion in the cloud. There are a number of steps you will need to complete to create the instance and this advice is provided on the assumption you already have an Amazon EC2 account.

6.1. Finding Diffusion AMI (Amazon Machine Image)

Sign into your Amazon EC2 account and navigate to the management console. On the left hand panel of the console, select the Images option and open the tab for AMI's. The main window will provide you with an option to search for the public Diffusion AMI;

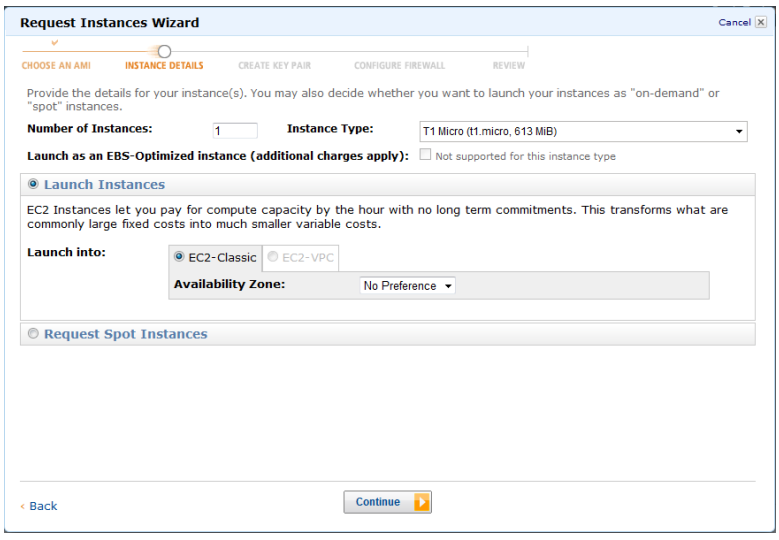


By selecting the filter for Public Images and entering Diffusion in the search field you will be presented with the screen shown above. If you are unable to locate the Diffusion AMI there is a definition table in Appendix B that you can use to identify the exact AMI you will be using for your specific region. The next step is to launch the Image creation wizard to create and launch your own Diffusion instance. Select the Diffusion AMI displayed in the main window and right click to select the Launch option.

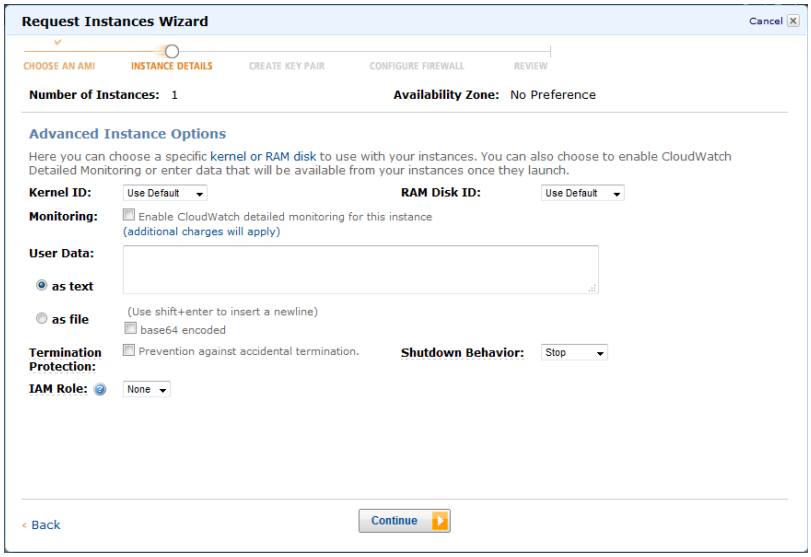


6.2. Creating an Instance

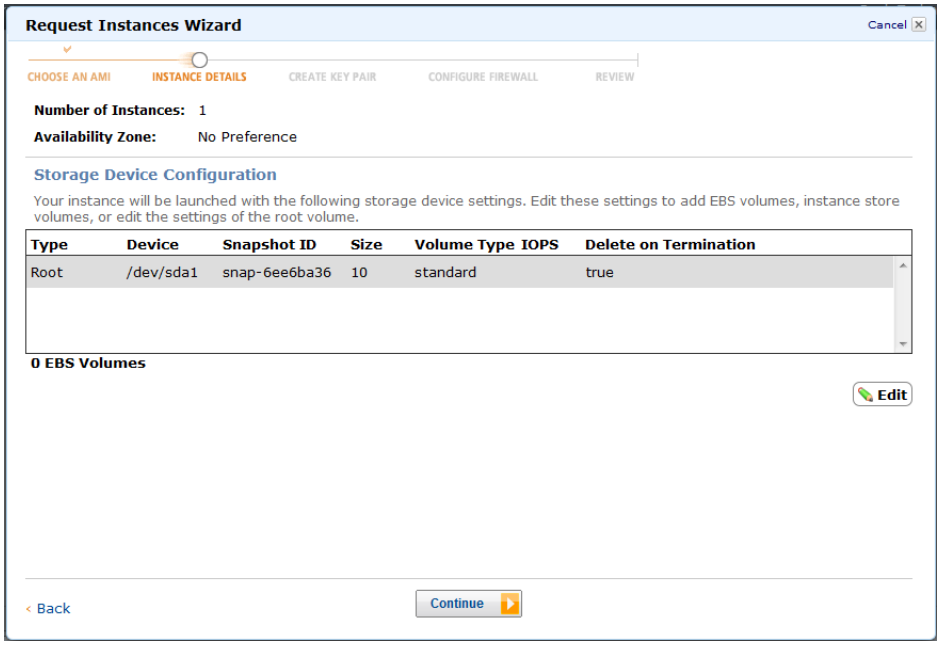
In the first window provided you will need to specify the number of instances, the instance type and what zone you want to launch the instance into. There are a number of options available to you for the Instance type and the Launch zone. Please refer to the appendix to understand what instance type suits your application and the different zones available. You do not need to specify a zone unless you know that you want your instance of Diffusion to operate within a specific zone. The initial recommendation would be to specify the T1 micro instance type and accept the default No Preference for the specific Zone. Click Continue to navigate to the next screen.



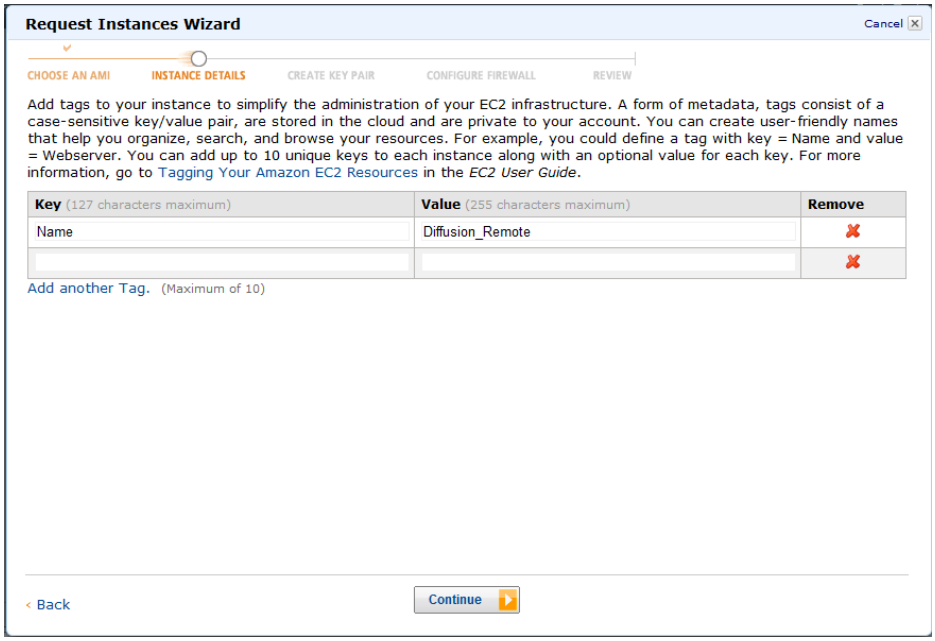
The next screen presented to you allows you to specify more specific advanced variable for the instance. Unless you require any advanced setting this step can be skipped with all default values being accepted.



The next section of the wizard will provide you with the ability to add additional storage devices. Click Continue to navigate to the next screen.



The next screen provided to you will allow you to enter a name for the instance of Diffusion you are launching. In this example I have called the instance Diffusion_Remote. Click Continue to navigate to the next screen.

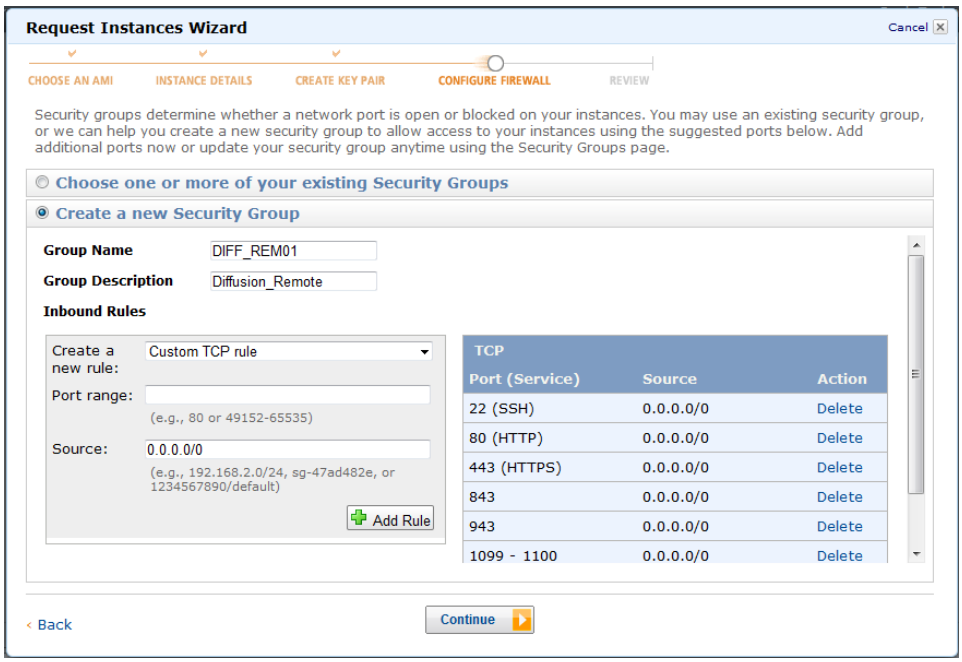


The next section of the wizard allows you to create a private / public key pair that will allow you to securely connect to the instance after it launches. This file provides a secure mechanism to allow SSH connections to the instance of Diffusion running in the cloud. Select the option to create a new key pair, enter a key pair name and then click on the “Create and Download your Key Pair” box. You will need to download the file that is created to a folder on the local machine that you will be able to reference at a later date, when using SSH to connect externally to the instance. Once the file has been saved the wizard will continue to the next option. If you do not need to provide external SSH connections to the instance then you can proceed without the key pair defined.

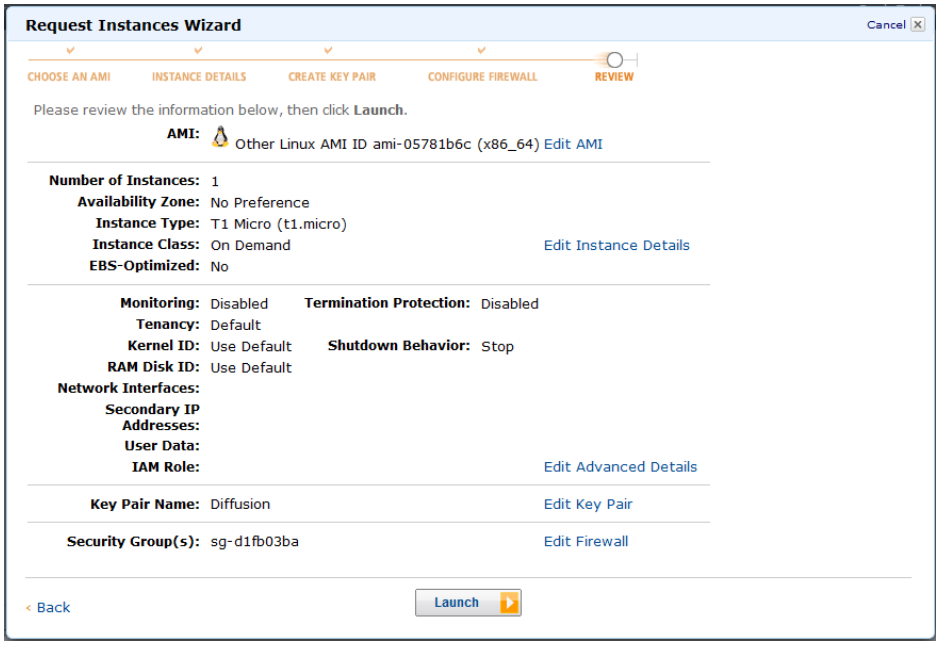
The final section of the wizard is to create the firewall rules for the instance you are going to operate. This window will ask you to define a new security group and you will need to specify the ports you are going to make available to the instance. This will help to help what communication protocols you will allow to communicate with Diffusion. As a standard the recommendation for port definitions can be found below:

Port Identifier	Connection Protocol
22	SSH
80	HTTP
443	HTTPS
843	Custom TCP
943	Custom TCP
1099 – 1100	Custom TCP
3389	RDP
5901	Custom TCP

In the example below the Group name is DIFF_REM01 and the description is Diffusion_Remote. Click Continue to navigate to the next screen.



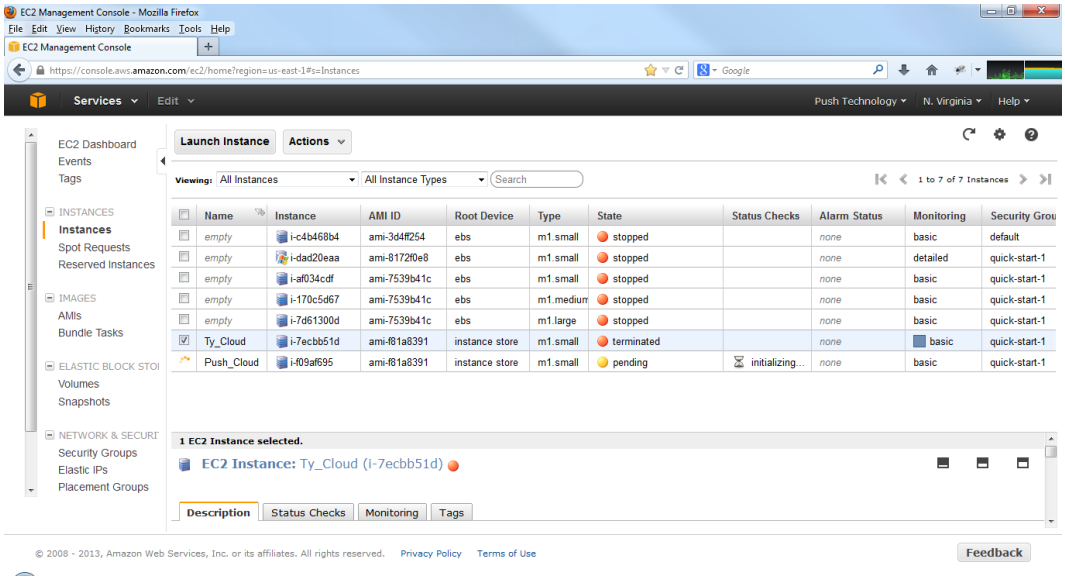
The final window provided to you is a confirmation window that allows you to check the configuration of the instance you are about to launch. By clicking on the Launch button you will be starting an instance of Diffusion running in the cloud.



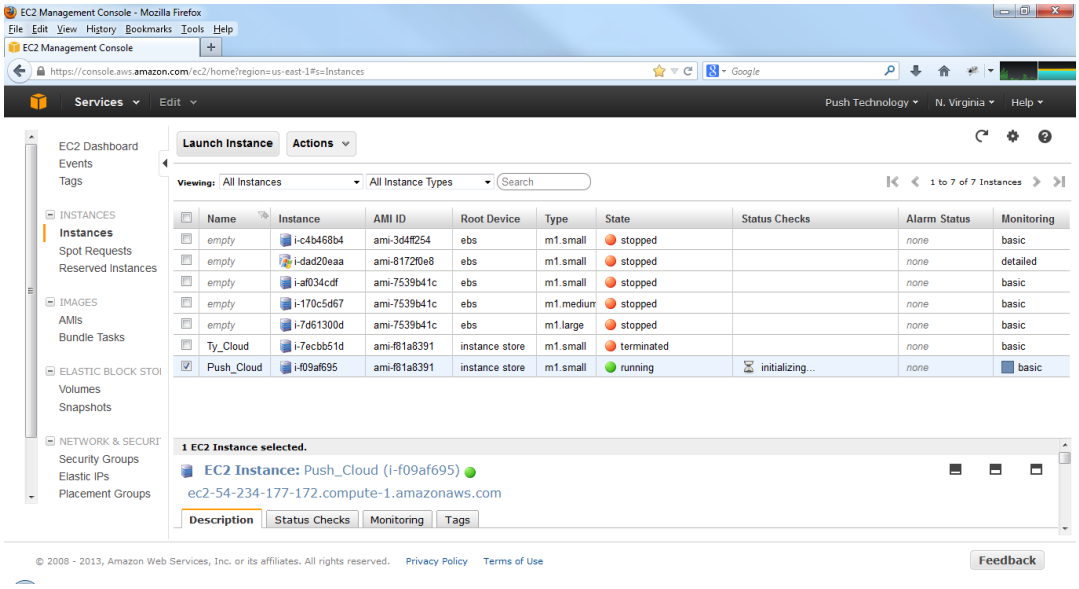
6.3. Launching the Instance

When the instance of Diffusion is launched there are a number of stages the launch process completes. Once the application is running, the final step of the launch process is to assign an IP address to the instance so that the web clients can connect to the Diffusion instance.

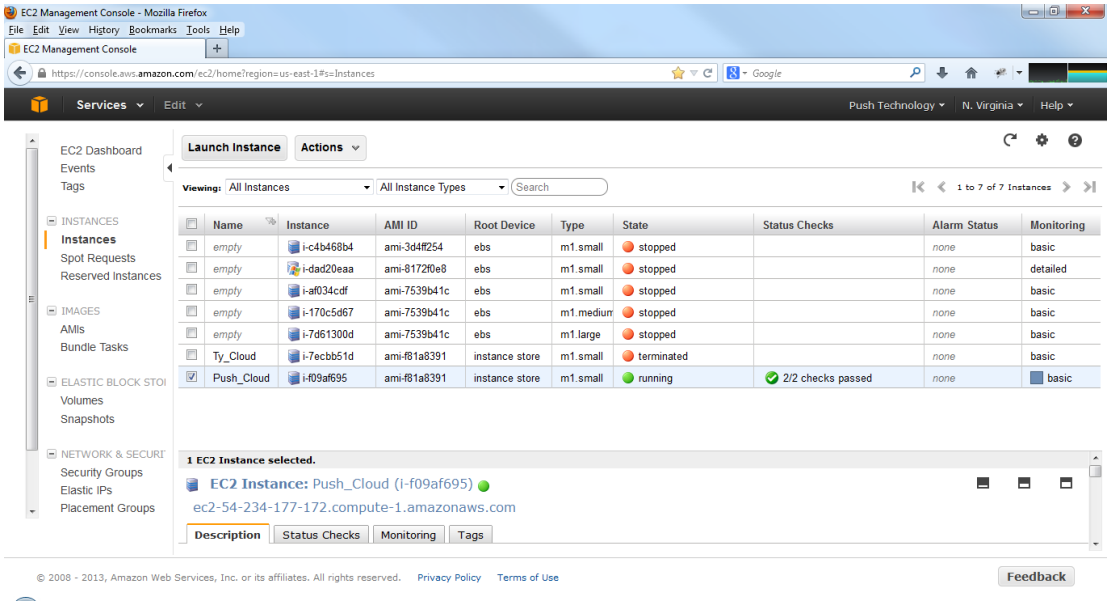
a) The first state will be **pending** with a status of **initialising**:



b) The second state will be **running** with a status of **initialising:**

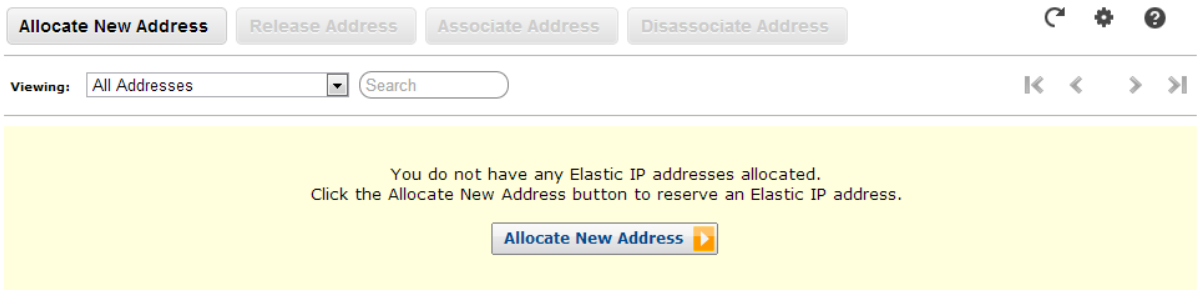


c) The final stage will display a state of **running** and a status of **2/2 checks passed**

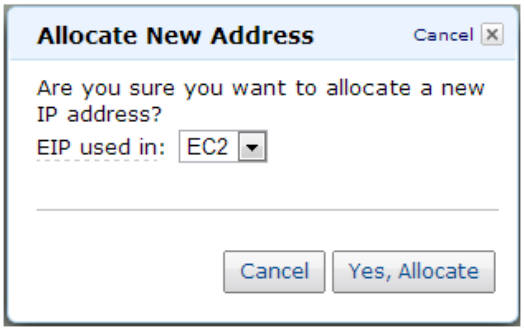


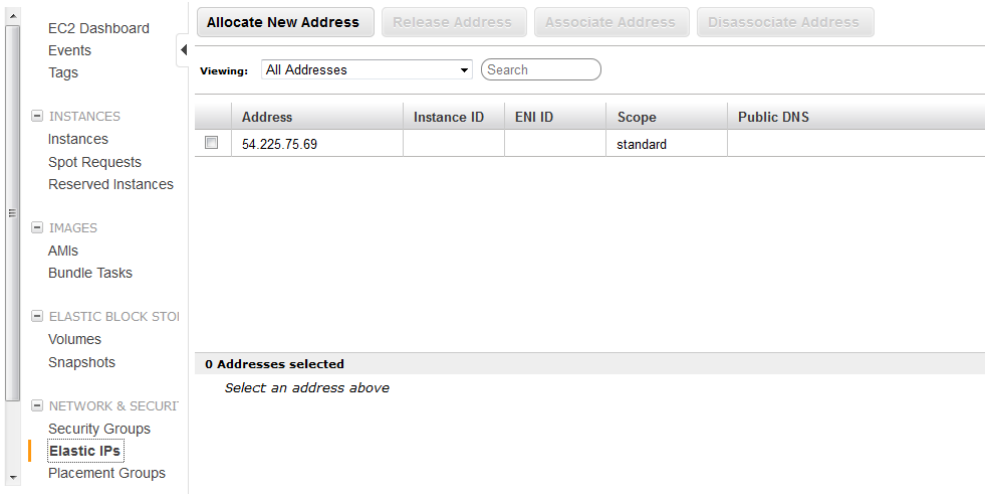
6.4. Assign an IP Address

When assigning an IP address to an existing instance of Diffusion you will need to navigate to the management console and select the Elastic IPs option under the menu heading Network and Security. When the main panel is displayed, there will either be a number of IP addresses that you can associate with the Diffusion instance you have created or you might need to create a new IP address. If you need to create a new IP address then there will be an option displayed to select and allocate new addresses:

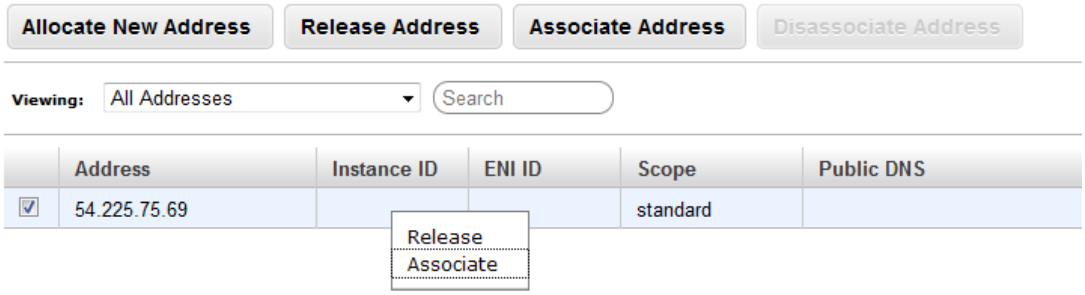


The next screen will ask you to confirm that you want to allocate new IP addresses, simply click on the Yes, Allocate button. Once the IP address has been allocated it will be displayed in the original main panel for Elastic IP's.

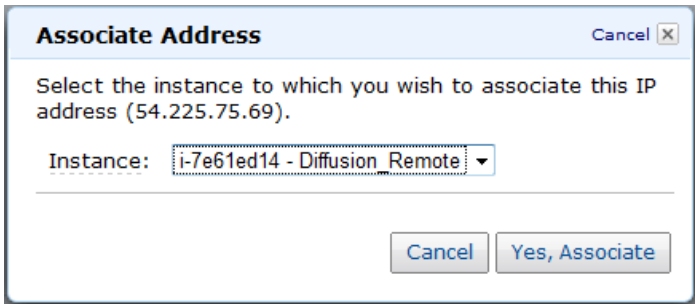




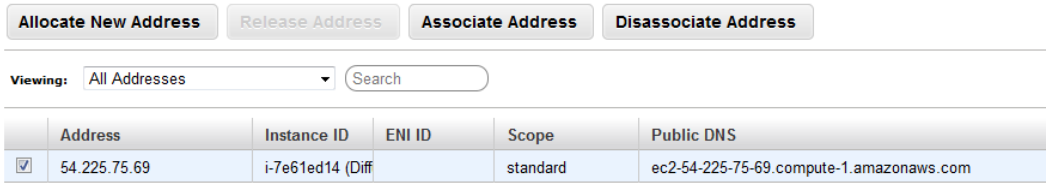
Select the IP address you would like to use by ticking the left hand box and once selected navigate to the address and right click. There will be two options available to you, Release and Associate.



Select Associate and another window is provided, where you can associate the IP address with an instance that is currently running.

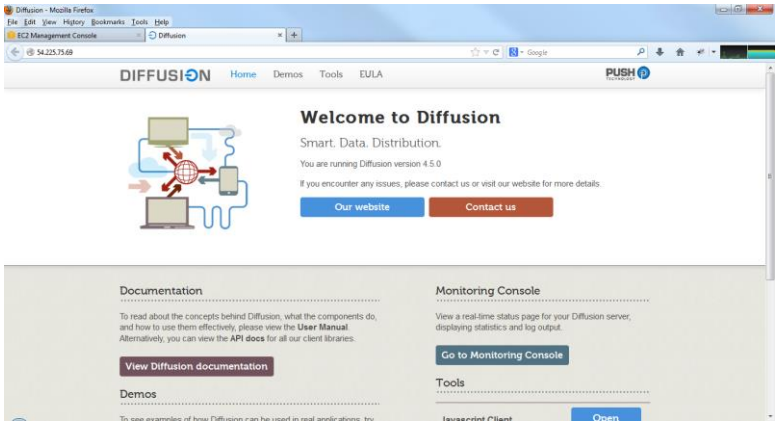


When you click the Yes, Associate button Amazon will associate the IP address with the instance you have specified and the IP Address main screen will be updated with the new association.



6.5. Confirm Successful Launch

To confirm that the Diffusion instance is up and running, try to access the URL that your instance is running on (in this example there is an elastic IP address assigned to this instance) and you will see the Diffusion landing page. The URL that is used to display the Diffusion landing page is **<<ipaddress>>**.



7. Build your Diffusion application

This section will provide you with a step by step guide into how to create your own Diffusion application. There are a number of pre-requisites before you start:

- You should have at least 6 months Java development experience.
- You should have a basic understanding of Object Oriented Programming.
- You should be familiar with the Eclipse Development environment.

Now you are ready to begin:

7.1. Download Remote Control

To start using Diffusion you will need to download the jar file the enables remote control within Diffusion. In order to do this you will need to navigate o the URL below and download the **diffusionremote.jar** file to the download directory locally:

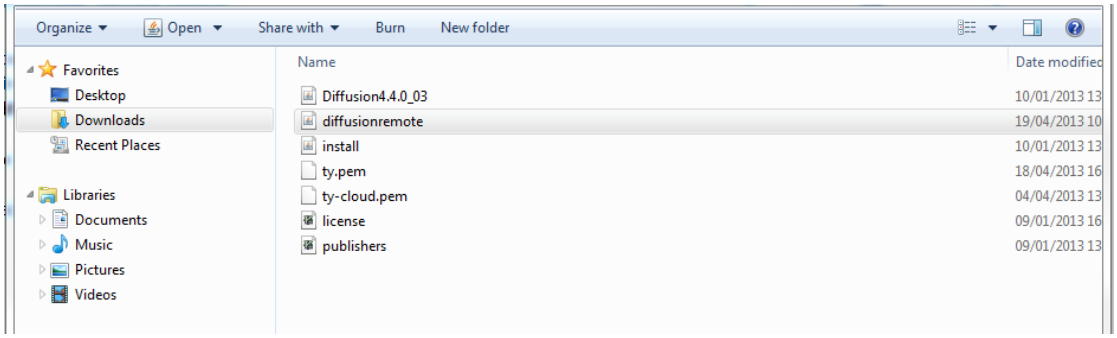
These are the steps you need to complete:

a) Navigate to the following URL and click Save:

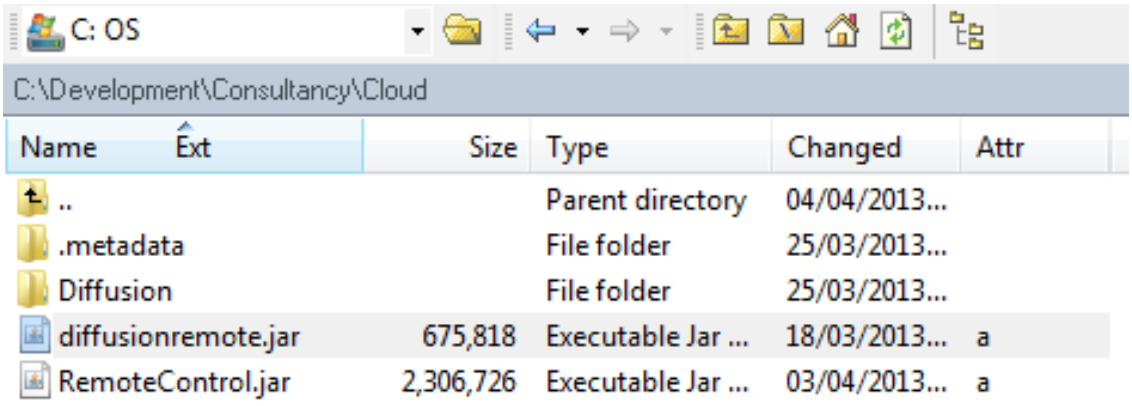
http://download.pushtechnology.com/builds/4.5.2/diffusionremote.jar

There will be an option provided to you to save the jar file, locally, to your default download directory.

b) By clicking Save you will be saving the **diffusionremote.jar** to the download directory on the local machine:



c) Copy the **diffusionremote.jar** file from the download directory to a specific folder that you can reference at a later date.



Once you have copied the remote jar file to your specific directory you are ready to build your Diffusion application.

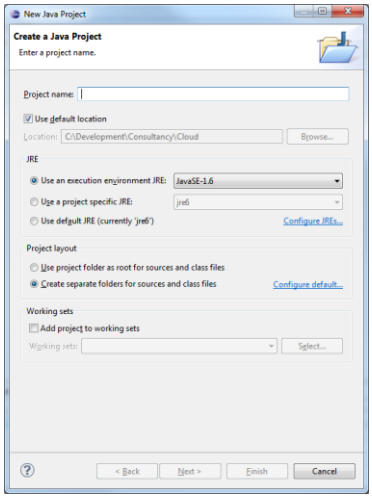
7.2. Create a standard Java project

Within the Eclipse development environment (IDE) there is a simple wizard that you can use to create a standard Java application and this section will guide you through this process. If you do not have the Eclipse IDE installed locally, you can download the application from the following URL:

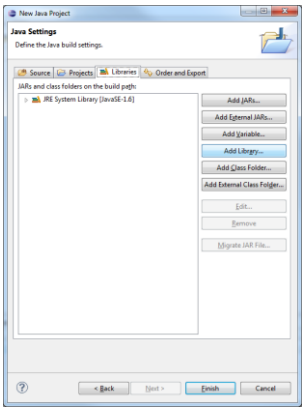
http://www.eclipse.org/downloads/

For the purpose of this sample project, the package names and the class definitions are specific to the sample code we are providing. As you begin to create your own project these definitions may well change to reflect your own development naming conventions.

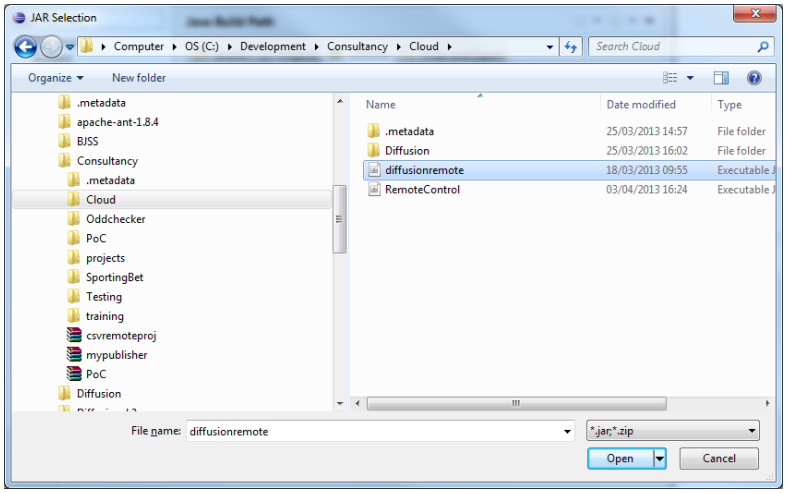
a) Create a new Java project; select File / New / Java Project:

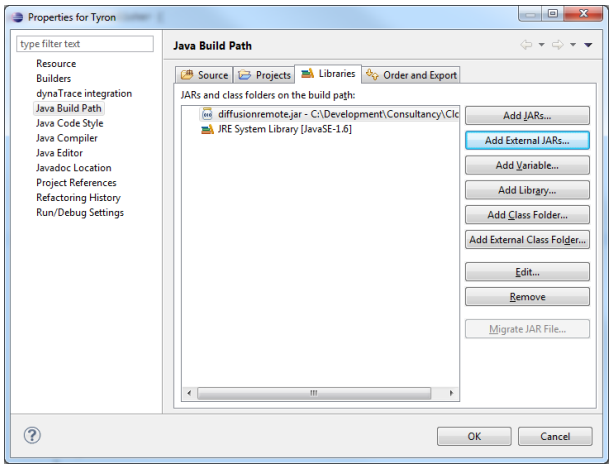


b) Enter a project name and select Next:



c) Navigate to the Libraries tab and select the Add External JARs option, then navigate to the directory that you downloaded the **diffusionremote.jar** file into, from the Diffusion image in the cloud, click Open.





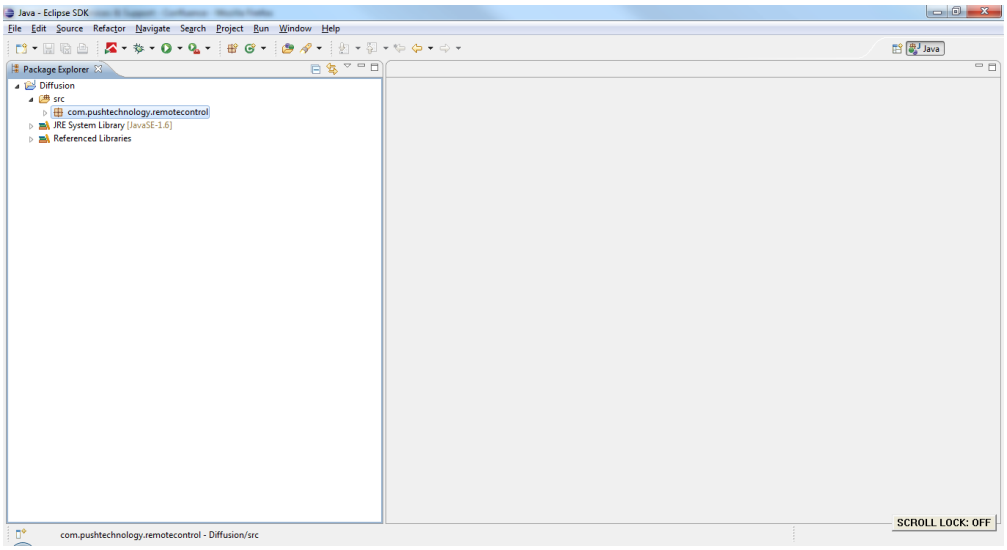
Once you have included the jar file within the project, click OK.

Eclipse will then build a Java project with the details you have provided and when complete you will be able to see the new project using the Package Explorer window within Eclipse.

The next step is to create the Remote Service Listener class file

a) Highlight the src folder in the project tree, right click and select New / Package; create a new package called com.pushtechnology.remotecontrol.

b) Highlight the package that you have created, right click and select New / File; Create a new file called MyRemoteServiceListener.java.



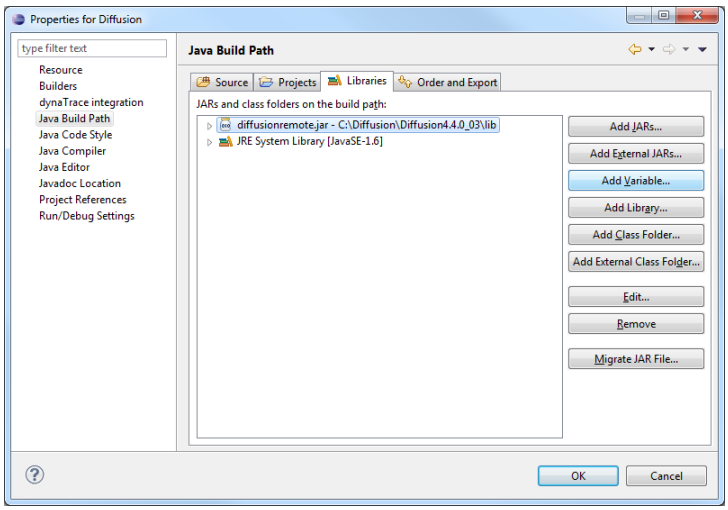
In the example above there is a package called; com.pushtechnology.remotecontrol and a class file called MyRemoteServiceListener.java.

7.3. Reference the Diffusion Remote jar file

The final step before you starting writing the Remote Service class and the Remote Publisher is to ensure the project you have created has the additional diffusionremote.jar file included as a library.

You can check this by highlighting the top level project and right click to select Properties;

a) In the window that appears select Java Build Path and then select the Libraries tab. You will see an entry to the diffusionremote.jar file in the main pane.



7.4. Build a Diffusion Remote Service Listener

To create a Remote Service class file you will need to implement the Remote Service Listener when you declare the class. The Remote Service Listener provides a number of notification methods that you can use to interact with the Diffusion instance running in the cloud. The API documentation for the Remote Service Listener can be found using the URL below:

http://docs.pushtechnology.com/docs/4.5.2/java/com/pushtechnology/diffusion/api/remote/RemoteServiceListener.html

The source examples for the Remote Service Listener can be found using the link below, and you can use this as a foundation for the application you are going to create. If you take the sample from the repository and create it within your Java project it will compile and run as is. This does however give you the opportunity to enhance the Service to make it more specific to the application you would like to build. The file you will need to reference is called MyRemoteServiceListener.java. The source for the Remote Service Listener can be obtained via this URL:

https://github.com/pushtechnology/cloud/tree/master/samples/csvremote/csvremote

If you copy all the source code from the MyRemoteServiceListener.java file into the class file you have created you will be in a position to run the application without making any modifications.

7.5. Build a Data Source Provider

The next stage to building a Diffusion application is to create a Java class that facilitates the creation of a data that can be used to publish to the Diffusion instance in the cloud.

In the sample provided the data source is a .csv file that contains price updates to individual pieces of data. The sample source file you will need to reference is called CSVDiffusion.java and can be obtained using the following URL:

https://github.com/pushtechnology/cloud/tree/master/samples/csvremote/csvremote

The example uses a comma separated value (csv) data file as the data source so when you start the data source provider you will specify the server name (the one you are going to publish messages to), the filename (included in the samples directory on GitHub) and the name of the topic you are going to use as a remote control topic to communicate with Diffusion in the cloud.

7.6. Build the Diffusion Publisher

The Diffusion publisher you are going to create will need to accept messages from the data source by reading the .csv file into memory, extracting the topic name and the fields for each line and subsequently create a topic, associate the data to that topic and publish the message.

The reading in of the .csv file is handled within a Java class and operates within a thread so as new data is generated from the data source and is received within the publisher, the application will respond to any subscriptions to topics by publishing the data to the connected clients.

There is a source code sample that provides the features of Diffusion to publish the data, via topics, to any clients that have connected to the Diffusion instance running in the cloud. The sample provides you with an insight on how to create a thread service that reads in and publishes data from a data source. The sample source file you will need to reference is called CSVPublisher.java and can be obtained using the following URL:

https://github.com/pushtechnology/cloud/tree/master/samples/csvremote/csvremote

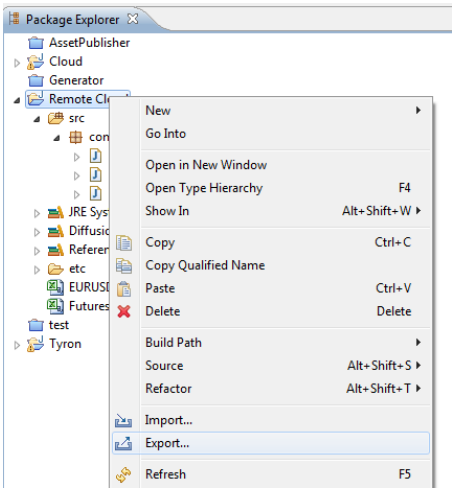
This example shows the creation of a Java thread that is executed with an instance of the Remote Service. It will accept the data from the data source provider and once the data has been transformed it will publish the data, via topic messages, out to any clients that have subscribed (shown an interest) to that data.

7.7. Build a Java Executable File

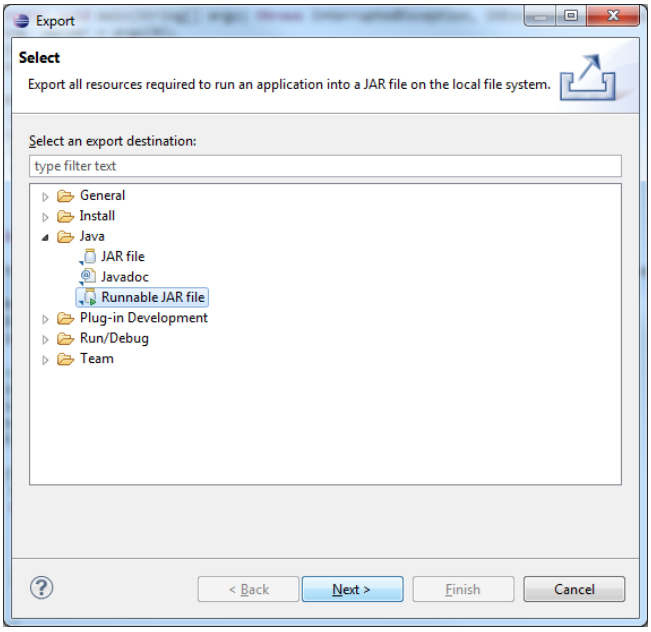
Once you have built the Data Source provider and the Remote Service Listener you will be able to run the application and publish messages from the Data Source provider to the Diffusion instance running in the cloud, the “edge” server.

In order to run the application you will need to build a Java executable file that can be run from your local machine. This next section provides a step by step guide on how to build a Java executable file.

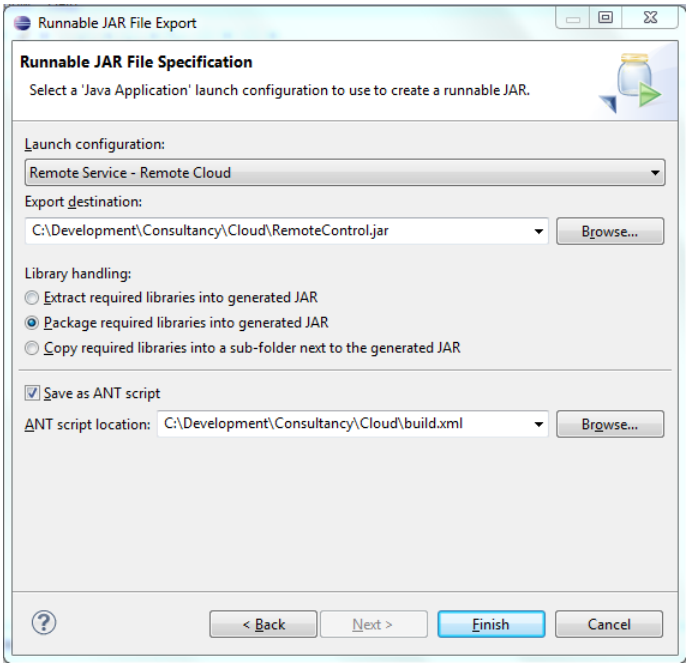
a) Within Eclipse, go to the Package Explorer panel, navigate to the project you have created and right click on the top level project name; select Export.



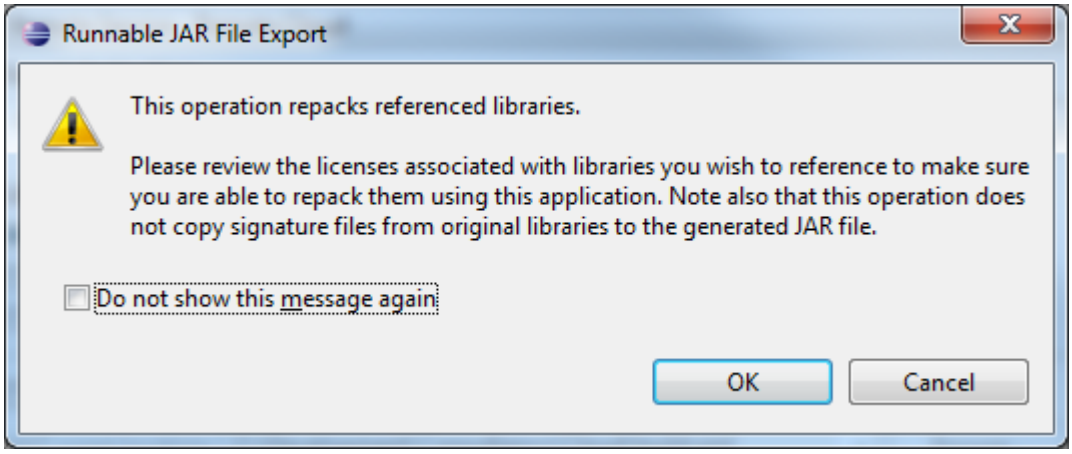
b) Highlight Java / Runnable JAR File; select Next.



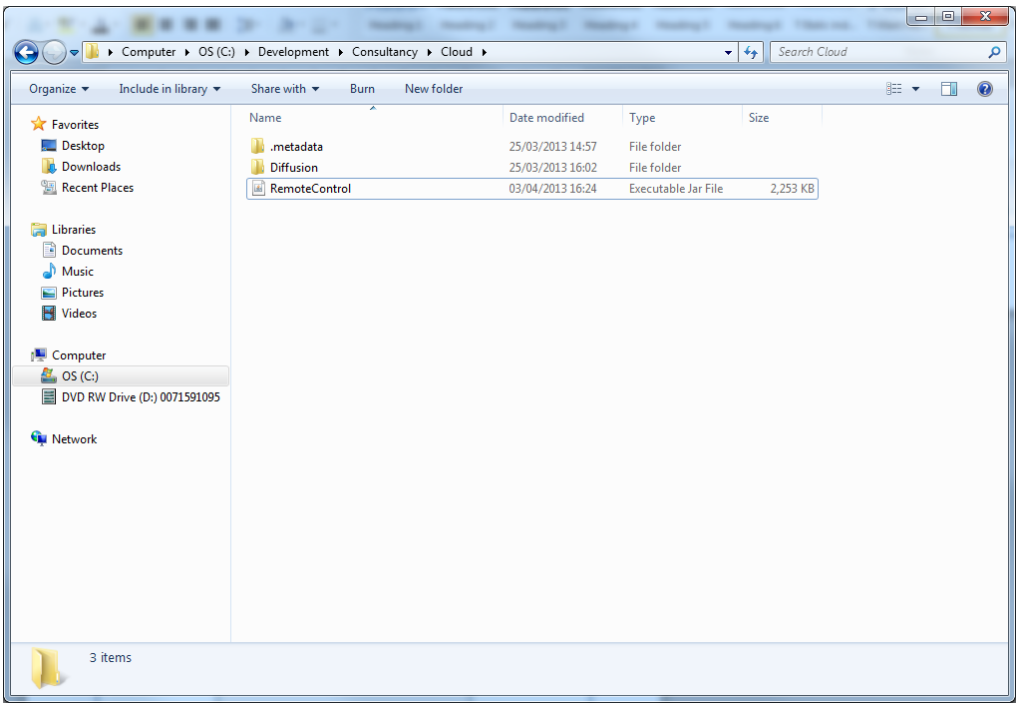
c) Select the correct launch configuration and enter the name and destination of the Jar file you are going to create, ensure you select Package required libraries into generated JAR; select Finish.
(Note: You can create an ANT build file during this process, which allows you to compile and build your application at a later date)



d) A message box will appear to confirm that the reference libraries you have included in the project can be repacked using the application you are exporting; select OK to continue.



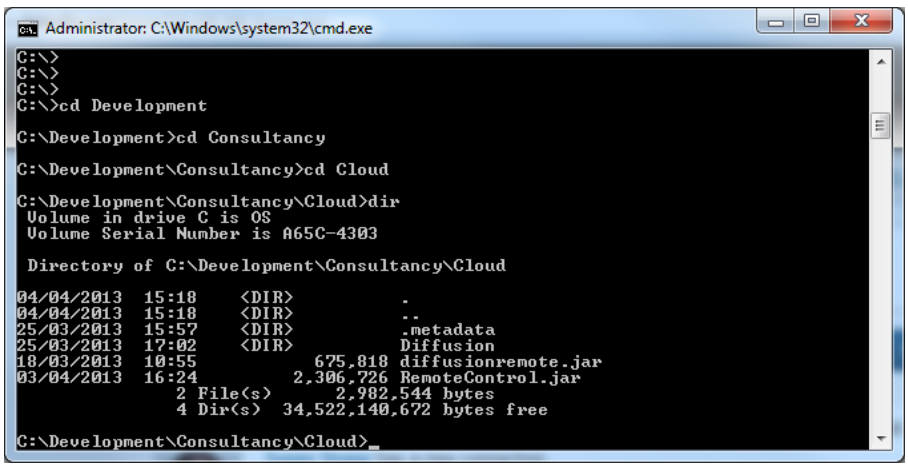
e) Once the build is complete you will find the Java executable JAR file located in the directory you specified during the export process.



You are now ready to run your application.

7.8. Start the Diffusion Remote Publisher

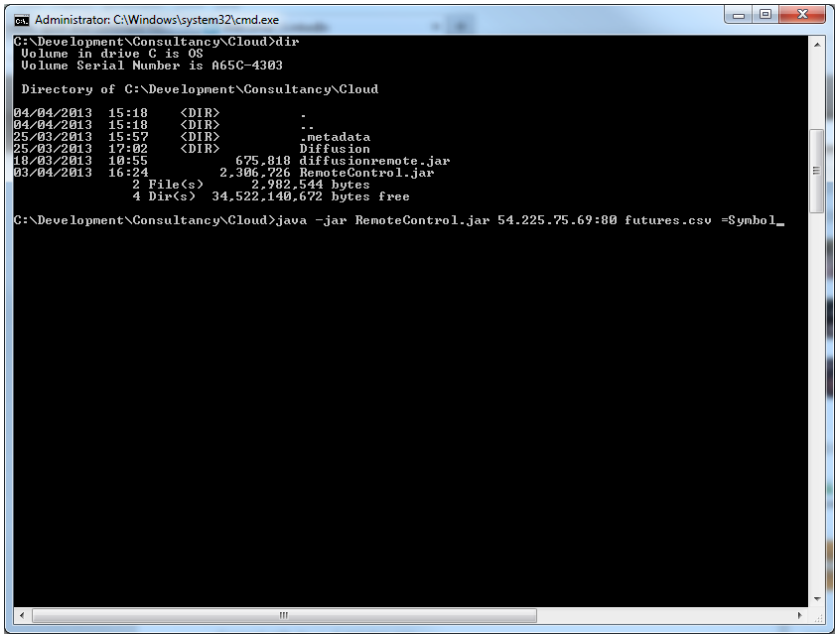
Once the instance of Diffusion in the cloud is up and running you can start the remote service that has been created locally. To do this you will need to open a command prompt window and navigate to the Java executable file that you created earlier (see section Build a Java Executable File). Ensure you have copied one of the .csv files from the Push Technology GitHub repository so you can use it as a source of data.



From this location you can issue the following command to start the Remote Service Publisher:

java -jar <<name of jar file>> <<server definition for "edge" server>> <<input data file>> <<Subscription Topic>>

e.g. java -jar RemoteControl.jar 54.225.75.69:80 futures.csv =Symbol

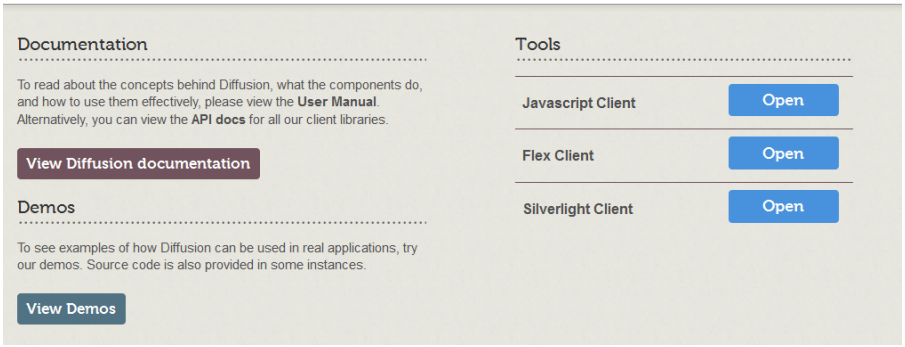


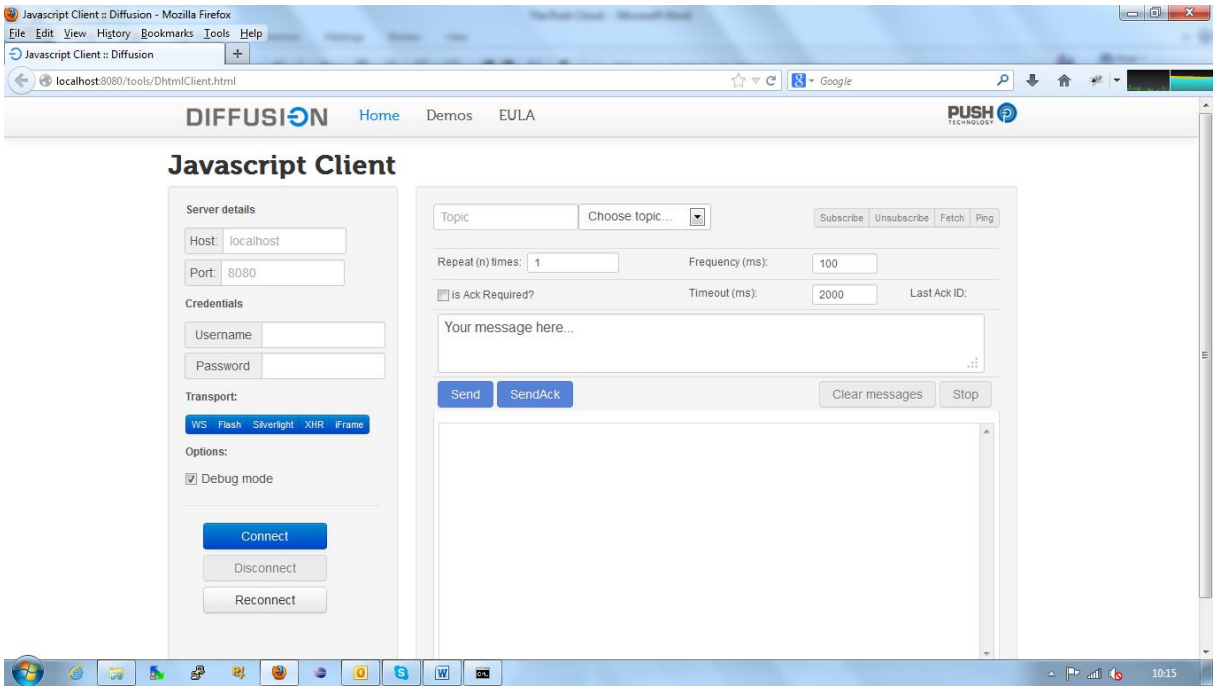
7.9. Subscribing to Data

The final step to your Diffusion application is to open a client front end and subscribe to the topics that you have created in your remote service. Once you have subscribed to the topics you will be able to see the messages being published to the client.

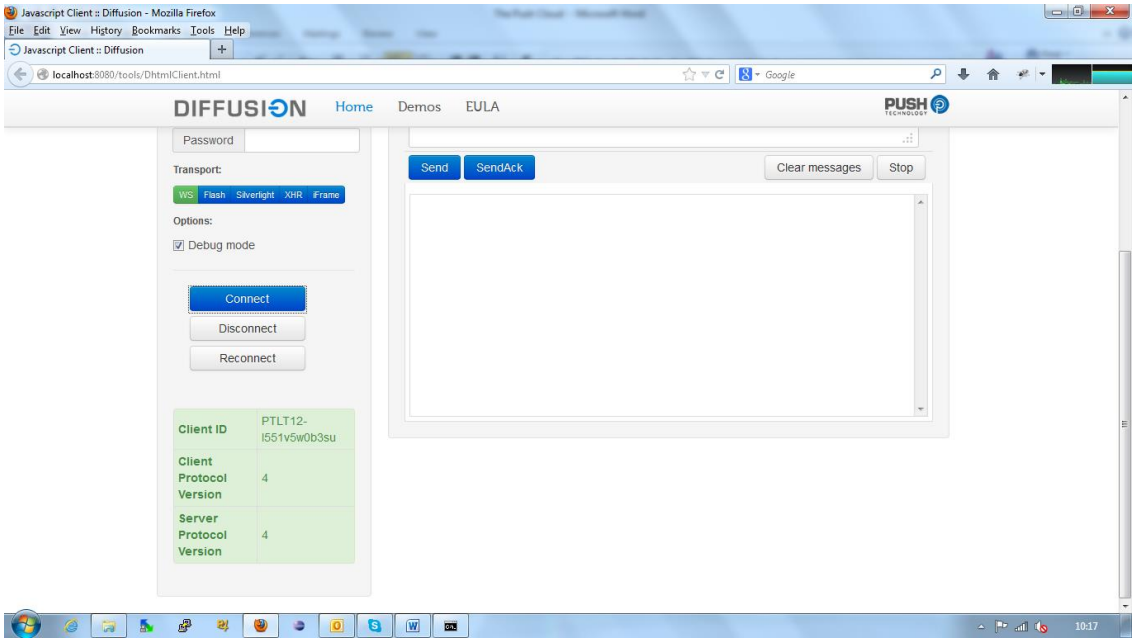
This section will guide you through a client front end initiation and shows you how to subscribe to the topics that you want to receive messages for.

a) Open the Diffusion landing page and you will notice an option to open a JavaScript client. Click to select JavaScript client:





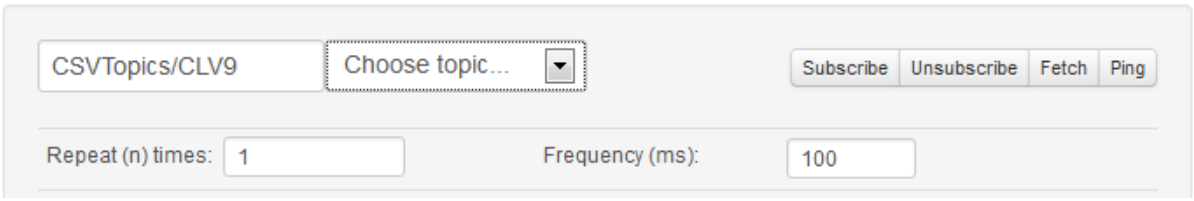
b) In the host field, enter the IP address and the port number of the Diffusion instance operating in the cloud. Select the WebSockets connection from the transport options available to you and then click Connect:



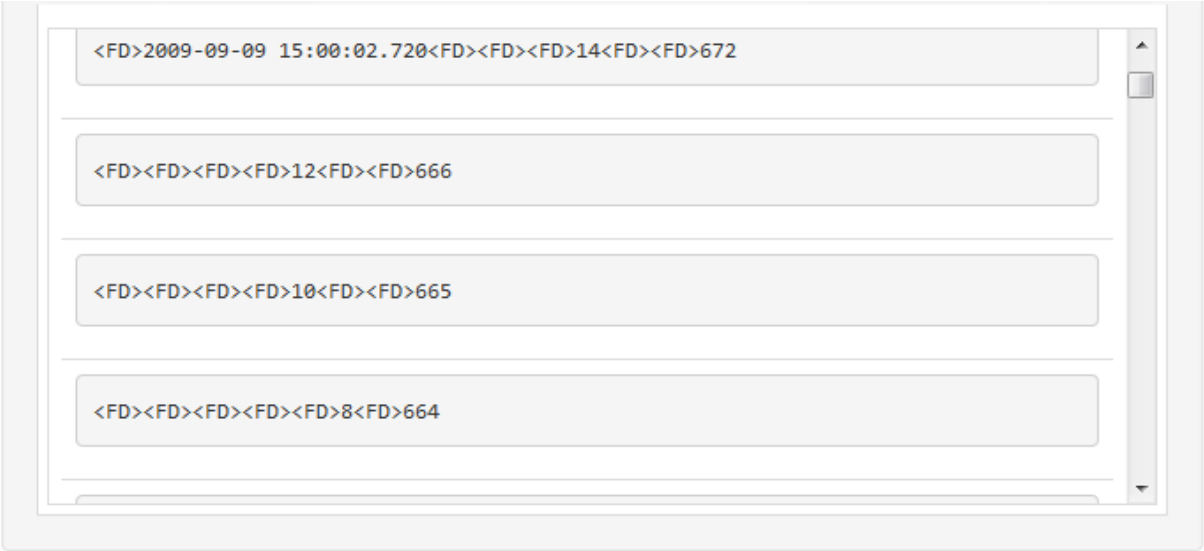
c) You will notice that the client ID has been assigned to the connection.

Client ID	PTLT12-I551v5w0b3su
Client Protocol Version	4
Server Protocol Version	4

d) Enter the name of the topic that you want to subscribe to. This will be the same name that you are publishing messages to from the Remote Service Listener that you created previously.



e) You will see in the messages box as soon as you have successfully subscribed to a topic the messages that are published from the "edge" server will start appearing in the main window.



You will notice that the messages that are displayed in the message window reflect the data that is being published from the data source .csv file, via the Remote Service Listener, to the Diffusion instance in the cloud and then to the front end client.

8. Conclusion

The creation of an application that is utilising Diffusion in the cloud is now complete. You have created a remote control service that accepts a data source (a .csv file), transforms the data into topics and topic data and then publishes the topic message to an instance of Diffusion running in the cloud.
The Diffusion publisher operating in the cloud will accept the messages, construct a topic tree and publish any messages to any connected clients that have subscribed to these topics.

The sample source files that you have been given access to will provide a framework that you can work with to enhance the application to suit the needs of your business or use case.

9. Support

There is support available to the Diffusion development community. If you navigate to the URL below you will have access to the Diffusion Developers forum that will provide you with an encyclopaedia of information. This is a publically available forum and it will allow you to communicate with other Diffusion developers to discuss and analyse particular issues or situations. You will need to create an account to access the forums and once complete, the forum is free to use. The URL for the Diffusion forum is:

www.pushtechnology.com/forums/forum/diffusion-4/

Appendix 1 - Further Information

Introduction to Diffusion
http://www.pushtechnology.com/wp-content/uploads/2013/01/Diffusion_Overview.pdf

Diffusion Documentation
http://docs.pushtechnology.com/4.5

Diffusion Downloads
http://download.pushtechnology.com/

Eclipse IDE Download
http://www.eclipse.org/downloads/

Creating an Amazon EC2 account
http://aws.amazon.com/ec2/

Sample Source Code
https://github.com/pushtechnology/cloud/tree/master/samples/csvremote/csvremote

Diffusion Remote Control JAR File
http://download.pushtechnology.com/builds/4.5.2/diffusionremote.jar

Appendix 2 - AMI Definition Table

Region	AMI Definition	EBS AMI Definitions
us-east-1 (N. Virginia)	ami-1026f98	Currently under investigation
us-west-1 (N. California)	ami-3d446b78	ami-2994b86c
us-west-2 (Oregon)	ami-c1c553f1	Currently under investigation
eu-west-1 (Ireland)	ami-ed978099	ami-45223431
sa-east-1 (Sao Paulo)	ami-5c01db41	Currently under investigation
ap-southeast-1 (Singapore)	ami-ac86c9fe	Currently under investigation
ap-southeast-2 (Sydney)	Currently under investigation	Currently under investigation
ap-northeast-1 (Tokyo)	ami-b748c4b6	Currently under investigation