Exercise - Configure applications to send or receive messages through an Event Hub

This module requires a sandbox to complete. A sandbox gives you access to free resources. Your personal subscription will not be charged. The sandbox may only be used to complete training on Microsoft Learn. Use for any other reason is prohibited, and may result in permanent loss of access to the sandbox.

Sign in to activate sandbox

You're now ready to configure your publisher and consumer applications for your Event Hub.

In this unit, you'll configure these applications to send or receive messages through your Event Hub. These applications are stored in a GitHub repository.

You'll configure two separate applications; one acts as the message sender (**SimpleSend**), the other as the message receiver (**EventProcessorSample**). These are Java applications, which enable you to do everything within the browser. However, the same configuration is needed for any platform, such as .NET.

Create a general-purpose, standard storage account

The Java receiver application, that you'll configure in this unit, stores messages in Azure Blob Storage. Blob Storage requires a storage account.

1. In Cloud Shell, create a storage account (general-purpose V2) running the **`storage account create`** command. Remember we set a default resource group and location, so even though those parameters are normally *required*, we can leave them off.

Parameter	Description
--name (required)	A name for your storage account.
--resource-group (required)	The resource group owner. We'll use the pre-created sandbox resource group.
--location (optional)	An optional location if you want the storage account in a specific place vs. the resource group location.

Set the storage account name into a variable. It must be between 3 and 24 characters in length and use numbers and lower-case letters only. It also must be unique within Azure.

```
STORAGE_NAME=storagename$RANDOM
```

Next, run the following command to create the storage account.

```
az storage account create --name $STORAGE_NAME --sku Standard_RAGRS --encryption-service blob
```

Tip If the storage account creation fails, change your environment variable, and try again.

2. List all the access keys associated with your storage account by running the `account keys list` command. It takes your account name and the resource group (which is defaulted).

```
az storage account keys list --account-name $STORAGE_NAME
```

Access keys associated with your storage account are listed. Copy and save the value of **key** for future use. You'll need this key to access your storage account.

3. View the connections string for your storage account running the following command.

```
az storage account show-connection-string -n $STORAGE_NAME
```

This command returns the connection details for the storage account. Copy and save the *value* of **connectionString**. It should look something like.

```
"DefaultEndpointsProtocol=https;EndpointSuffix=core.windows.net;AccountName=storage_account_name;AccountKey=VZjXuMeuDqjCkT60xX6L5fmtXixYuY2wiPmsrXwYHIhwo736kSAUAj08XBockRZh7CZwYxuYBPe31hi8XfHlWw=="
```

4. Create a container called **messages** in your storage account by running the following command. Use the **connectionString** you copied in the previous step.

```
az storage container create --name messages --connection-string "<connection string here>"
```

Clone the Event Hubs GitHub repository

Perform the following steps to clone the Event Hubs GitHub repository with `git`. You can execute this right in Cloud Shell.

1. The source files for the applications that you'll build in this unit are located in a [GitHub repository](). Run the following commands to make sure that you are in your home directory in Cloud Shell, and then to clone this repository.

```
cd ~
git clone https://github.com/Azure/azure-event-hubs.git
```

The repository is cloned to your home folder.

Edit SimpleSend.java

You're going to use the built-in Cloud Shell editor. You'll use the editor to modify the SimpleSend application, and add your Event Hubs namespace, Event Hub name, shared access policy name, and primary key. The main commands appear at the bottom of the editor window.

You'll need to write out your edits by pressing Ctrl+O, and then pressing Enter to confirm the output file name. Exit the editor by pressing Ctrl+X. Alternatively, the editor has a "..." menu in the top/right corner for all the editing commands.

1. Change to the **SimpleSend** folder.

```
cd ~/azure-event-hubs/samples/Java/Basic/SimpleSend/src/main/java/com/microsoft/azure/eventhubs/samples/SimpleSend
```

2. Open Cloud Shell editor in the current folder. This shows a list of files on the left and an editor space on the right.

```
code .
```

3. Open the **SimpleSend.java** file by selecting it from the file list.

4. In the editor, locate and replace the following strings:

- **`"Your Event Hubs namespace name"`** with the name of your Event Hub namespace.
- **`"Your Event Hub"`** with the name of your Event Hub.
- **`"Your policy name"`** with **RootManageSharedAccessKey**.
- **`"Your primary SAS key"`** with the value of the **primaryKey** key for your Event Hub namespace that you saved earlier.

Tip Unlike the terminal window, the editor can use typical copy/paste keyboard accelerator keys for your OS.

If you've forgotten some of them, you can switch down to the terminal window below the editor and run the **`echo`** command to list out one of the environment variables. For example:

```
echo $NS_NAME
echo $HUB_NAME
echo $STORAGE_NAME
```

When you create an Event Hubs namespace, a 256-bit SAS key called **RootManageSharedAccessKey** is created that has an associated pair of primary and secondary keys that grant send, listen, and manage rights to the namespace. In the previous unit, you displayed the key running an Azure CLI command, and you can also find this key by opening the **Shared access policies** page for your Event Hubs namespace in the Azure portal.

5. Save **SimpleSend.java** either through the "..." menu, or the accelerator key (Ctrl+S on Windows and Linux, Cmd+S on macOS).

6. Close the editor with the "..." menu, or the accelerator key CTRL+Q.

Use Maven to build SimpleSend.java

You'll now build the Java application running **mvn** commands.

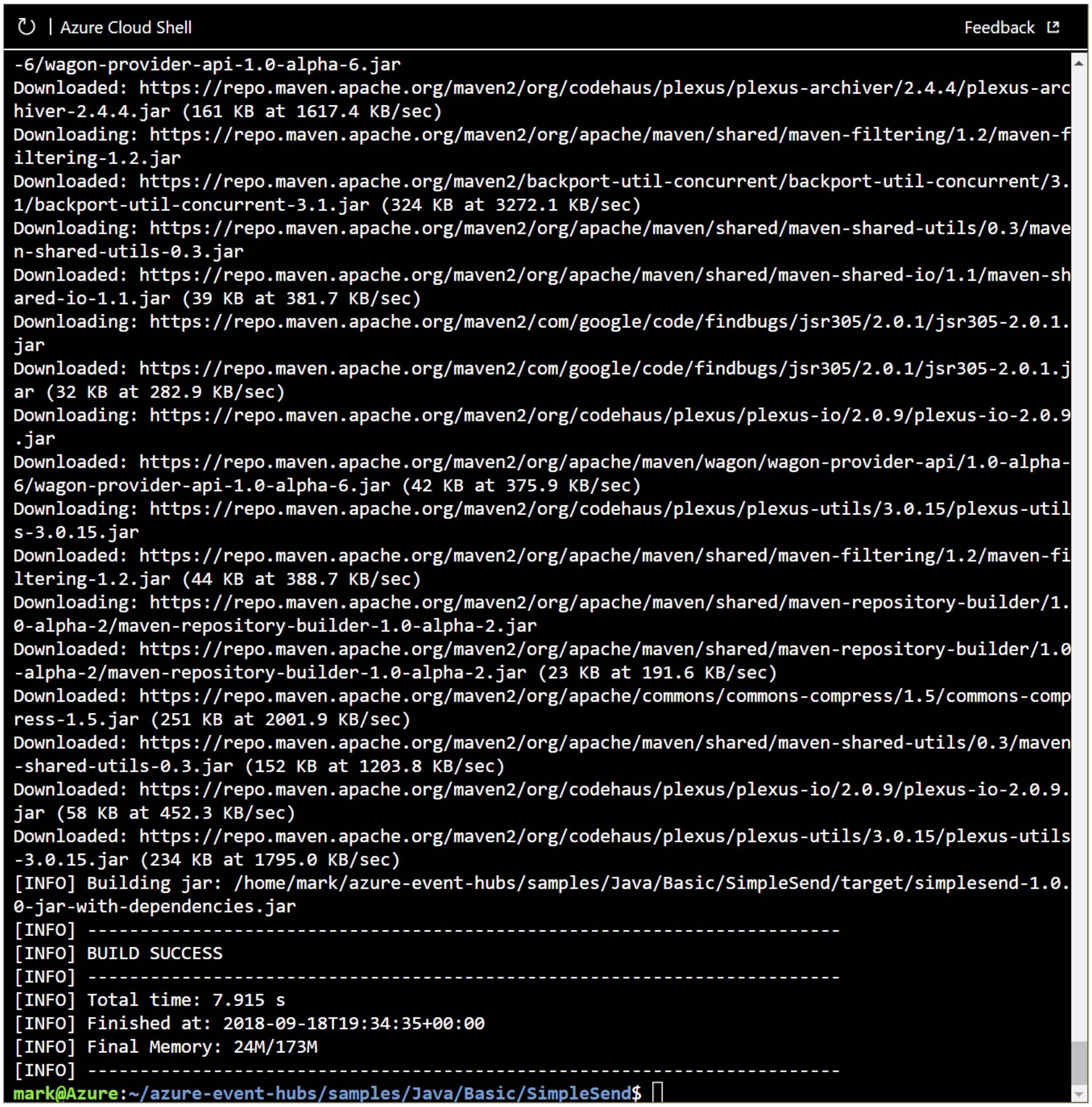
1. Revert to the main **SimpleSend** folder.

```
cd ~/azure-event-hubs/samples/Java/Basic/SimpleSend
```

2. Build the Java SimpleSend application. This ensures that your application uses the connection details for your Event Hub.

```
mvn clean package -DskipTests
```

The build process may take several minutes to complete. Ensure that you see the **[INFO] BUILD SUCCESS** message before continuing.



```
-6/wagon-provider-api-1.0-alpha-6.jar
Downloaded: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-archiver/2.4.4/plexus-arc
hiver-2.4.4.jar (161 KB at 1617.4 KB/sec)
Downloading: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-filtering/1.2/maven-f
iltering-1.2.jar
Downloaded: https://repo.maven.apache.org/maven2/backport-util-concurrent/backport-util-concurrent/3.
1/backport-util-concurrent-3.1.jar (324 KB at 3272.1 KB/sec)
Downloading: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-shared-utils/0.3/mave
n-shared-utils-0.3.jar
Downloaded: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-shared-io/1.1/maven-sh
ared-io-1.1.jar (39 KB at 381.7 KB/sec)
Downloading: https://repo.maven.apache.org/maven2/com/google/code/findbugs/jsr305/2.0.1/jsr305-2.0.1.
jar
Downloaded: https://repo.maven.apache.org/maven2/com/google/code/findbugs/jsr305/2.0.1/jsr305-2.0.1.j
ar (32 KB at 282.9 KB/sec)
Downloading: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-io/2.0.9/plexus-io-2.0.9
.jar
Downloaded: https://repo.maven.apache.org/maven2/org/apache/maven/wagon/wagon-provider-api/1.0-alpha-
6/wagon-provider-api-1.0-alpha-6.jar (42 KB at 375.9 KB/sec)
Downloading: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0.15/plexus-util
s-3.0.15.jar
Downloaded: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-filtering/1.2/maven-fi
ltering-1.2.jar (44 KB at 388.7 KB/sec)
Downloading: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-repository-builder/1.
0-alpha-2/maven-repository-builder-1.0-alpha-2.jar
Downloaded: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-repository-builder/1.0
-alpha-2/maven-repository-builder-1.0-alpha-2.jar (23 KB at 191.6 KB/sec)
Downloaded: https://repo.maven.apache.org/maven2/org/apache/commons/commons-compress/1.5/commons-comp
ress-1.5.jar (251 KB at 2001.9 KB/sec)
Downloaded: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-shared-utils/0.3/maven
-shared-utils-0.3.jar (152 KB at 1203.8 KB/sec)
Downloaded: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-io/2.0.9/plexus-io-2.0.9.
jar (58 KB at 452.3 KB/sec)
Downloaded: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0.15/plexus-utils
-3.0.15.jar (234 KB at 1795.0 KB/sec)
[INFO] Building jar: /home/mark/azure-event-hubs/samples/Java/Basic/SimpleSend/target/simplesend-1.0.
0-jar-with-dependencies.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 7.915 s
[INFO] Finished at: 2018-09-18T19:34:35+00:00
[INFO] Final Memory: 24M/173M
[INFO] ------------------------------------------------------------------------
mark@Azure:~/azure-event-hubs/samples/Java/Basic/SimpleSend$
```

Build results for sender application.

Edit EventProcessorSample.java

You'll now configure a **receiver** (also known as **subscribers** or **consumers**) application to ingest data from your Event Hub.

For the receiver application, two classes are available: **EventHubReceiver** and **EventProcessorHost**. EventProcessorHost is built on top of EventHubReceiver, but provides simpler programmatic interface than EventHubReceiver. EventProcessorHost can automatically distribute message partitions across multiple instances of EventProcessorHost using the same storage account.

In this unit, you'll use the EventProcessorHost method. You'll edit the EventProcessorSample application to add your Event Hubs namespace, Event Hub name, shared access policy name and primary key, storage account name, connection string, and container name.

1. Change to the **EventProcessorSample** folder running the following command.

```
cd ~/azure-event-hubs/samples/Java/Basic/EventProcessorSample/src/main/java/com/microsoft/azure/eventhubs/samples/eventprocessorsample
```

2. Open Cloud Shell editor.

```
code .
```

3. Select the **EventProcessorSample.java** file.

4. Locate and replace the following strings in the editor:

- **`----ServiceBusNamespaceName----`** with the name of your Event Hubs namespace.
- **`----EventHubName----`** with the name of your Event Hub.
- **`----SharedAccessSignatureKeyName----`** with **RootManageSharedAccessKey**.
- **`----SharedAccessSignatureKey----`** with the value of the **primaryKey** key for your Event Hubs namespace that you saved earlier.
- **`----AzureStorageConnectionString----`** with your storage account connection string that you saved earlier.
- **`----StorageContainerName----`** with "messages".
- **`----HostNamePrefix----`** with the name of your storage account.

5. Save **EventProcessorSample.java** either with the "..." menu, or the accelerator key (Ctrl+S on Windows and Linux, Cmd+S on macOS).

6. Close the editor.

Use Maven to build EventProcessorSample.java

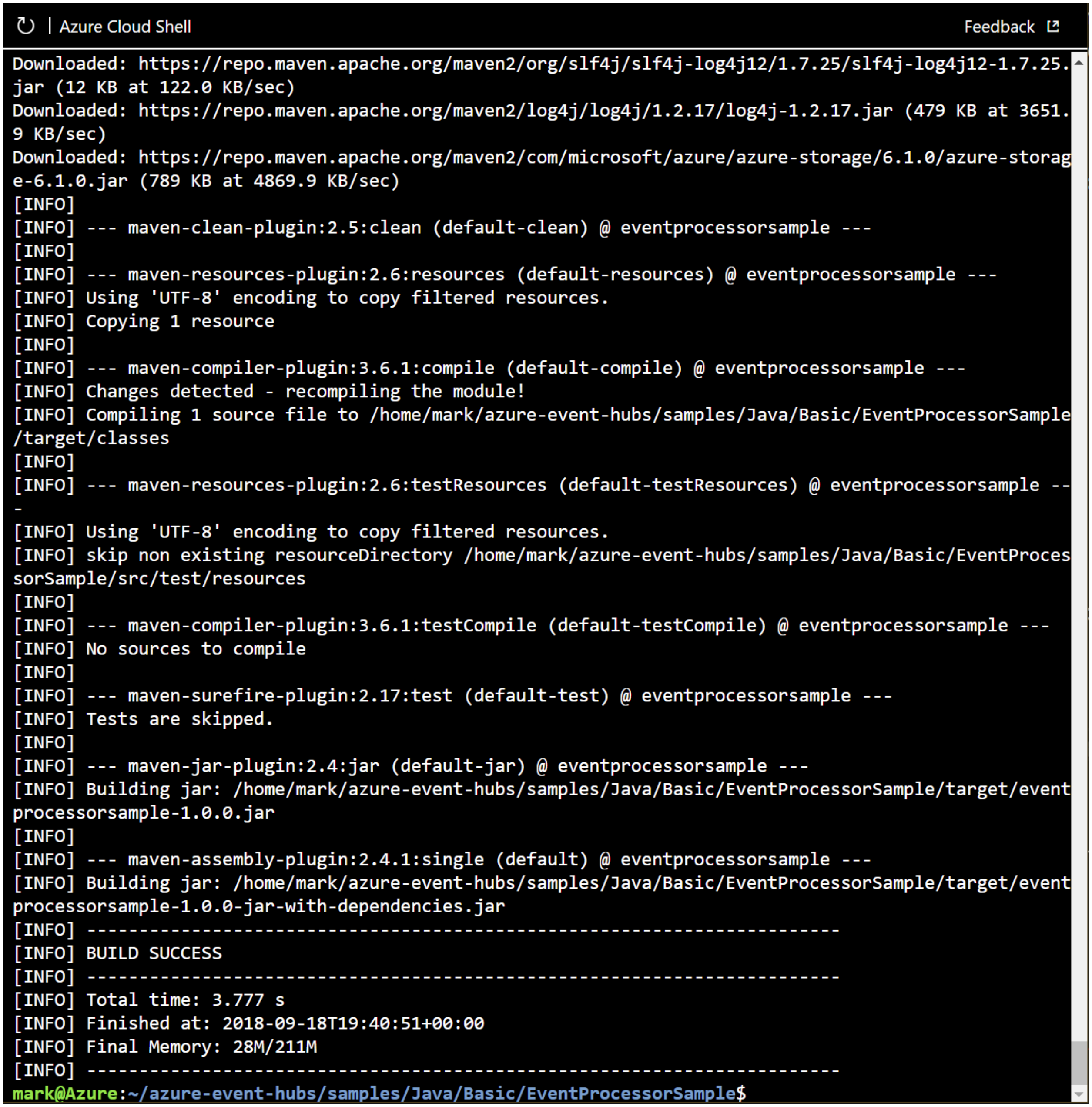
1. Change to the main **EventProcessorSample** folder running the following command.

```
cd ~/azure-event-hubs/samples/Java/Basic/EventProcessorSample
```

2. Build the Java SimpleSend application running the following command. This ensures that your application uses the connection details for your Event Hub.

```
mvn clean package -DskipTests
```

The build process may take several minutes to complete. Ensure that you see a **[INFO] BUILD SUCCESS** message before continuing.



```
Downloaded: https://repo.maven.apache.org/maven2/org/slf4j/slf4j-log4j12/1.7.25/slf4j-log4j12-1.7.25.
jar (12 KB at 122.0 KB/sec)
Downloaded: https://repo.maven.apache.org/maven2/log4j/log4j/1.2.17/log4j-1.2.17.jar (479 KB at 3651.
9 KB/sec)
Downloaded: https://repo.maven.apache.org/maven2/com/microsoft/azure/azure-storage/6.1.0/azure-storag
e-6.1.0.jar (789 KB at 4869.9 KB/sec)
[INFO]
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ eventprocessorsample ---
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ eventprocessorsample ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 1 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.6.1:compile (default-compile) @ eventprocessorsample ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to /home/mark/azure-event-hubs/samples/Java/Basic/EventProcessorSample
/target/classes
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ eventprocessorsample --
-
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /home/mark/azure-event-hubs/samples/Java/Basic/EventProces
sorSample/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.6.1:testCompile (default-testCompile) @ eventprocessorsample ---
[INFO] No sources to compile
[INFO]
[INFO] --- maven-surefire-plugin:2.17:test (default-test) @ eventprocessorsample ---
[INFO] Tests are skipped.
[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ eventprocessorsample ---
[INFO] Building jar: /home/mark/azure-event-hubs/samples/Java/Basic/EventProcessorSample/target/event
processorsample-1.0.0.jar
[INFO]
[INFO] --- maven-assembly-plugin:2.4.1:single (default) @ eventprocessorsample ---
[INFO] Building jar: /home/mark/azure-event-hubs/samples/Java/Basic/EventProcessorSample/target/event
processorsample-1.0.0-jar-with-dependencies.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 3.777 s
[INFO] Finished at: 2018-09-18T19:40:51+00:00
[INFO] Final Memory: 28M/211M
[INFO] ------------------------------------------------------------------------
mark@Azure:~/azure-event-hubs/samples/Java/Basic/EventProcessorSample$
```

Build results for receiver application.

Start the sender and receiver apps

1. Run Java application from the command line by running the **java** command, and specifying a .jar package. Run the following commands to start the SimpleSend application.

1
2

```
cd ~/azure-event-hubs/samples/Java/Basic/SimpleSend
java -jar ./target/simplesend-1.0.0-jar-with-dependencies.jar
```

2. When you see **Send Complete...**, press Enter.

```
jar-with-dependencies.jar
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details
.
2018-09-18T19:42:15.146Z: Send Complete...
```

3. Start the EventProcessorSample application running the following command.

```
cd ~/azure-event-hubs/samples/Java/Basic/EventProcessorSample
java -jar ./target/eventprocessorsample-1.0.0-jar-with-dependencies.jar
```

4. When messages stop appearing on the console, press Enter or press CTRL+C to end the program.

```
...
SAMPLE: Partition 0 checkpointing at 1064,19
SAMPLE (3,1120,20): "Message 80"
SAMPLE (3,1176,21): "Message 84"
SAMPLE (3,1232,22): "Message 88"
SAMPLE (3,1288,23): "Message 92"
SAMPLE (3,1344,24): "Message 96"
SAMPLE: Partition 3 checkpointing at 1344,24
SAMPLE (2,1120,20): "Message 83"
SAMPLE (2,1176,21): "Message 87"
SAMPLE (2,1232,22): "Message 91"
SAMPLE (2,1288,23): "Message 95"
SAMPLE (2,1344,24): "Message 99"
SAMPLE: Partition 2 checkpointing at 1344,24
SAMPLE: Partition 1 batch size was 3 for host mystorageacct2018-46d60a17-7060-4b53-b0e0-cca70c970a47
SAMPLE (0,1120,20): "Message 81"
SAMPLE (0,1176,21): "Message 85"
SAMPLE: Partition 0 batch size was 10 for host mystorageacct2018-46d60a17-7060-4b53-b0e0-cca70c970a47
SAMPLE: Partition 0 got event batch
SAMPLE (0,1232,22): "Message 89"
SAMPLE (0,1288,23): "Message 93"
SAMPLE (0,1344,24): "Message 97"
SAMPLE: Partition 0 checkpointing at 1344,24
SAMPLE: Partition 3 batch size was 8 for host mystorageacct2018-46d60a17-7060-4b53-b0e0-cca70c970a47
SAMPLE: Partition 2 batch size was 9 for host mystorageacct2018-46d60a17-7060-4b53-b0e0-cca70c970a47
SAMPLE: Partition 0 batch size was 3 for host mystorageacct2018-46d60a17-7060-4b53-b0e0-cca70c970a47
```

Summary

You've now configured a sender application ready to send messages to your Event Hub. You've also configured a receiver application ready to receive messages from your Event Hub.