Deploy Geotab-to-Velocity-gRPC-DataFeed to Azure App Service using Visual Studio Code

Geotab-to-Velocity-gRPC-DataFeed is a C#.Net console app that can be configured and run to send event messages to a gRPC service. Intended to be deployed to an Azure portal as an Azure WebJob running in an App Service to support real-time demonstrations for Velocity, the app provides a simulated stream of event messages so that Velocity can receive them as if from remote sensors sending updates.

These instructions will guide you through the process of deploying your own Geotab-to-Velocity-gRPC-DataFeed app to support your industry demos of Velocity. Specifically, it will lead you through the following steps:

- Set up your deployment environment
- Clone the Geotab-to-Velocity-gRPC-DataFeed app
- Configure it to use your demo simulation data and gRPC endpoint
- Create an App Service resource to host the Geotab-to-Velocity-gRPC-DataFeed app
- Deploy your Geotab-to-Velocity-gRPC-DataFeed to your App Service

1. Gather your deployment resources

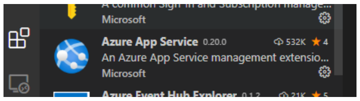
You'll need:

- An Azure account with an active subscription. Create one for free.
- A gRPC feed in Velocity configured with the schema from the Geotab to Velocity gRPC schema.csv file.
- A Geotab account username and password with a paid or trial service and database.
- Visual Studio Code (VS Code) installed on your local machine. Get it here.
- The Azure App Service extension for VS Code (Get it here or install from within VS Code)
- Git installed on your local machine. Get it here.

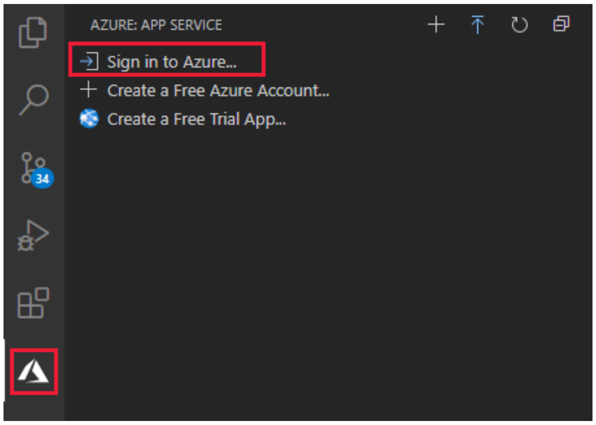
2. Sign in to Azure in VS Code

If you already use the Azure service extensions, you should already be logged in and can skip this step. If you don't use the Azure service extensions, continue in this section to install it.

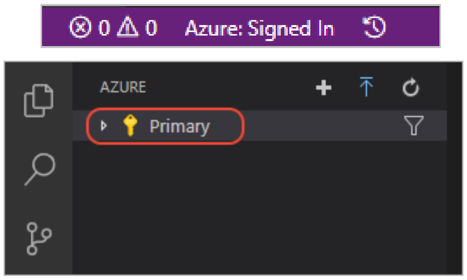
1. In VS Code click **View > Extensions**
2. In the search bar type **Azure App Service**
3. Click **Install** on the Azure App Service extension


A common sign-in and subscription manage...
Microsoft
Azure App Service 0.20.0
532K
4
An Azure App Service management extensio...
Microsoft
Azure Event Hub Explorer 0.1.2
21K
5

Once you've installed the Azure App Service extension in VS Code, you need to sign into your Azure account by navigating to the **Azure Explorer**, select **Sign in to Azure**, and follow the prompts. (If you have multiple Azure extensions installed, select **App Service**.)



After signing in, verify that the email address of your Azure account (or "Signed In") appears in the Status Bar and your subscription(s) appears in the Azure Explorer:

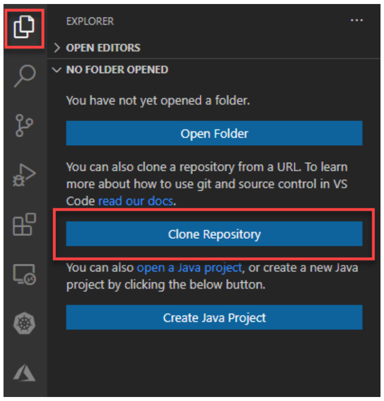


3. Clone the Geotab-to-Velocity-gRPC-DataFeed repo

Create the app by cloning a Git repository. Two methods for doing so are illustrated below.

Method 1: Use Git in VS Code:

1. In VS Code, select the explorer icon to open the Files and Folders Explorer, then click **Clone Repository**
2. Paste **https://github.com/kengorton/Geotab-to-Velocity-gRPC-DataFeed.git** into the search bar and click '**Clone from URL https://github.com/kengorton/Geotab-to-Velocity-gRPC-DataFeed.git**'
3. Navigate to the folder where you will save the cloned application files and click **Select Repository Location**.

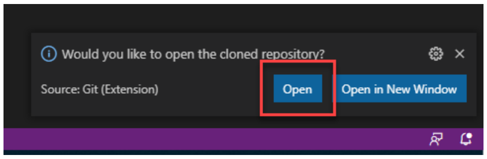


You have not yet opened a folder.

You can also clone a repository from a URL. To learn more about how to use git and source control in VS Code read our docs.

You can also open a Java project, or create a new Java project by clicking the below button.

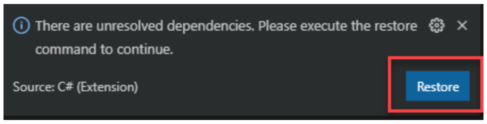
4. Click **Open**.



Would you like to open the cloned repository?

Source: Git (Extension)

5. If a message appears indicating unresolved dependencies click **Restore**.



There are unresolved dependencies. Please execute the restore command to continue.

Source: C# (Extension)

6. The application files appear in the **Files and Folders Explorer**.

Method 2: Use Git directly:

1. Open a terminal command prompt and change directories to the location where you want to create the app folder.

   Or

1. In Windows Explorer in the folder where you want to create the app, right click and click **Git Bash Here** to open a Git Bash command window.
2. Enter the following git command in the terminal or Git Bash window to clone the repository:

   git clone https://github.com/kengorton/Geotab-to-Velocity-gRPC-DataFeed.git

3. **cd** into the correct subfolder of the cloned repo directory

4. Install the application's dependencies by running the command:

   npm install

5. Start VS Code with the following command:

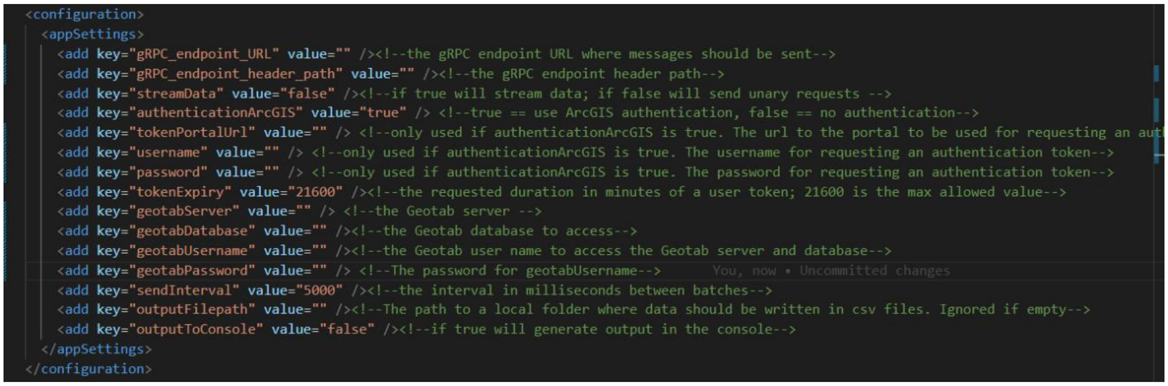
   code .

4. Update app.config to reflect your gRPC endpoint URL, gRPC endpoint header path, ArcGIS credentials (if required) and the Geotab server, database, username and password

For this step you’ll need the gRPC endpoint URL and header path for your feed. To obtain these, in Velocity go to your feed’s details page and note the values in the Feed Details section.

The app.config file in the Geotab-to-Velocity-gRPC-DataFeed code files contains a set of key/value pairs that the app uses to initialize settings for your deployment. You will need to update some of the values in this file in order to configure the app for your needs.

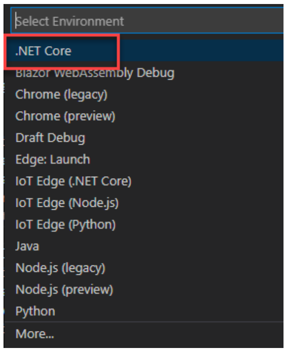
4. In VS Code’s **File and Folders Explorer** click app.config to open the file in the VS Code editor.



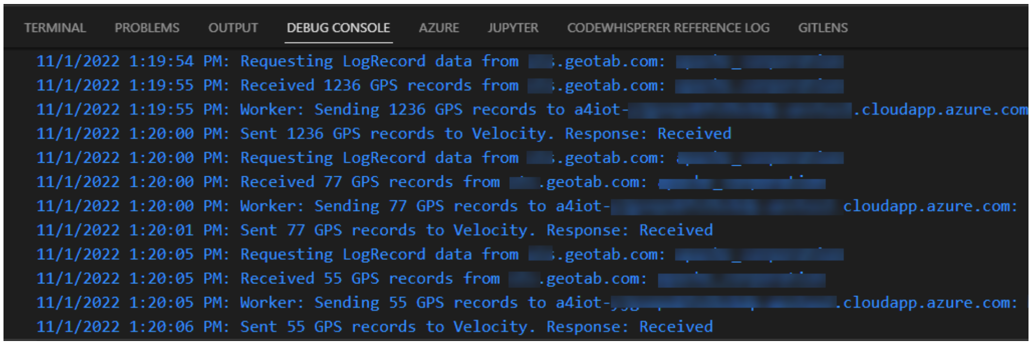
```
<configuration>
  <appSettings>
    <add key="gRPC_endpoint_URL" value="" /><!--the gRPC endpoint URL where messages should be sent-->
    <add key="gRPC_endpoint_header_path" value="" /><!--the gRPC endpoint header path-->
    <add key="streamData" value="false" /><!--if true will stream data; if false will send unary requests -->
    <add key="authenticationArcGIS" value="true" /> <!--true == use ArcGIS authentication, false == no authentication-->
    <add key="tokenPortalUrl" value="" /> <!--only used if authenticationArcGIS is true. The url to the portal to be used for requesting an aut
    <add key="username" value="" /> <!--only used if authenticationArcGIS is true. The username for requesting an authentication token-->
    <add key="password" value="" /> <!--only used if authenticationArcGIS is true. The password for requesting an authentication token-->
    <add key="tokenExpiry" value="21600" /><!--the requested duration in minutes of a user token; 21600 is the max allowed value-->
    <add key="geotabServer" value="" /> <!--the Geotab server -->
    <add key="geotabDatabase" value="" /><!--the Geotab database to access-->
    <add key="geotabUsername" value="" /><!--the Geotab user name to access the Geotab server and database-->
    <add key="geotabPassword" value="" /> <!--The password for geotabUsername-->      You, now • Uncommitted changes
    <add key="sendInterval" value="5000" /><!--the interval in milliseconds between batches-->
    <add key="outputFilepath" value="" /><!--The path to a local folder where data should be written in csv files. Ignored if empty-->
    <add key="outputToConsole" value="false" /><!--if true will generate output in the console-->
  </appSettings>
</configuration>
```

5. Enter or update values for the keys as follows:
   - gRPC_endpoint_URL - In the empty quotes after ‘value=’ paste the complete gRPC endpoint URL you noted above.
   - gRPC_endpoint_header_path – In the empty quotes after ‘value=’ paste the complete gRPC endpoint header path you noted above.
   - streamData – set to True to have the app send data in a stream (better for high velocity data), or False to send data in discrete requests (not recommended for high velocity data).
   - authenticationArcGIS - True if your Velocity feed requires ArcGIS authentication, false if not.
   - tokenPortalUrl – Used only if authenticationArcGIS is true. The root url to the ArcGIS portal to be used for obtaining a token. This is the home to your ArcGIS Online organization, not the url for Velocity.
   - username – Used only if authenticationArcGIS is true. The ArcGIS Online username for generating a token.
   - password – Used only if authenticationArcGIS is true. The ArcGIS Online user’s password for generating a token.
   - tokenExpiry – Used only if authenticationArcGIS is true. The token expiration in minutes from 1 to 21600.
   - geotabServer – Enter the name of the Geotab server without **https://**. If using a trial Geotab database the server is **mypreview.geotab.com**.
   - geotabDatabase – Enter the name of the database you set up in Geotab.
   - geotabUsername – The username for your Geotab account with access to the server and database.
   - geotabPassword – The password for the above geotabUsername.
   - sendInterval – Enter the number of milliseconds between batches sent to the gRPC endpoint. This time includes the time required to send a batch. Thus, if this value is set to 1000ms, and it takes 700ms to send a batch, the app will wait 300ms before sending the next batch. If it takes longer than this value to send a batch, it will not wait before sending the next batch.
   - outputFilepath – The path to a local folder where data should be written in csv files. Ignored if empty. This is intended for diagnostic purposes only and should normally be left blank.
   - outputToConsole – Enter true or false to indicate whether to generate detailed data output in the console. This is intended for diagnostic purposes only and should normally be set to false.
6. Click File > Save.

5. **(Optional) Run your local Geotab-to-Velocity-gRPC-DataFeed app**
   1. In VS Code click Run > Start Debugging.
   2. If a message appears prompting you to select an Environment, select **.NET Core**.



After a pause while the app initializes and loads in the simulation file you configured, in the VS Code Debug Console you should see a scrolling list of messages indicating that the app has fetched LogRecord data from Geotab and sent the data to your gRPC endpoint.

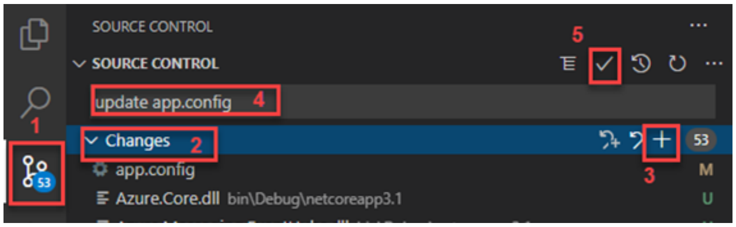


```
11/1/2022 1:19:54 PM: Requesting LogRecord data from [redacted].geotab.com: [redacted]
11/1/2022 1:19:55 PM: Received 1236 GPS records from [redacted].geotab.com: [redacted]
11/1/2022 1:19:55 PM: Worker: Sending 1236 GPS records to a4iot-[redacted].cloudapp.azure.com
11/1/2022 1:20:00 PM: Sent 1236 GPS records to Velocity. Response: Received
11/1/2022 1:20:00 PM: Requesting LogRecord data from [redacted].geotab.com: [redacted]
11/1/2022 1:20:00 PM: Received 77 GPS records from [redacted].geotab.com: [redacted]
11/1/2022 1:20:00 PM: Worker: Sending 77 GPS records to a4iot-[redacted]cloudapp.azure.com:
11/1/2022 1:20:01 PM: Sent 77 GPS records to Velocity. Response: Received
11/1/2022 1:20:05 PM: Requesting LogRecord data from [redacted].geotab.com: [redacted]
11/1/2022 1:20:05 PM: Received 55 GPS records from [redacted].geotab.com: [redacted]
11/1/2022 1:20:05 PM: Worker: Sending 55 GPS records to a4iot-[redacted].cloudapp.azure.com:
11/1/2022 1:20:06 PM: Sent 55 GPS records to Velocity. Response: Received
```

3. **Commit changes in the app.config file to the local repo**

Having edited the app.config file, you must commit the changes to your local repository so that they will be reflected in the published App Service.

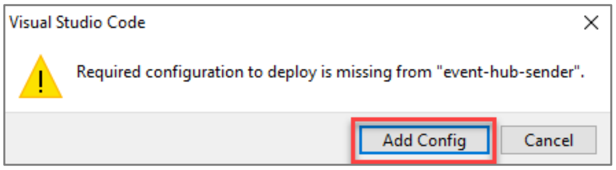
1. Open the Source Control explorer.
2. Select the Changes list.
3. Click the **+** button to stage all changes for a commit.
4. Enter a comment such as “updated app.config” to indicate the reason for the commit.
5. Click the checkmark button ✓ to commit the changes.



4. **Create App service resource in VS Code**

   1. From the command palette (**Ctrl**+**Shift**+**P** on Windows**, Cmd**+**Shift**+**P** on Mac), type "create web" and select **Azure App Service: Create New Web App...Advanced**. You use the advanced command to have full control over the deployment including resource group, App Service Plan, and operating system rather than use Linux defaults.

2. If a message appears indicating a required configuration to deploy is missing, click **Add Config**.



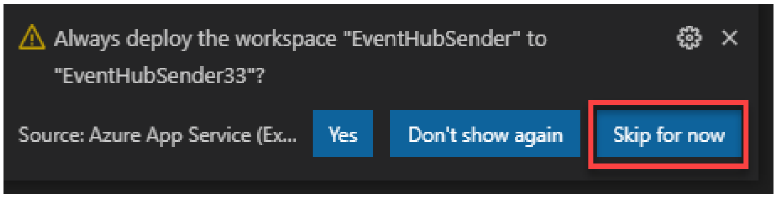
Visual Studio Code

Required configuration to deploy is missing from "event-hub-sender".

3. Respond to the prompts as follows:

- If prompted to do so, select your **Subscription** account.
- For **Enter a globally unique name**, enter a name that's unique across all of Azure. Use only alphanumeric characters ('A-Z', 'a-z', and '0-9') and hyphens ('-')
- Select **Create new resource group** and provide a name like Geotab-to-Velocity-gRPC-DataFeed-rg.
- Select the **.Net Core 3.1 (LTS)** runtime stack.
- Select Windows as the operating system.
- Select a location near you or where you want the app to run.
- Select **Create a new App Service plan**, provide a name like Geotab-to-Velocity-gRPC-DataFeed-plan, and select any pricing tier. The app will incur costs on any pricing tier other than **F1 Free**.
- Select **Skip for now** for the Application Insights resource.

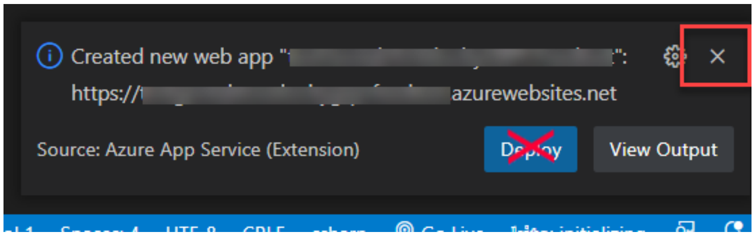
If a message asks to 'Always deploy the workspace Geotab-to-Velocity-gRPC-DataFeed to ' this App Service click **Skip for now**.



Always deploy the workspace "EventHubSender" to "EventHubSender33"?

Source: Azure App Service (Ex...

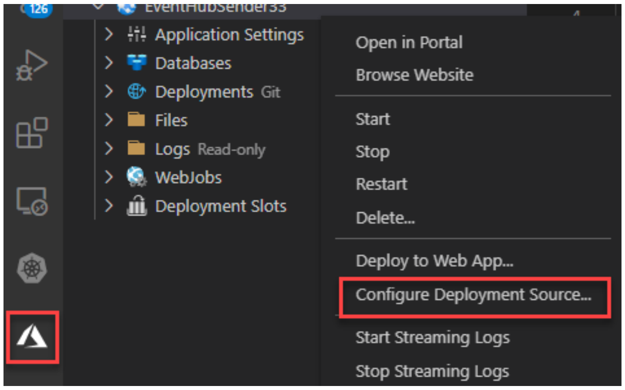
4. After a short time, VS Code notifies you that creation is complete. Close the notification with the **X** button: Do not click **Deploy.**



Created new web app "[redacted]": https://[redacted]azurewebsites.net

Source: Azure App Service (Extension)

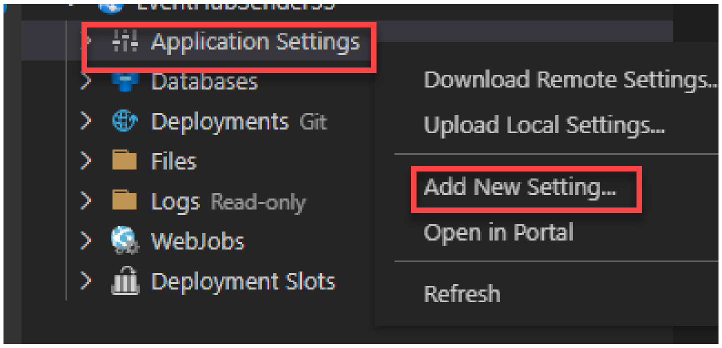
5. With the web app in place, you next instruct VS Code to deploy your code from the local Git repo. Select the Azure icon to open the **Azure App Service** explorer, expand your subscription node, right-click the name of the web app you just created, and select **Configure Deployment Source**.



6. When prompted, select **LocalGit**.

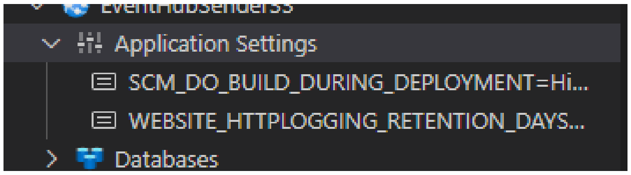
When deploying to an App Service on Windows, you need to create an additional setting before deployment:

i. In VS Code, expand the node for the new App Service, right-click **Application Settings**, and select **Add New Setting**: This will open the command palette.



ii. Enter **SCM_DO_BUILD_DURING_DEPLOYMENT** for the setting key and **1** for the setting value. This setting forces the server to run npm install upon deployment.

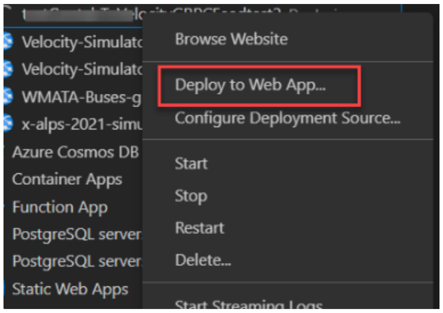
iii. Expand the **Application Settings** node to verify the setting is in place.



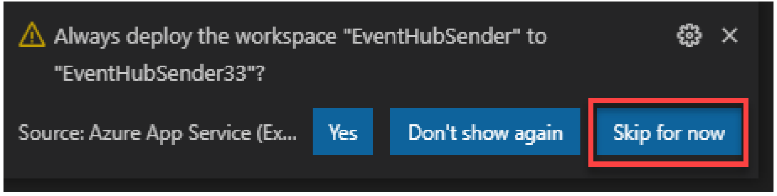
SCM_DO_BUILD_DURING_DEPLOYMENT=Hi...

WEBSITE_HTTPLOGGING_RETENTION_DAYS...

7. Right-click the new app and click **Deploy to Web App…** to deploy your code to Azure:



8. If prompted, select your **subscription** account again.
9. If a message appears about uncommitted change(s) in your local repo, click **Deploy Anyway**.
10. If a message asks to ‘Always deploy the workspace Geotab-to-Velocity-gRPC-DataFeed to’ this App Service click **Skip for now**.



Always deploy the workspace "EventHubSender" to "EventHubSender33"?

Source: Azure App Service (Ex...

Congratulations. You have deployed an Azure App Service and WebJob that continuously fetches Geotab LogRecord data and sends it to your Velocity gRPC feed.

DATA DISTRIBUTION PERMISSION

Esri – the sample data and related material(s) may be used for **Esri internal use only**:

_X_ Demonstrations/Benchmarks

_X_ Trade Shows/Esri User Conferences

_X_ Esri World Wide Web

_X_ Esri Educational/Training Materials

_X_ General Marketing, Advertising, and Promotion (i.e. brochures, slides, videos, etc.)

Esri Domestic Business Partners – the sample data and related material(s) may be forwarded to Esri Business Partners for:

_X_ Demonstrations/Benchmarks

_X_ Trade Shows/Esri User Conferences

Esri International Distributors – the sample data and related material(s) may be forwarded to Esri International Distributors for:

_X_ Demonstrations/Benchmarks

_X_ Trade Shows/Esri User Conferences