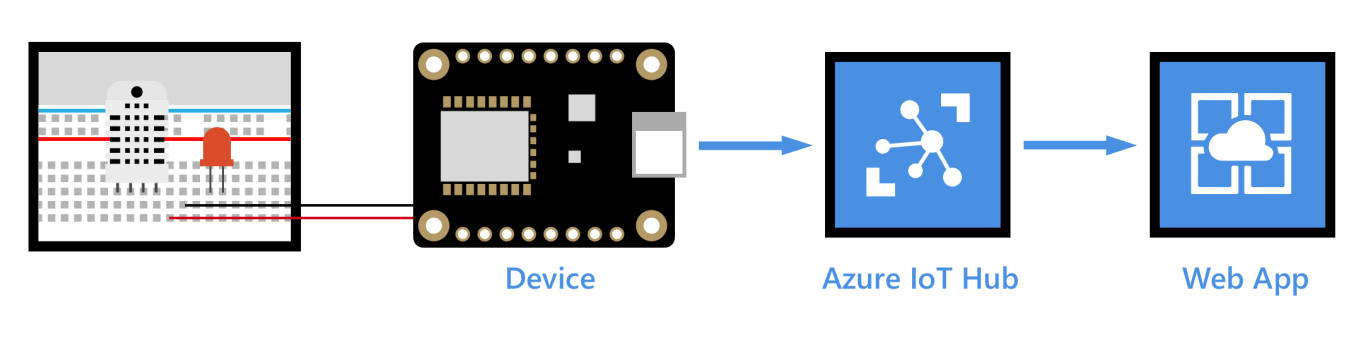
Filter by title
Data Visualization in Power BI
Data Visualization with Web Apps
Weather forecast using Azure Machine Learning
Device management with Azure IoT Toolkit for VS Code
Device management with IoT extension for Azure CLI
Remote monitoring and notifications with Logic Apps
Troubleshoot
Reference
Related
Resources
Download PDF

Visualize real-time sensor data from your Azure IoT hub by using the Web Apps feature of Azure App Service

04/11/2018 • 4 minutes to read • Contributors all



In this article
What you learn
What you do
What you need
Create a web app
Add a consumer group to your IoT hub
Configure the web app to read data from your IoT hub
Upload a web application to be hosted by the web app
Open the web app to see real-time temperature and humidity data from your IoT hub
Next steps

Is this page helpful? Yes No

> Note
>
> Before you start this tutorial, set up your device. In the article, you set up your Azure IoT device and IoT hub, and you deploy a sample application to run on your device. The application sends collected sensor data to your IoT hub.

What you learn

In this tutorial, you learn how to visualize real-time sensor data that your IoT hub receives by running a web application that is hosted on a web app. If you want to try to visualize the data in your IoT hub by using Power BI, see Use Power BI to visualize real-time sensor data from Azure IoT Hub.

What you do

- Create a web app in the Azure portal.
- Get your IoT hub ready for data access by adding a consumer group.
- Configure the web app to read sensor data from your IoT hub.
- Upload a web application to be hosted by the web app.
- Open the web app to see real-time temperature and humidity data from your IoT hub.

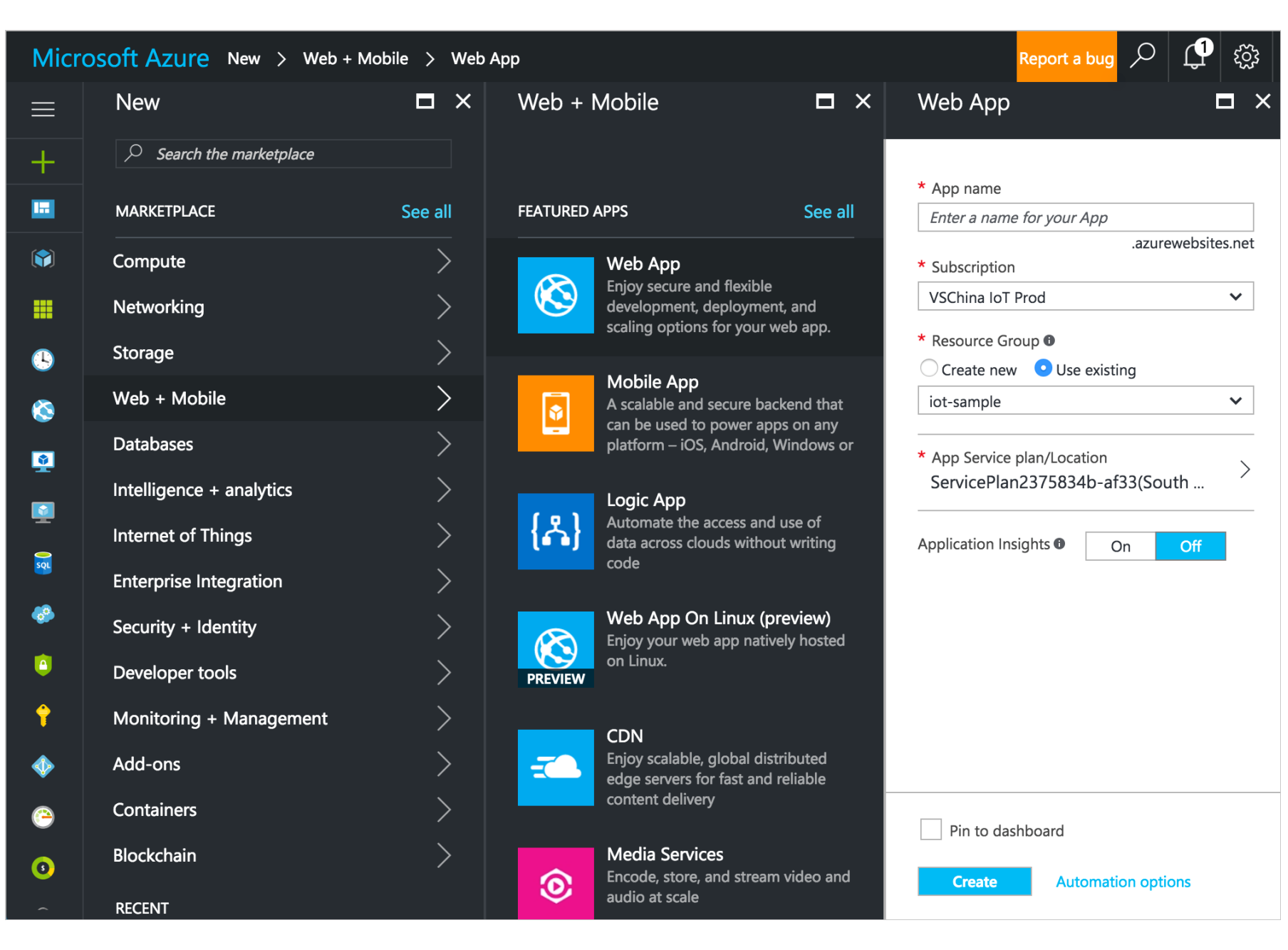
What you need

- Set up your device, which covers the following requirements:
  - An active Azure subscription
  - An Iot hub under your subscription
  - A client application that sends messages to your Iot hub
- Download Git

Create a web app

1. In the Azure portal, click **Create a resource** > **Web + Mobile** > **Web App**.
2. Enter a unique job name, verify the subscription, specify a resource group and a location, select **Pin to dashboard**, and then click **Create**.

   We recommend that you select the same location as your resource group.

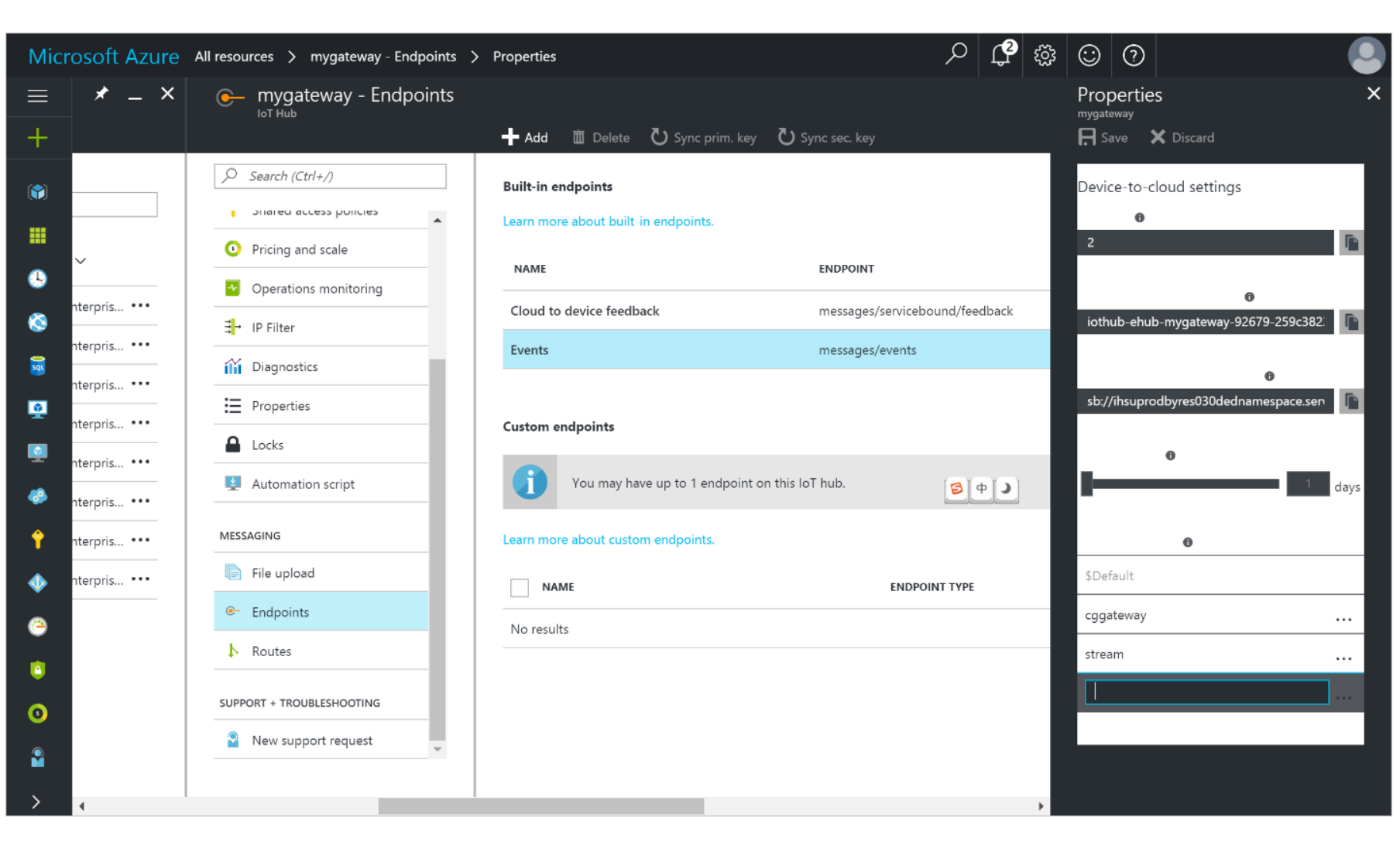


Add a consumer group to your IoT hub

Consumer groups are used by applications to pull data from Azure IoT Hub. In this tutorial, you create a consumer group to be used by a coming Azure service to read data from your IoT hub.

To add a consumer group to your IoT hub, follow these steps:

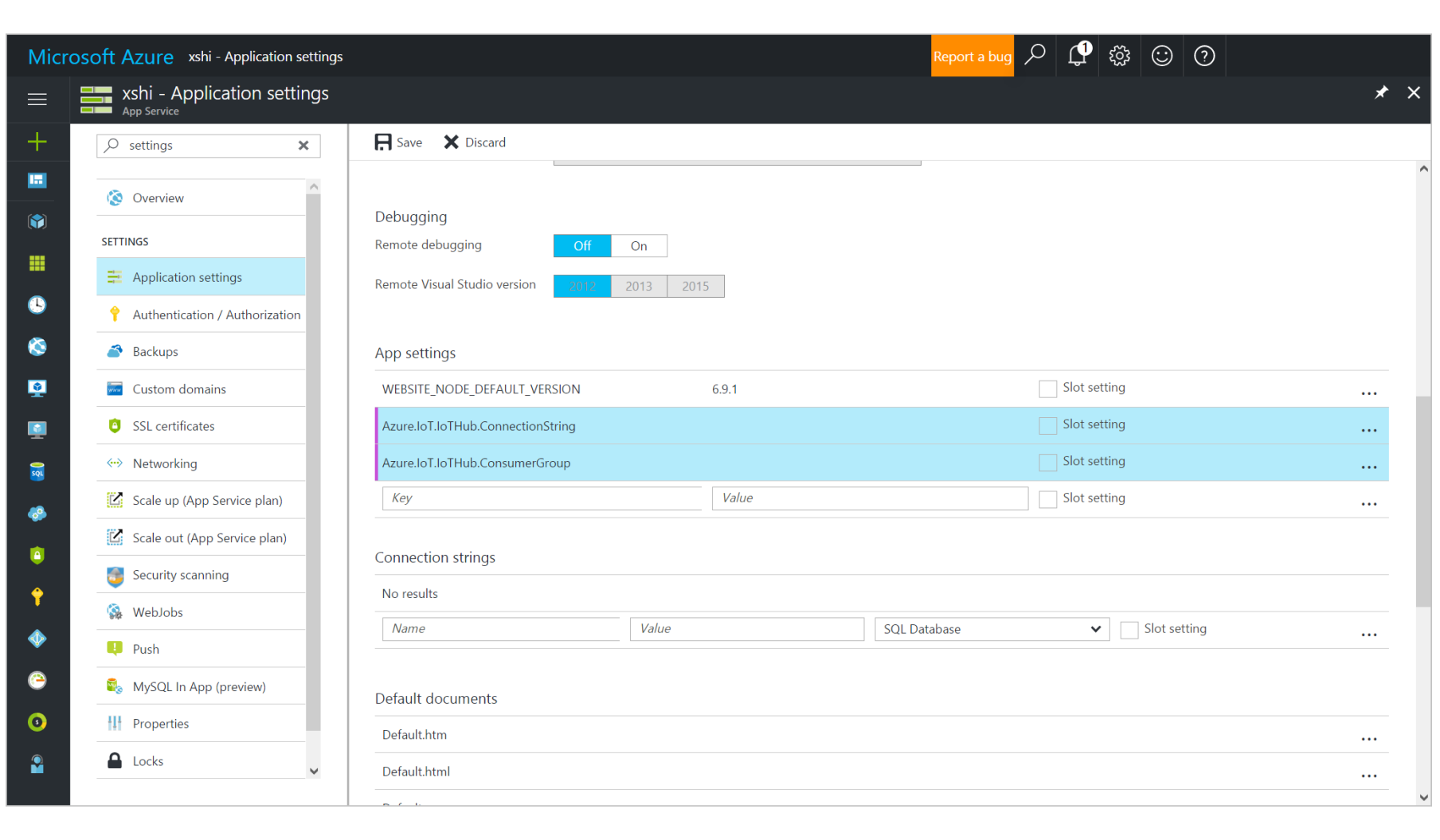
1. In the Azure portal, open your IoT hub.
2. In the left pane, click **Endpoints**, select **Events** on the middle pane, enter a name under **Consumer groups** on the right pane, and then click **Save**.



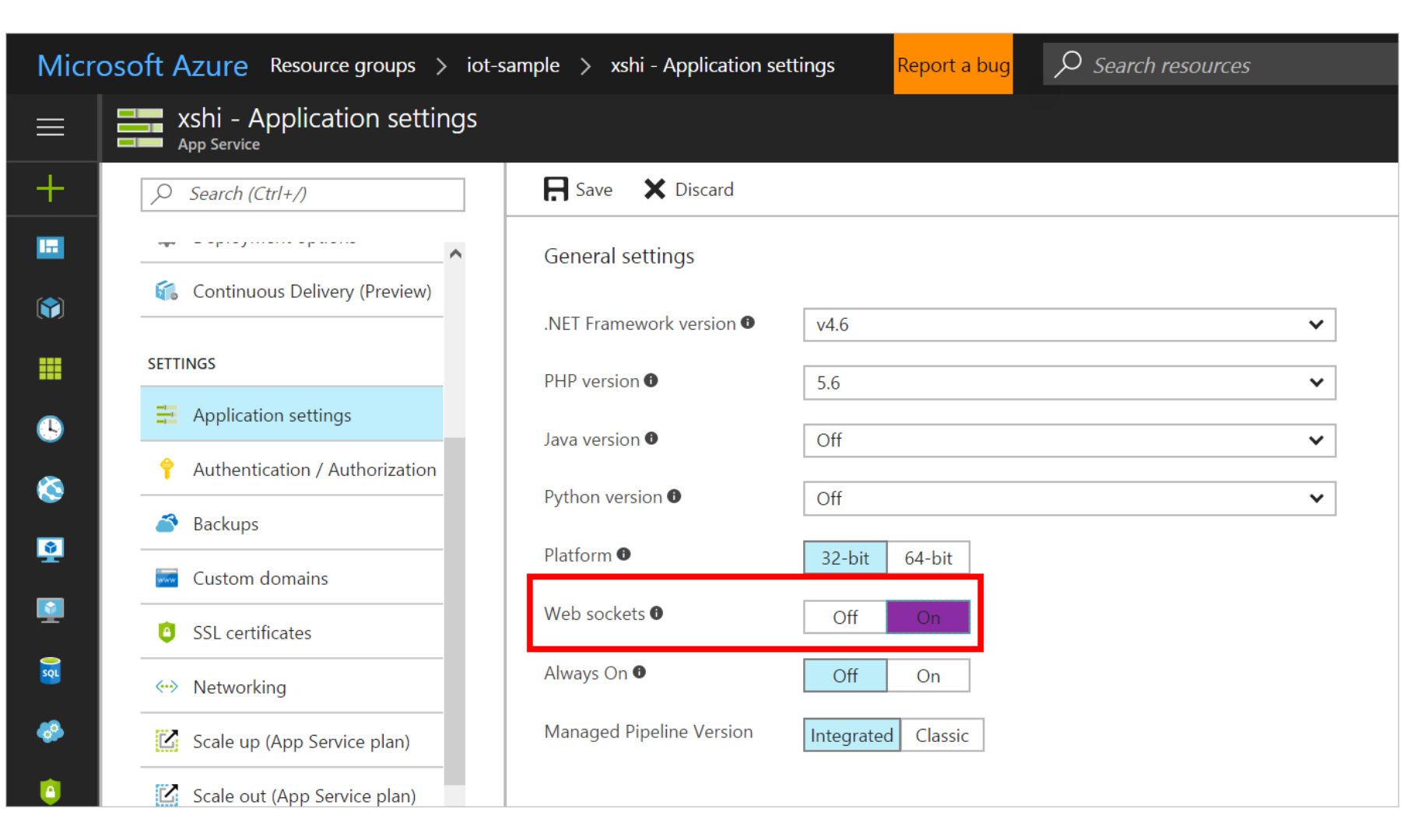
Configure the web app to read data from your IoT hub

1. Open the web app you've just provisioned.
2. Click **Application settings**, and then, under **App settings**, add the following key/value pairs:

Key	Value
Azure.IoT.IoTHub.ConnectionString	Obtained from Azure CLI
Azure.IoT.IoTHub.ConsumerGroup	The name of the consumer group that you add to your IoT hub
WEBSITE_NODE_DEFAULT_VERSION	8.9.4



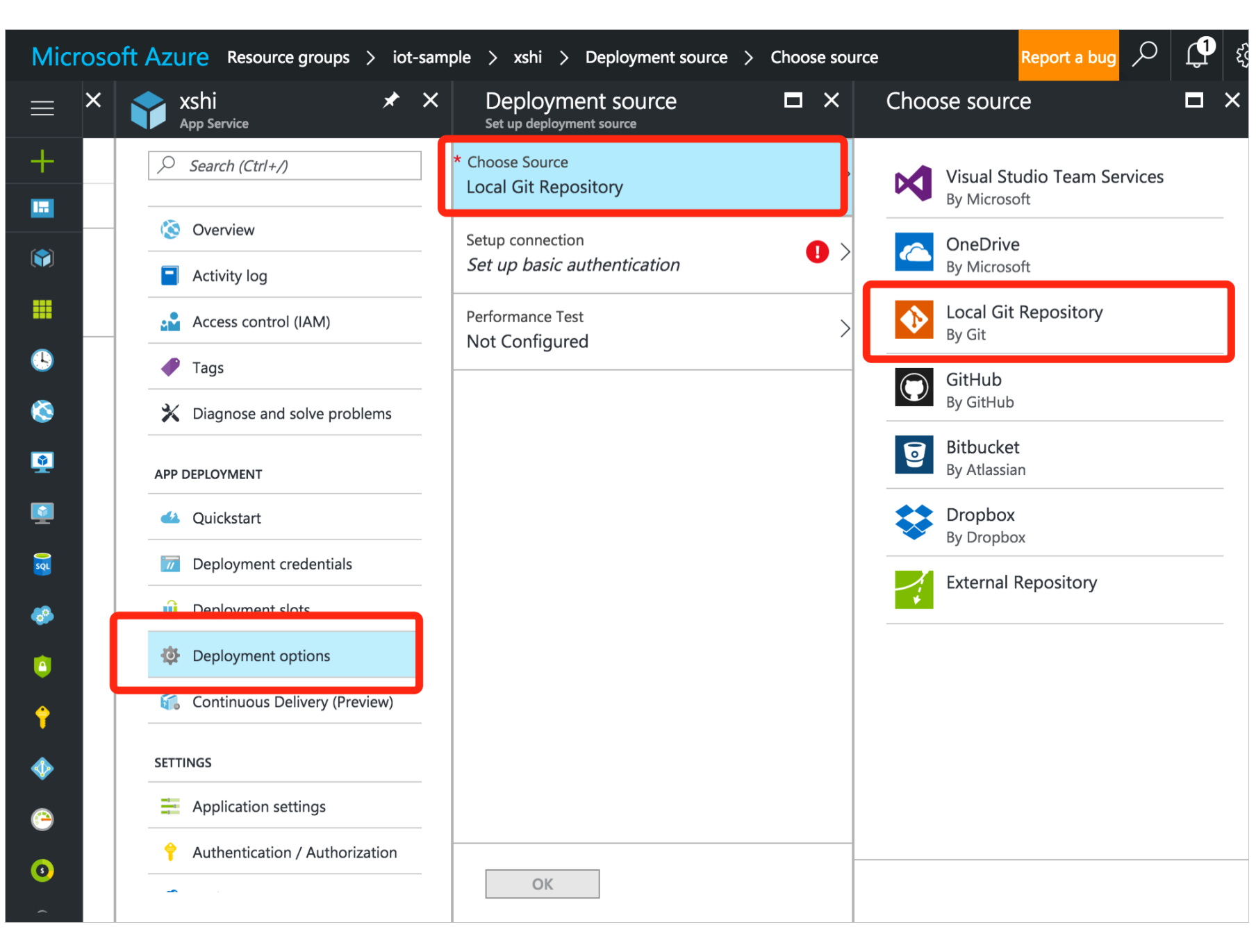
3. Click **Application settings**, under **General settings**, toggle the **Web sockets** option, and then click **Save**.



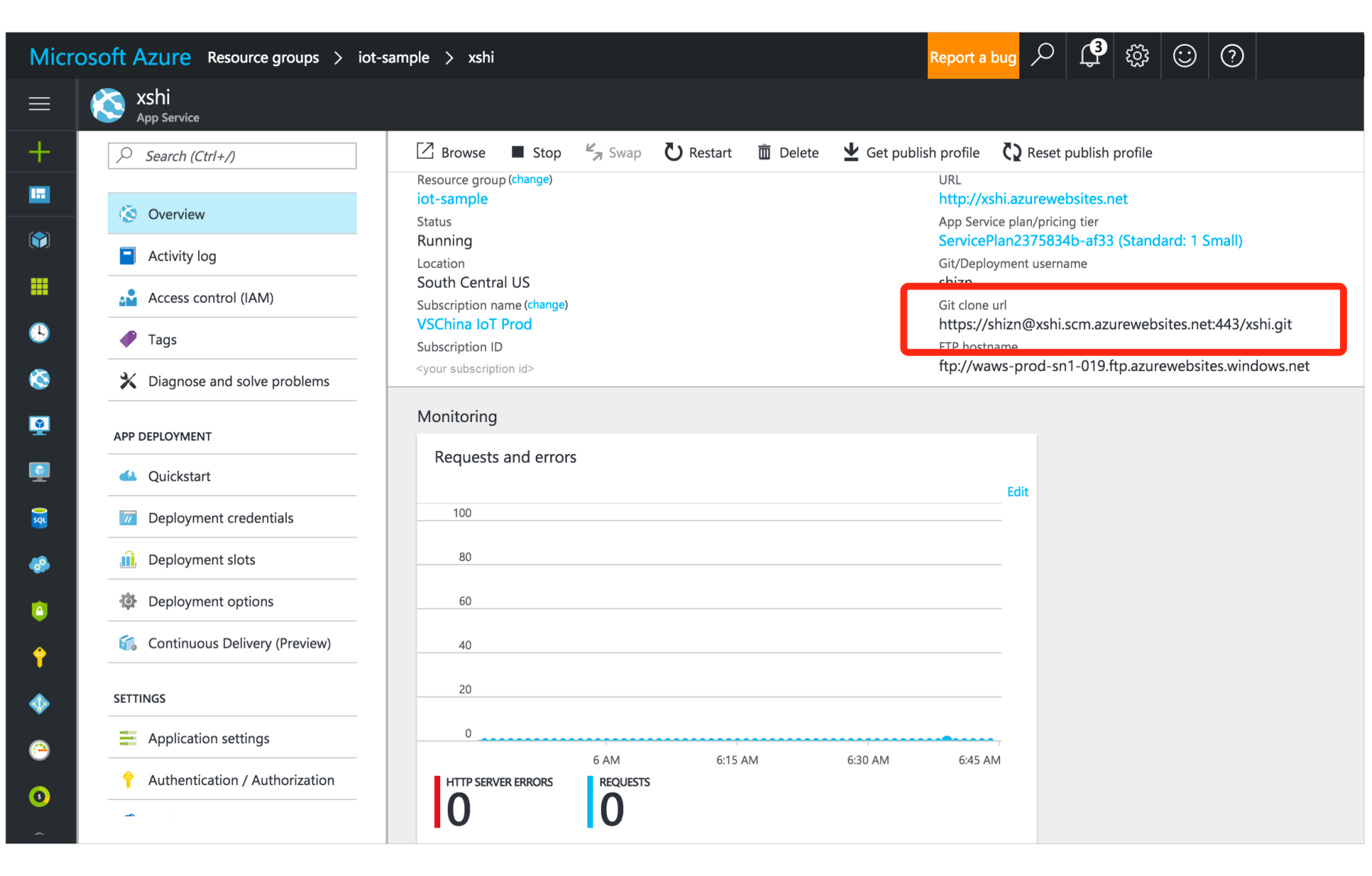
Upload a web application to be hosted by the web app

On GitHub, we've made available a web application that displays real-time sensor data from your IoT hub. All you need to do is configure the web app to work with a Git repository, download the web application from GitHub, and then upload it to Azure for the web app to host.

1. In the web app, click **Deployment Options** > **Choose Source** > **Local Git Repository**, and then click **OK**.



2. Click **Deployment Credentials**, create a user name and password to use to connect to the Git repository in Azure, and then click **Save**.
3. Click **Overview**, and note the value of **Git clone url**.



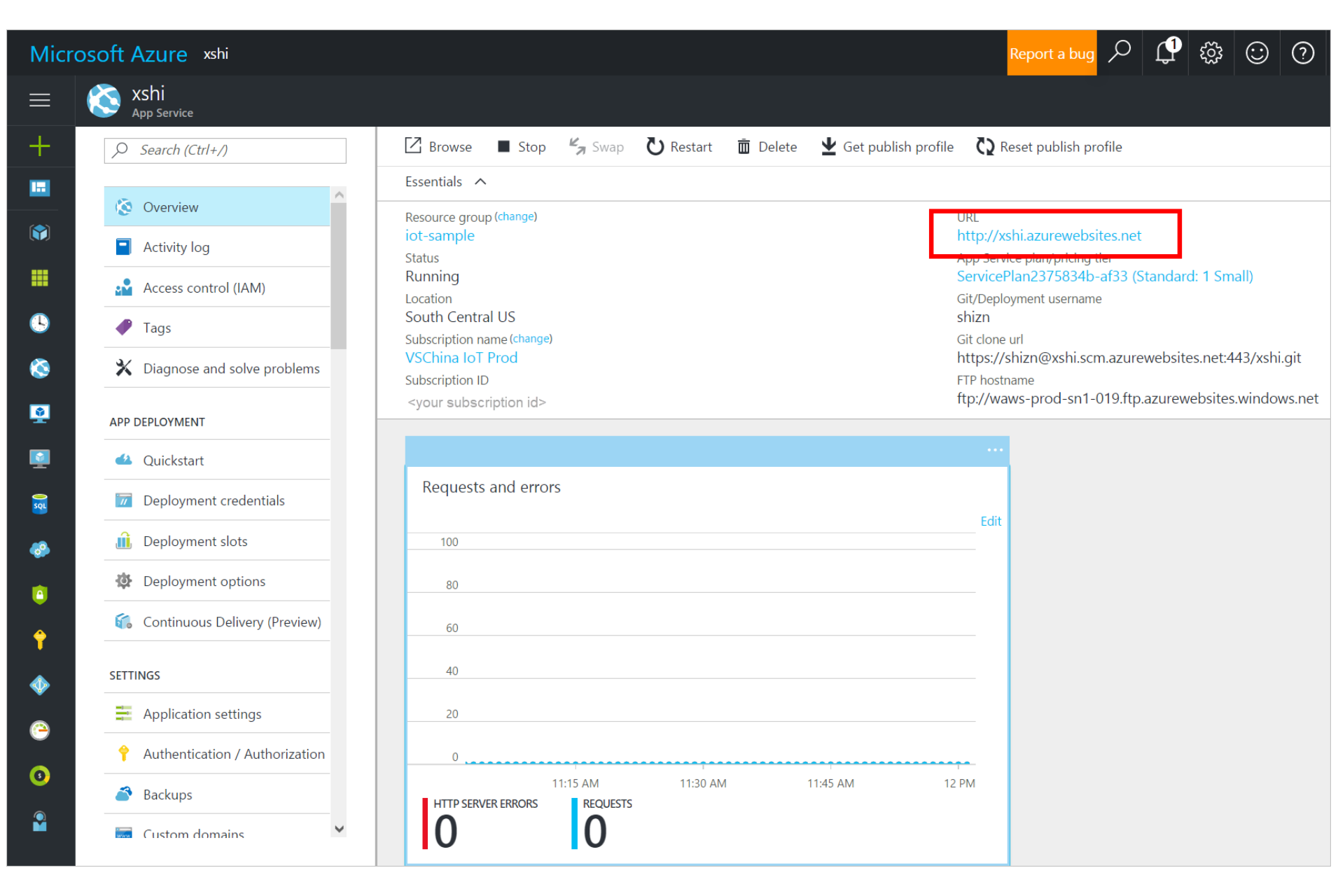
4. Open a command or terminal window on your local computer.
5. Download the web app from GitHub, and upload it to Azure for the web app to host. To do so, run the following commands:

```bash
git clone https://github.com/Azure-Samples/web-apps-node-iot-hub-data-visualization.g
cd web-apps-node-iot-hub-data-visualization
git remote add webapp <Git clone URL>
git push webapp master:master
```

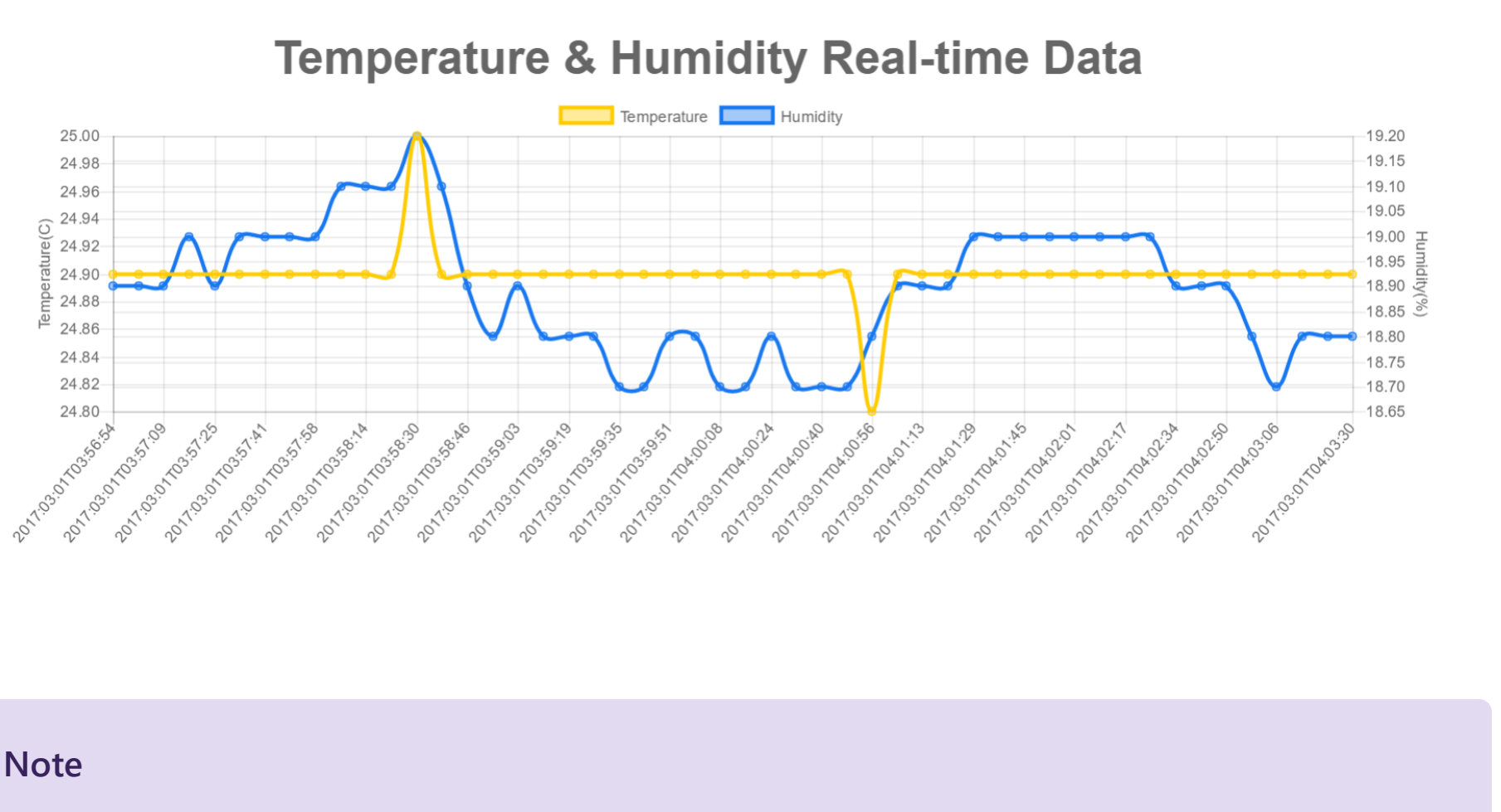
> Note
>
> <Git clone URL> is the URL of the Git repository found on the **Overview** page of the web app.

Open the web app to see real-time temperature and humidity data from your IoT hub

On the **Overview** page of your web app, click the URL to open the web app.



You should see the real-time temperature and humidity data from your IoT hub.



> Note
>
> Ensure the sample application is running on your device. If not, you will get a blank chart, you can refer to the tutorials under Setup your device.

Next steps

You've successfully used your web app to visualize real-time sensor data from your IoT hub.

For an alternative way to visualize data from Azure IoT Hub, see Use Power BI to visualize real-time sensor data from your IoT hub.

To continue to get started with Azure IoT Hub and to explore other IoT scenarios, see the following:

- Manage cloud device messaging with Azure IoT Toolkit extension for Visual Studio Code
- Save your Azure IoT hub messages to Azure data storage
- Use Power BI to visualize real-time sensor data from your IoT hub
- Use the Web Apps feature of Azure App Service to visualize real-time sensor data from your IoT hub
- Forecast weather by using the sensor data from your IoT hub in Azure Machine Learning
- Manage devices with Azure IoT Toolkit extension for Visual Studio Code
- Use Logic Apps for remote monitoring and notifications

Feedback

We'd love to hear your thoughts. Choose the type you'd like to provide:

Product feedback | Sign in to give documentation feedback

Our new feedback system is built on GitHub Issues. Read about this change in **our blog post**.

Loading feedback...