Lab: R in WSL

Oct 28th, 2018

Author: Elena Lowery elowery@us.ibm.com
Updated By: Sidney Phoon yfphoon@us.ibm.com
Updated By: Joel Patterson jpatter@us.ibm.com

IBM

Table of contents

Overview

In this lab you will work with various R assets in Watson Studio Local.

Required software, access, and files

- To complete this lab, you will need access to a Watson Studio Local cluster.

Part 1: Create a WSL Project and run R notebook and Shiny

1. Log in to a **Watson Studio Local cluster**.

2. If you have not already created the WSL_Demos project do so now.

3. Switch to the **Notebooks** tab in **Assets**. Open the *DriverClassification* notebook.

4. Work through the notebook.

   Note: notice that in the notebook we save the model into the RStudio directory. We use this approach because when we open RStudio, the model is displayed in the UI, which makes it easier for integration with the Shiny application.

   In the sample notebook we overwrite the file each time we run the "save code", but additional code can be added to add a version number to the model name.

5. Open **RStudio** from the Project view.

6. If *serverR* file is not already loaded in the top left window, click on the *demoBreakEvents* folder in the file explorer (right bottom window), then click on *serverR*.

   After the application is loaded, click the **Run App** button.
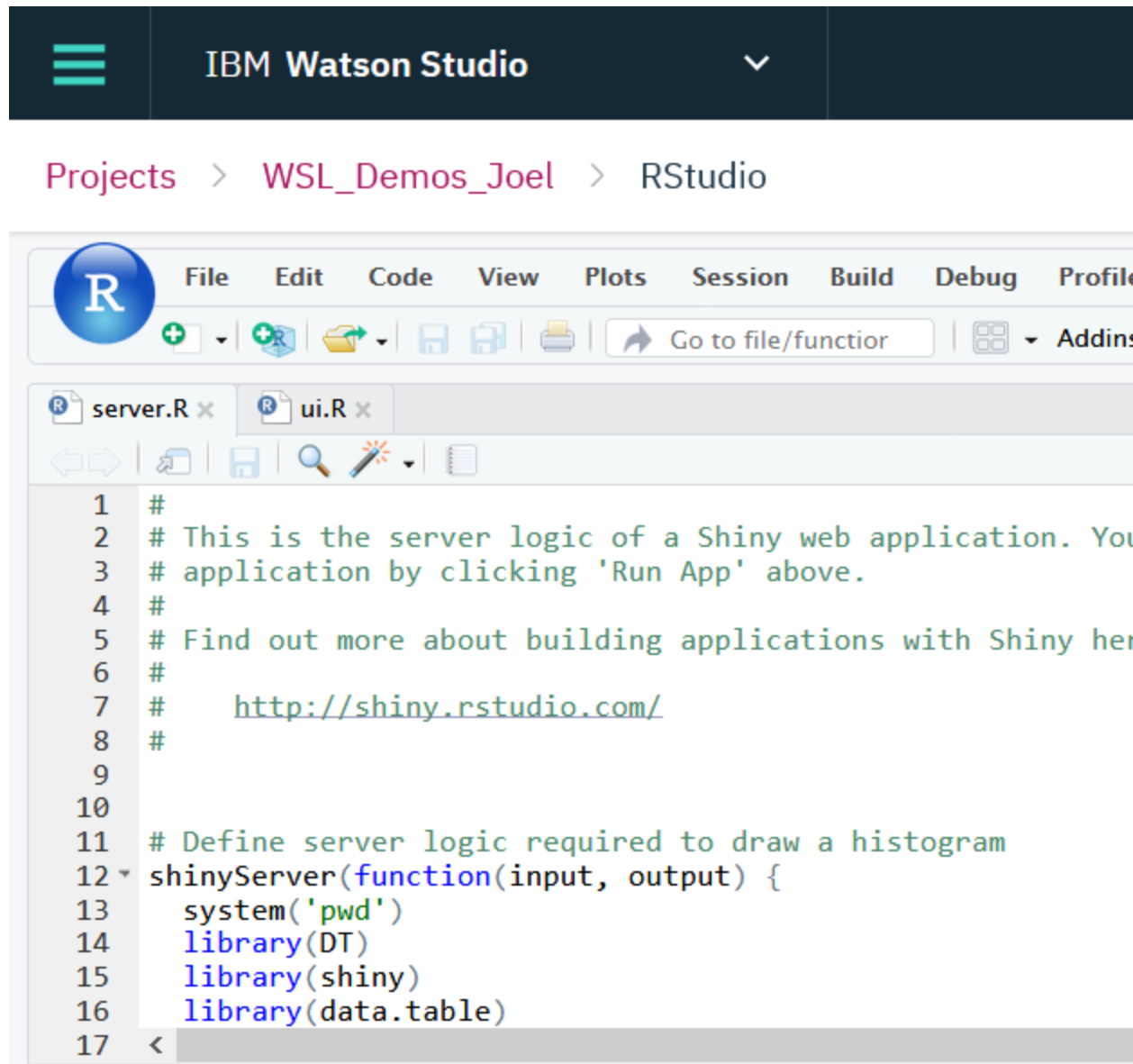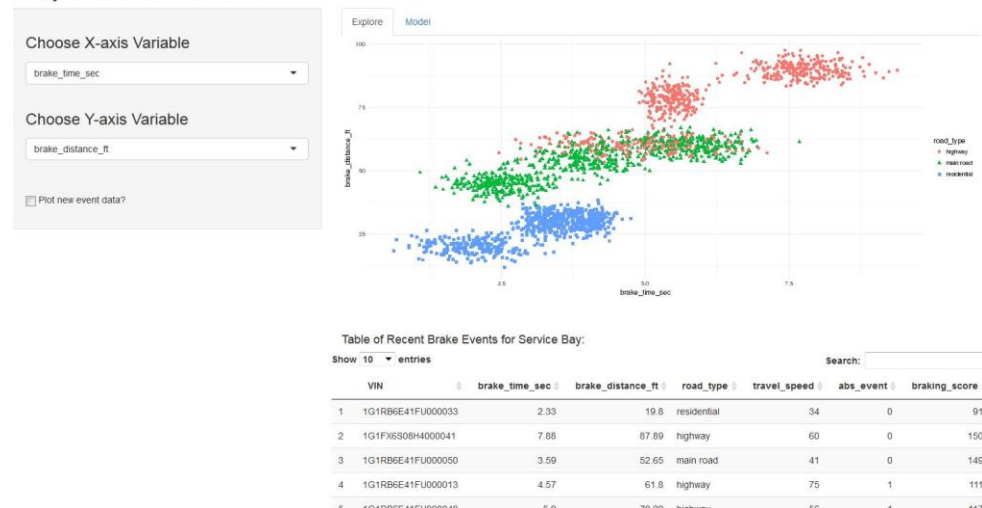
7. **Allow popups** or select **Try Again** to bring up the application if it fails to display.
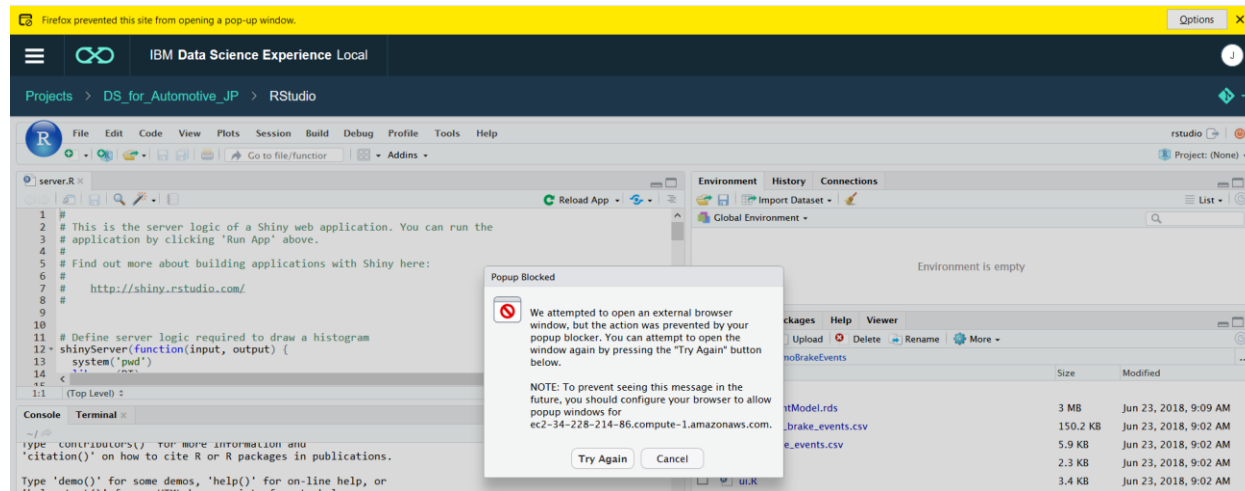
8. Test the application - both the Explore and Model tabs.

Part 2: Publish Shiny

In this section we will publish the Shiny application.

1. Navigate back to the **Project->RStudio** view and click **Publish** from the ellipses next to the Shiny application (*demoBreakEvents*) in **RStudio**. Note: you may have to **view all** in order to see the application (it be will of type SHINY)

RStudio view all (8)		
NAME	**TYPE**	**LA**
📖 spark_mtcars	R	01
📖 spark_kernel_basic	R	01
📖 spark_flights	R	01
📖 readme	TXT	01
📖 demoBrakeEvents	SHINY	01

2. If you get an error during publish, check with the lab instructor.

3. Provide *Published name* (make it unique, for example, add your initials) and select *Published content visibility* that you would like to test. Save the permalink to the Shiny app. Click **Publish**.

   Note: All Shiny applications in WSL are published to a shared Shiny server. At this time there is only 1 instance of the Shiny server in WSL (i.e. it's not configured for HA). However, because the Shiny server is deployed as a pod, Kubernetes will monitor its status and restart it, when needed.

Shiny app name
demoBrakeEvents

Published name *
ShinyApp_EL

Description

Published content visibility

○ Anyone with the link
● Any authenticated user
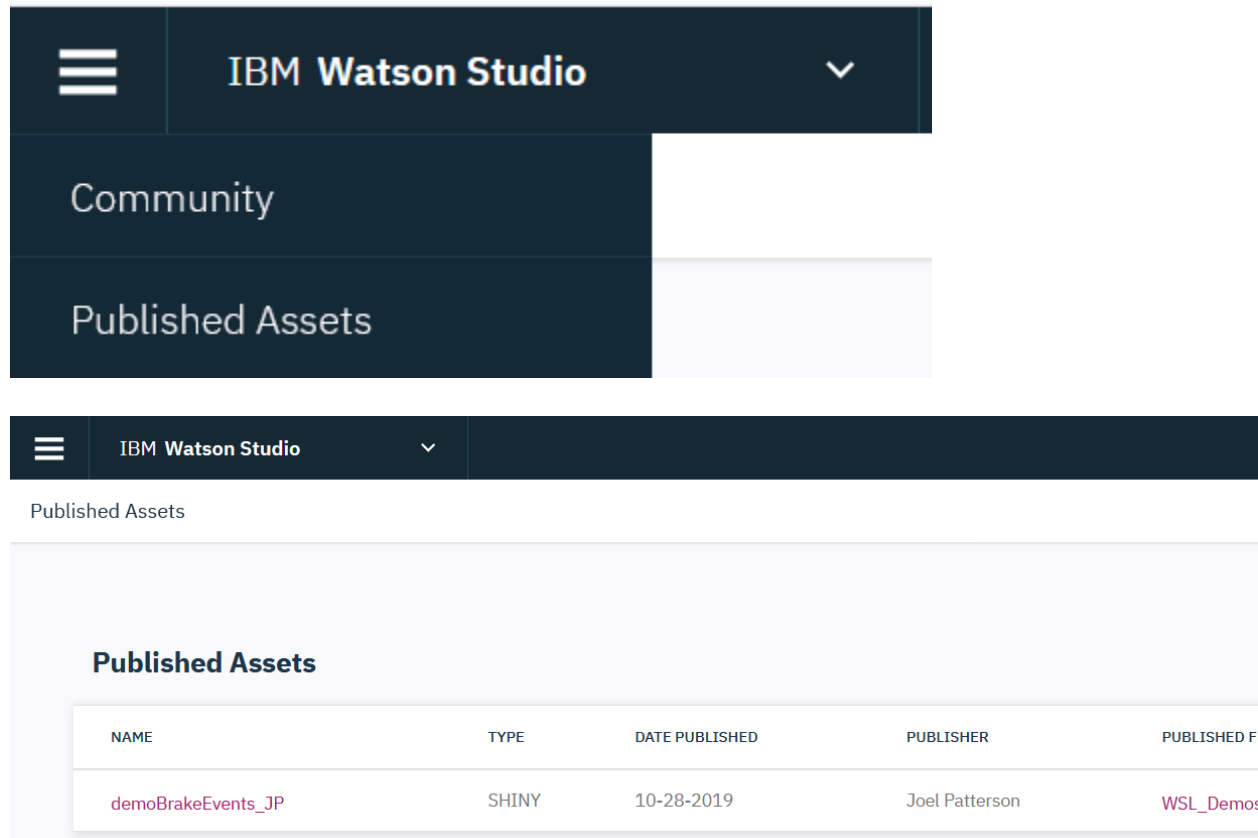○ Restricted to collaborators in the selected project

DSX_Local_Workshop ⌄

Permalink to read-only published shiny app

https://9.30.230.119/#/publishedAssets/view/dsx-publish-view-shiny-all-auth%2F%5E%5Eall_auth%2Frstudio%2FShinyApp_EL%2F

4. Navigate to **Published Assets** from the main WSL menu.

5. Verify that the application works.

6. If published globally make sure the link works (try this by starting a different browser and trying the link there).

Part 3: Check Status of Shiny Server (optional)

At this time the Shiny Server is deployed as a pod in WSL, and the only way to view status is by looking at the status of the pod. You must have admin privileges to perform these functions.

There is only 1 instance of Shiny Server, but Kubernetes monitors pod status, and the pod should be automatically restarted if it goes down.

1. Switch to the **Admin Console** view

2. Click on Pods. Scroll down to locate the *r-publish… pod*. This is the Shiny Server.

r-publish-content-3361764361-dnphb • ✅ dsxl-jan07-0-storage-1.fyre.ibm.com

3. If the pod is not running, you can redeploy it from this UI or via Kubernetes commands in ssh.

   Display all pods: `kubectl get pods --all-namespaces`

   Delete pod: `kubectl delete pod <pod name> -n <namespace>`

   Force delete for a pod (if it's stuck in "terminating" state):
   `kubectl delete pod <pod name> -n <namespace> --grace-period=0`