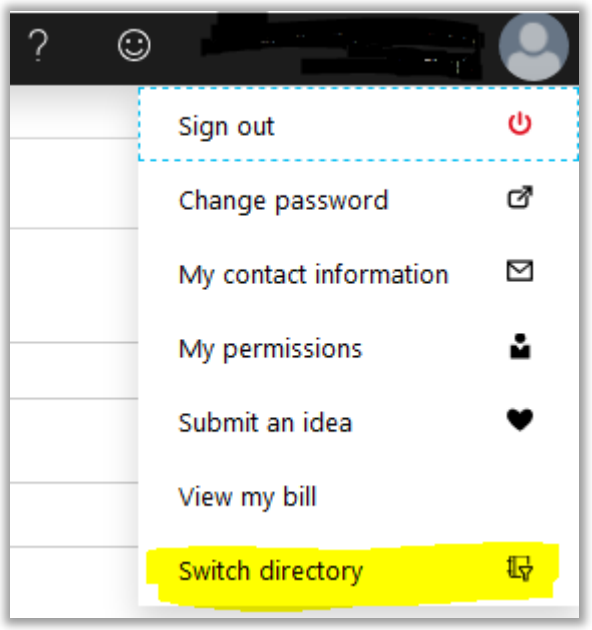
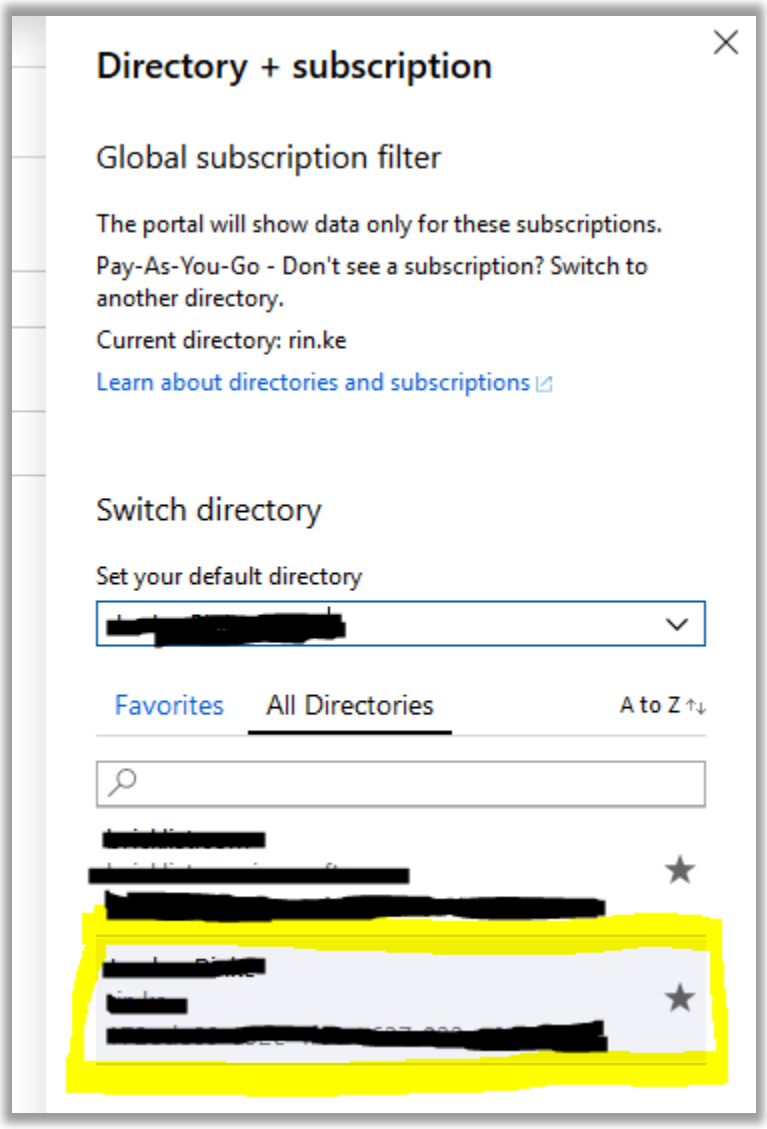
Lab 1: Building your first Azure Function

Log into your Azure account. Your instructor will either give you an email and password to use, or will have added your account as a guest account to the student/training demo account.

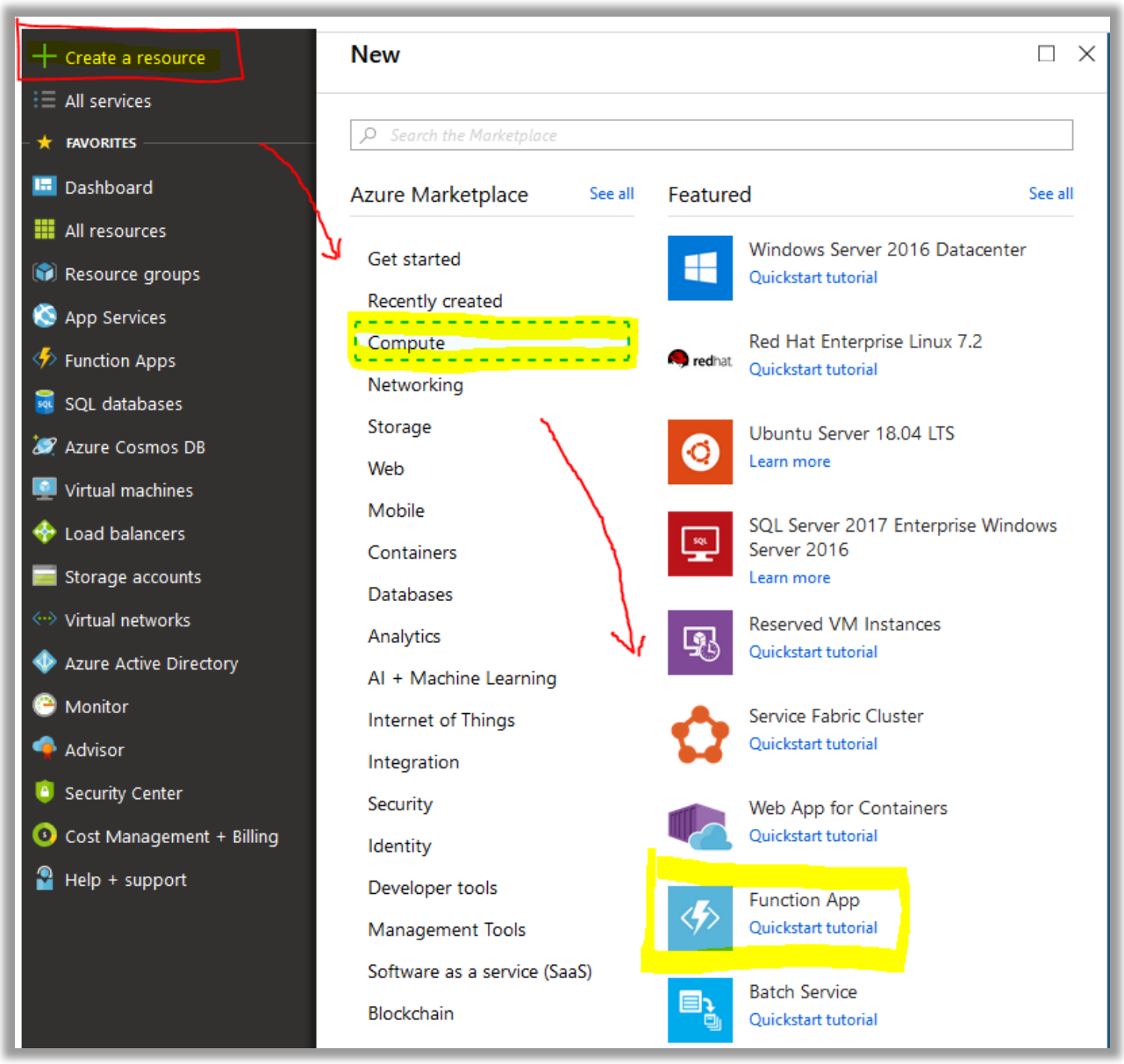
Once you log in you will need to be sure to switch to the subscription that matches the class, the instructor will have this information. Click on your name in the upper right hand corner and select “Switch Directory”:



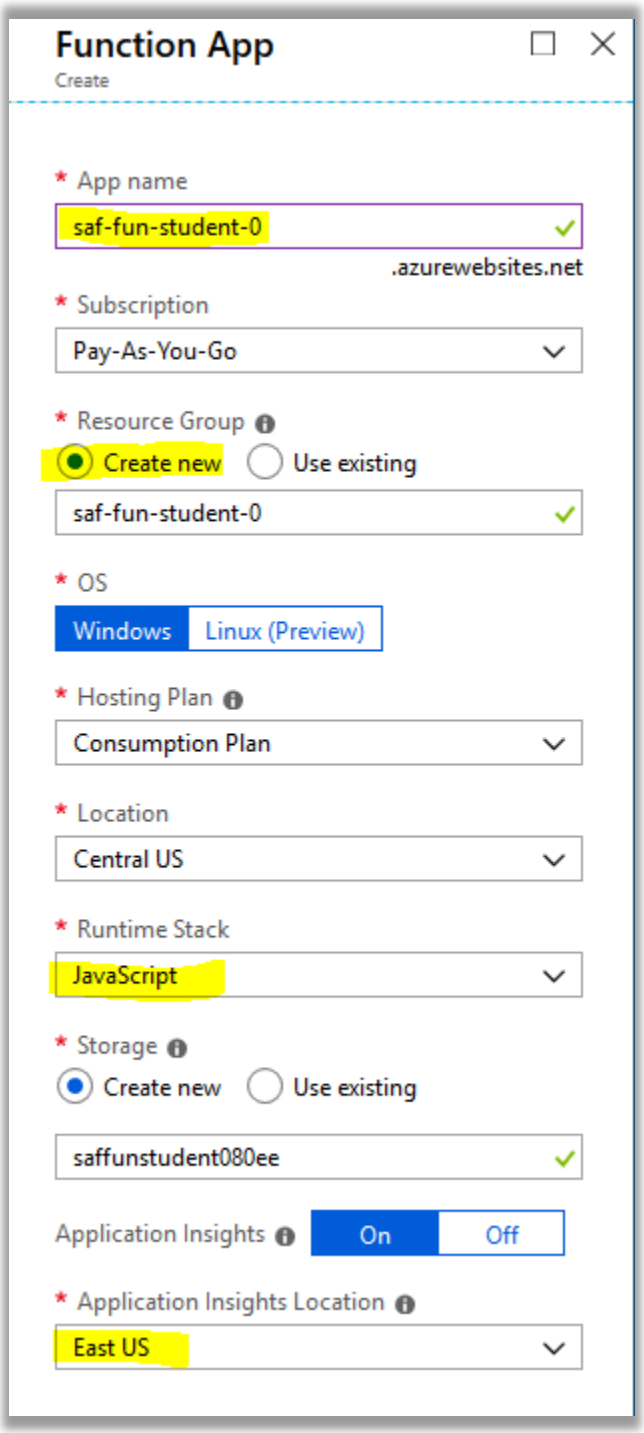
Click on the directory option that matches the one your instructor has given you for this class:



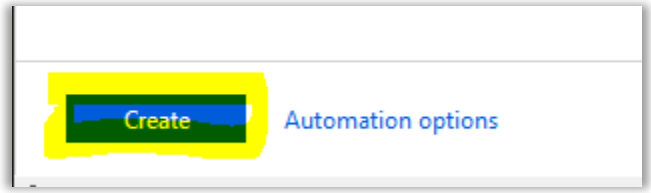
On the left hand side of the portal click on “Create Resource” then click on “Computer” and select “Function App” Be sure to click on the words function app and not the quick start guide by accident:



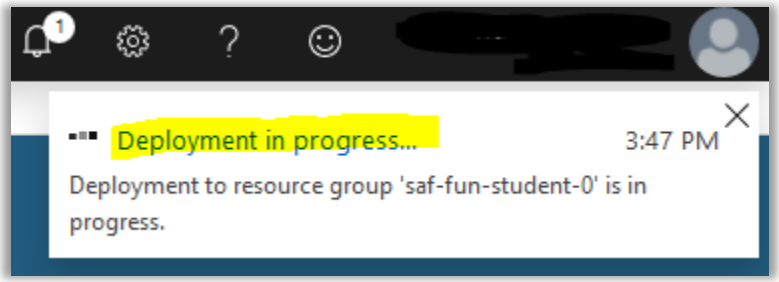
Give you app a unique name that matches your student number such as “saf-func-student-X”. Select the option to create a new resource group. Change the runtime stack to JavasScript and make sure the application insights location is in your continent or as close as possible.



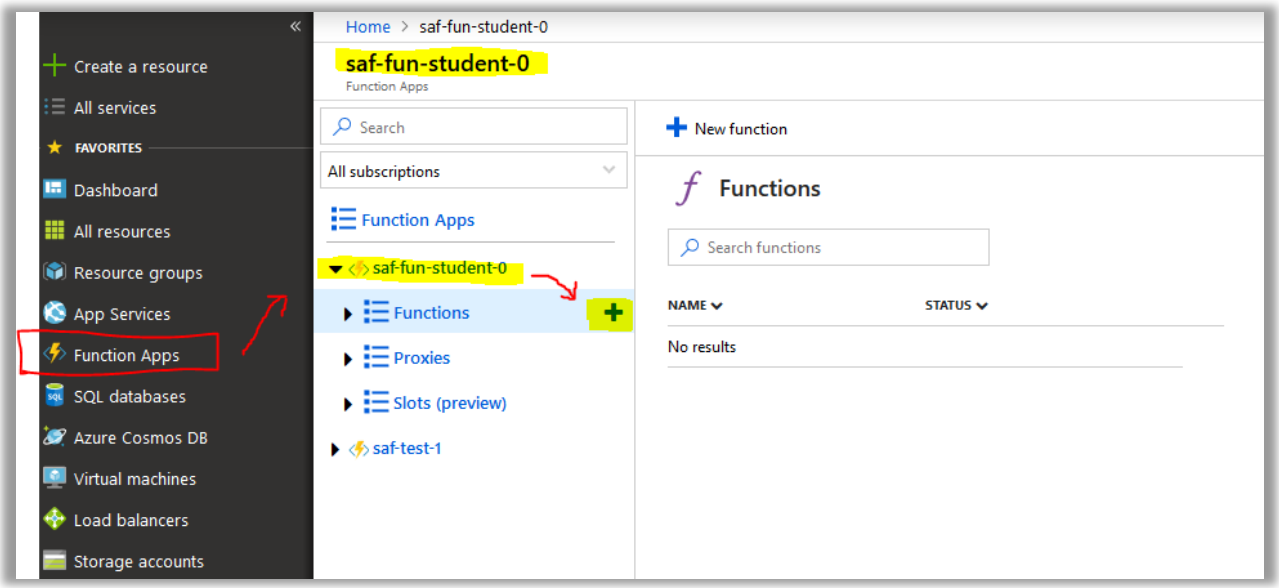
Create the function:



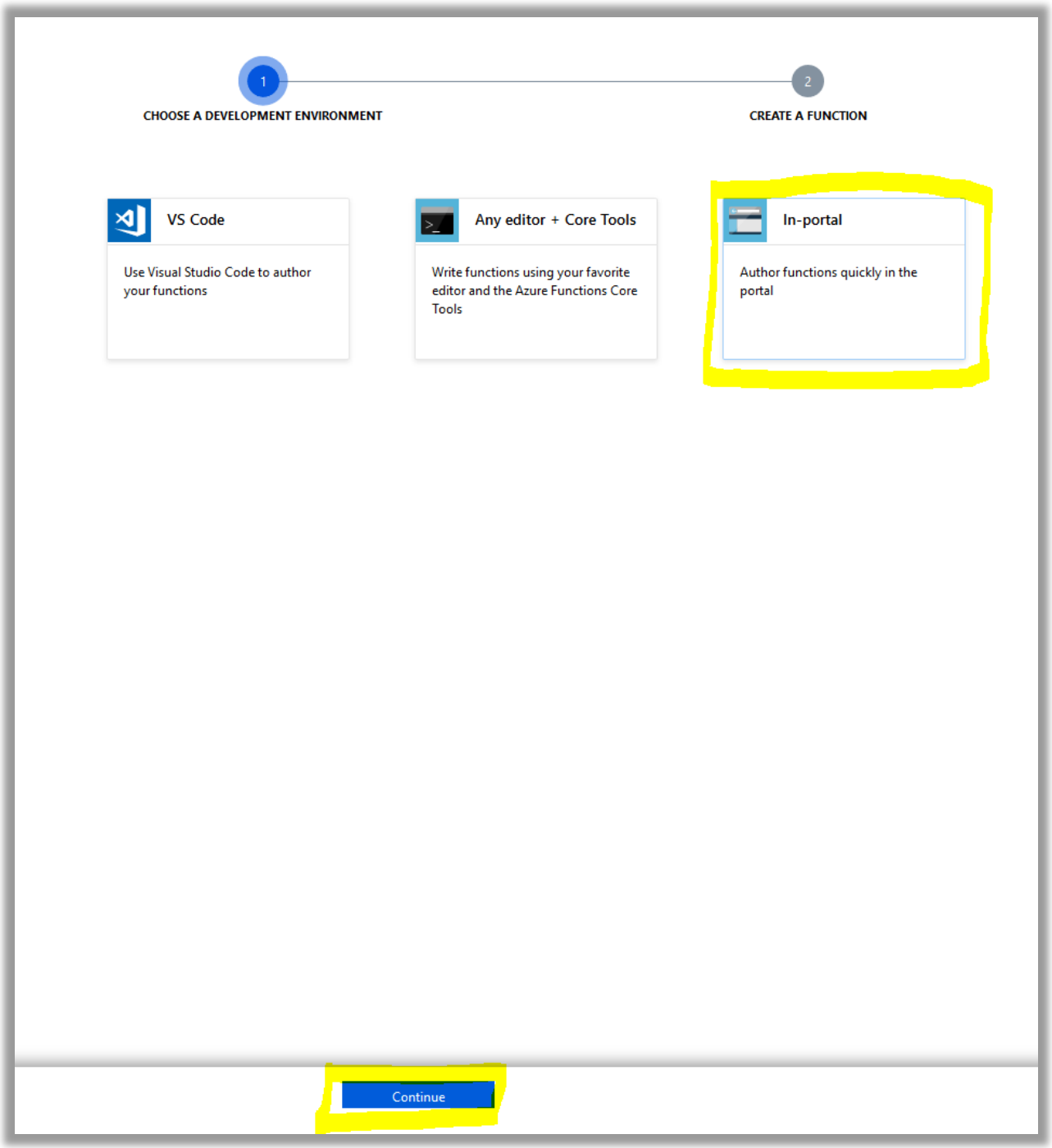
You will see a pop up indicating the function is being deployed, this will take 1-3 minutes.



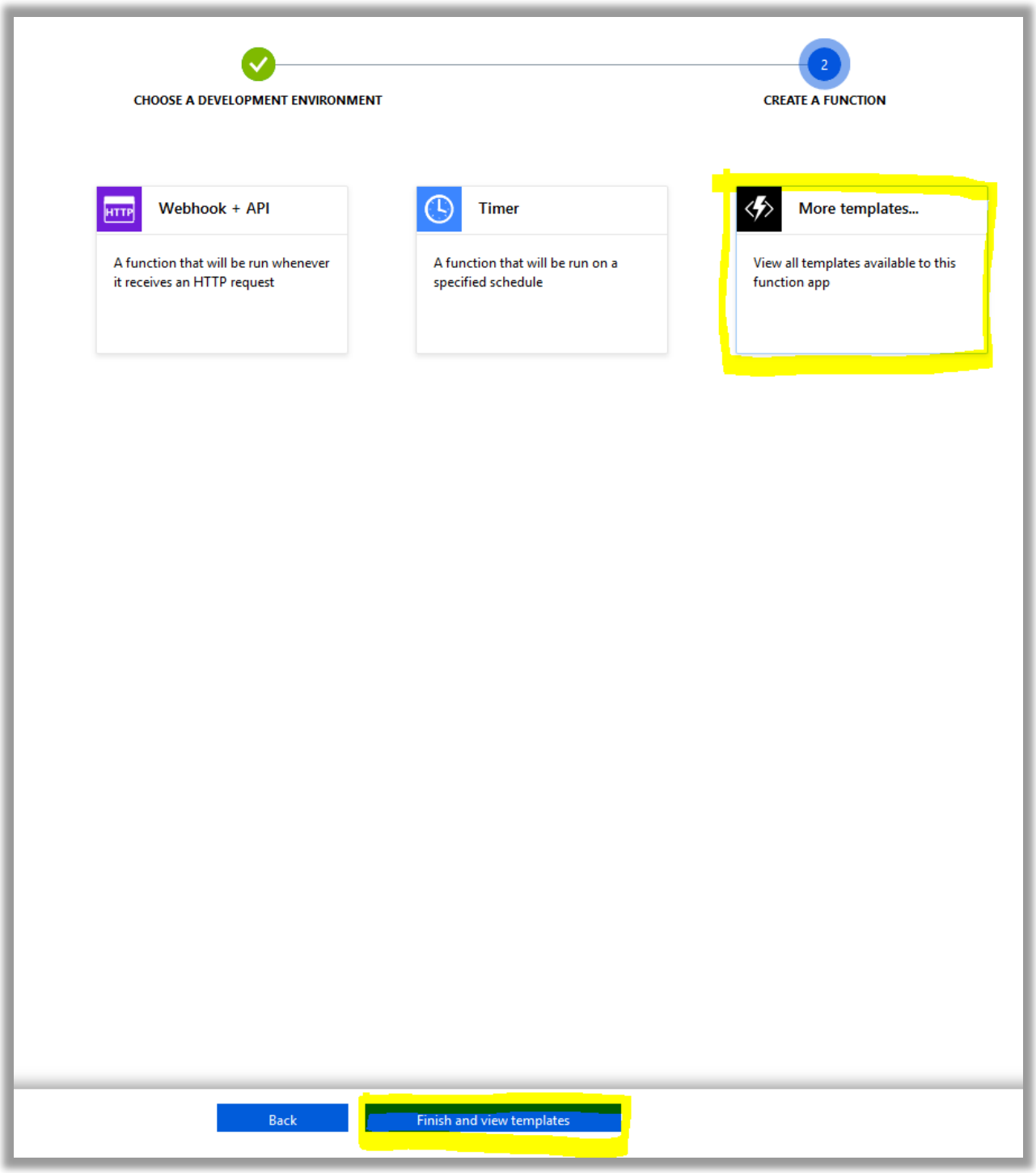
Once it is deployed click on Function Apps, select the app name that you created in the previous step and click on the plus sign to create a new function:



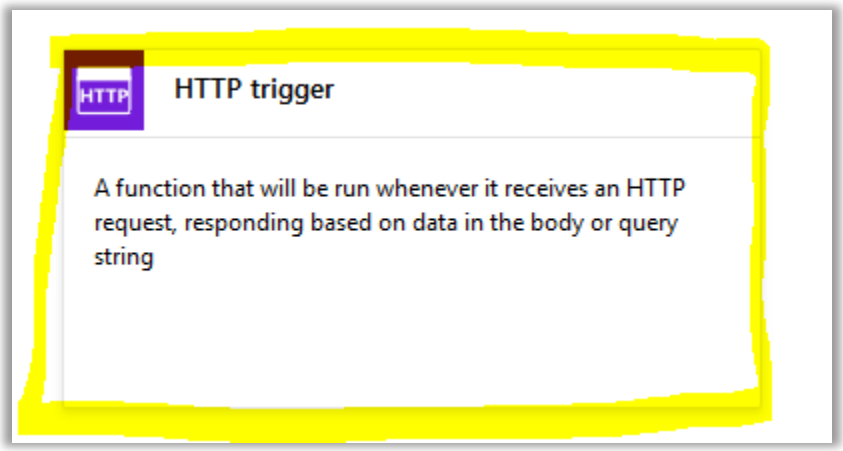
There are a number of options here but for ease of use for this class select “In-Portal” in production you would likely use VS Code or the Core Tools, click “Continue”:



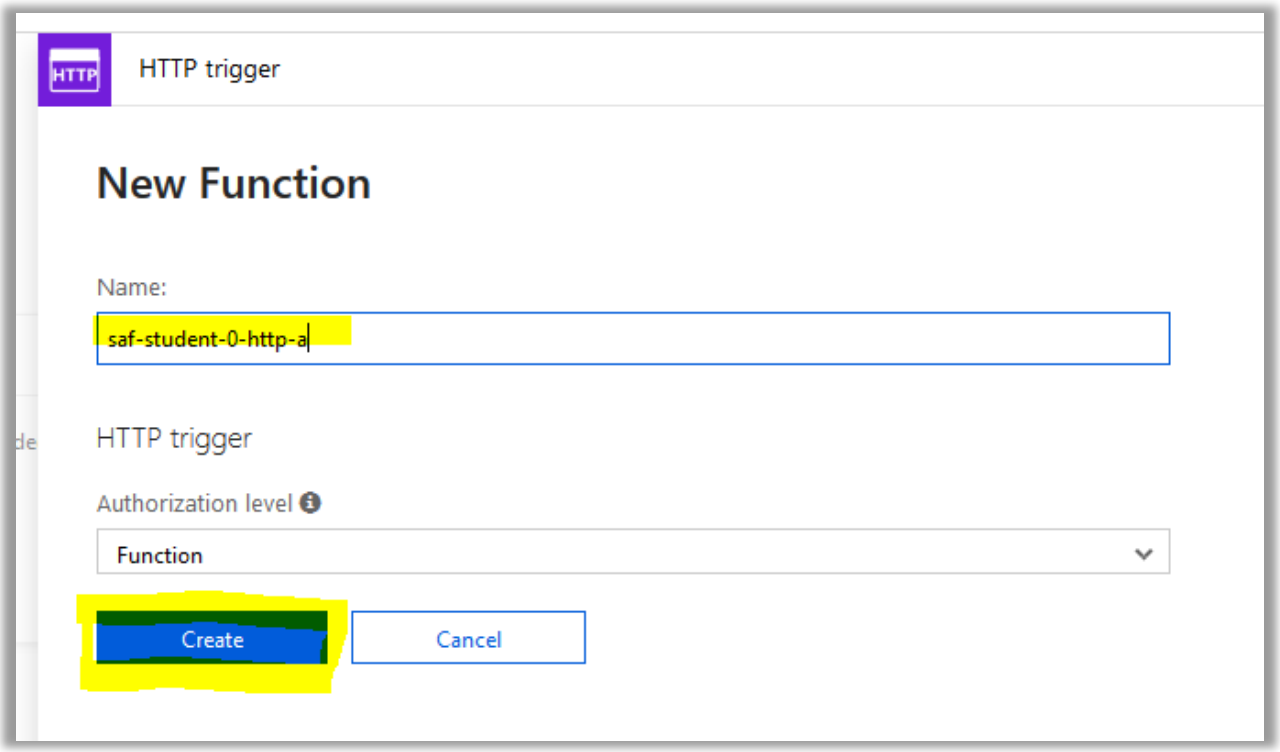
Select “More templates…” from the right hand side and then click “Finish and view templates” to see the options that we will need to use for the rest of the labs:



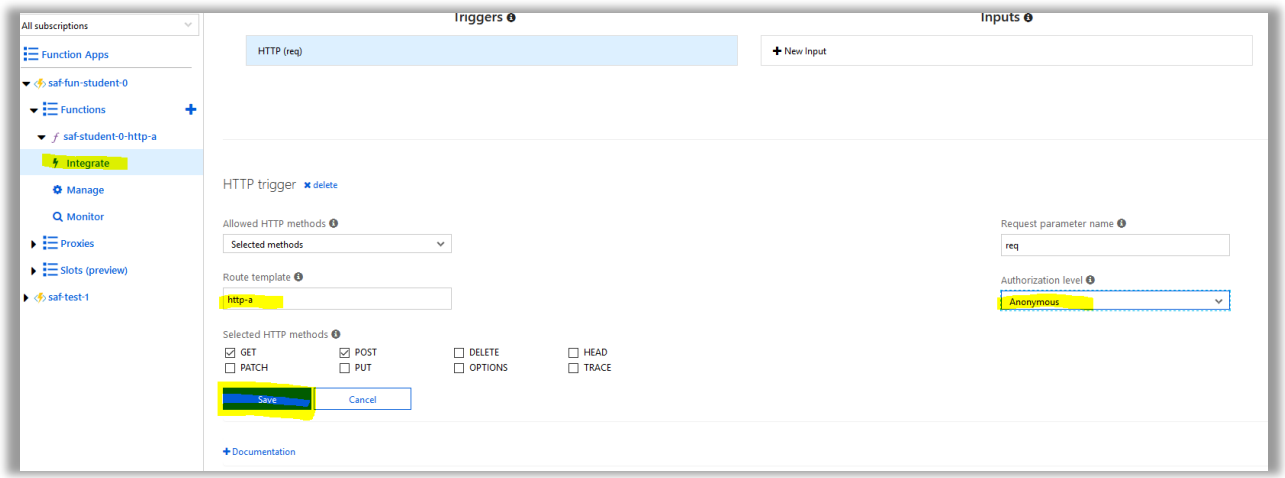
Select the HPPT Trigger option from the templates that are displayed:



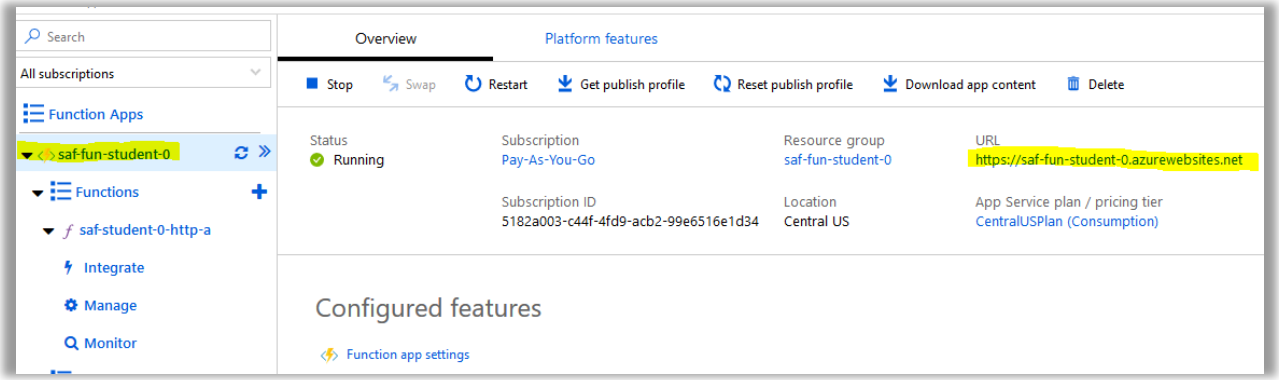
Give your function a meaningful name that matches your student number. Also differentiate this as a specific endpoint. Either #1, or A, we will be using this to create a canary later so we need two version that we know go together:



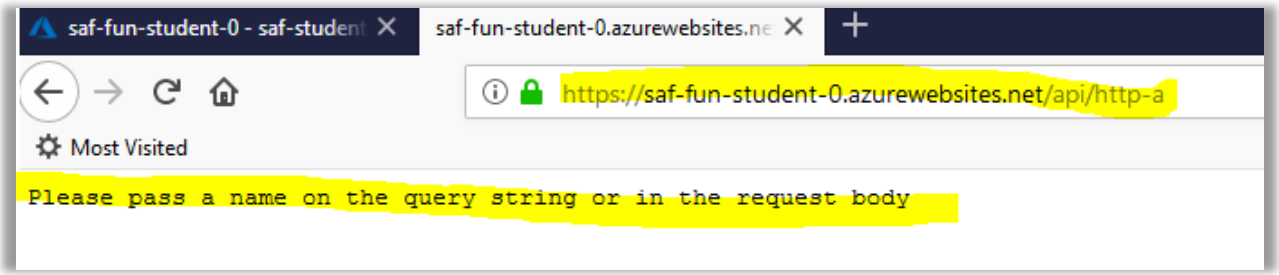
Click on the new function name in the Function Apps menu, select “Integrate” enter a router template – this will be the end URL that we hit to access this web service. Unlike AWS Azure Functions are connected to an external endpoint automatically, you can use them without attaching them to an API Gateway first. Change the authorization to “Anonymous” as well here so we can access this from any browser without requiring a token:



Click on the name of the function you created and look at the overview, this will show you the base URL for your function application. Each functions “Route” template will be added as a url on to the end of this:



If you combine the URL and the route you should see something similar to the below screen shot. Be sure to substitute your URL and the route you set:



The HTTP trigger template takes a variable called “Name” in order to display a name as a query string, add that to the query to see it respond appropriately: