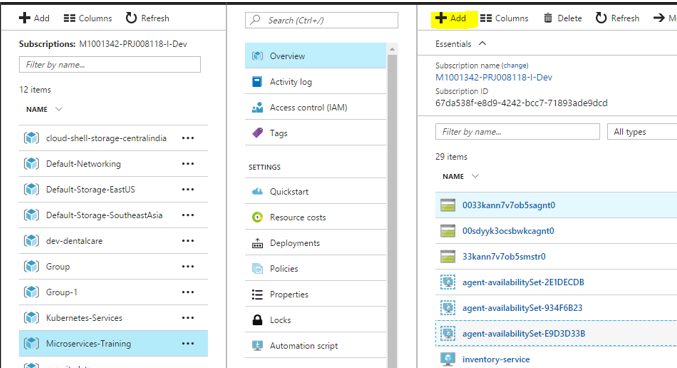
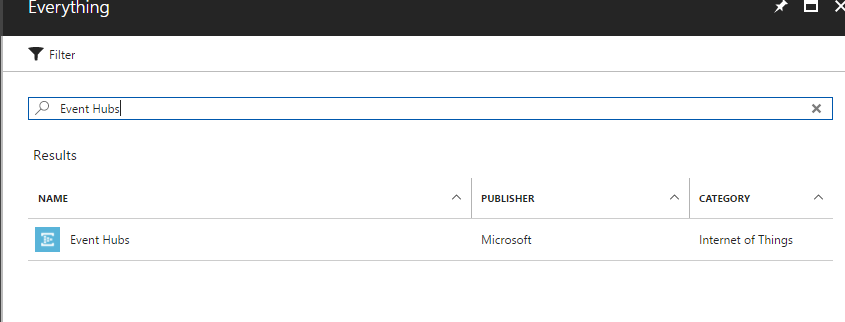
# Create the BuYnow APi

## Create eventhub

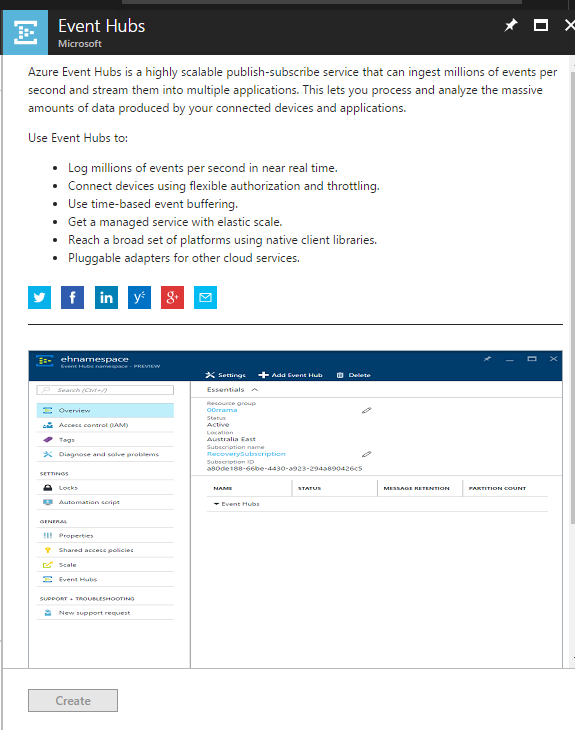
1. Login to Azure Portal; navigate to respective resource group. Click on Add.



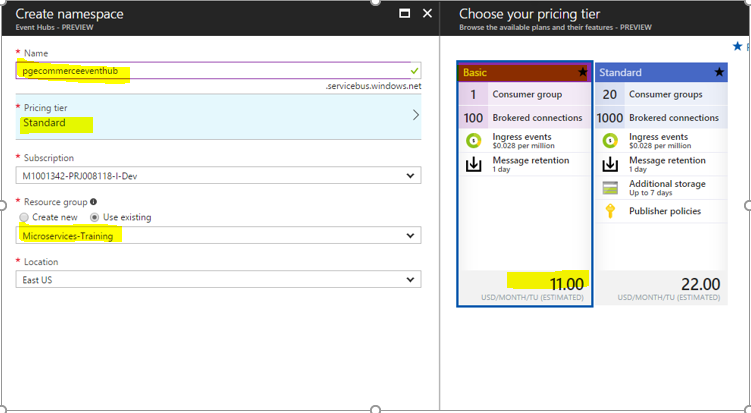
1. Search for ‘Event Hubs’ in the search text box.



1. Select the above Event Hubs service and click ‘Create’ button.

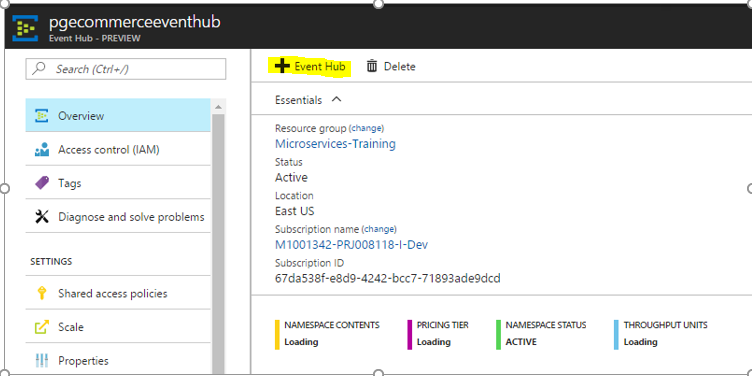


1. Enter the below highlighted details and click on ‘Create’.

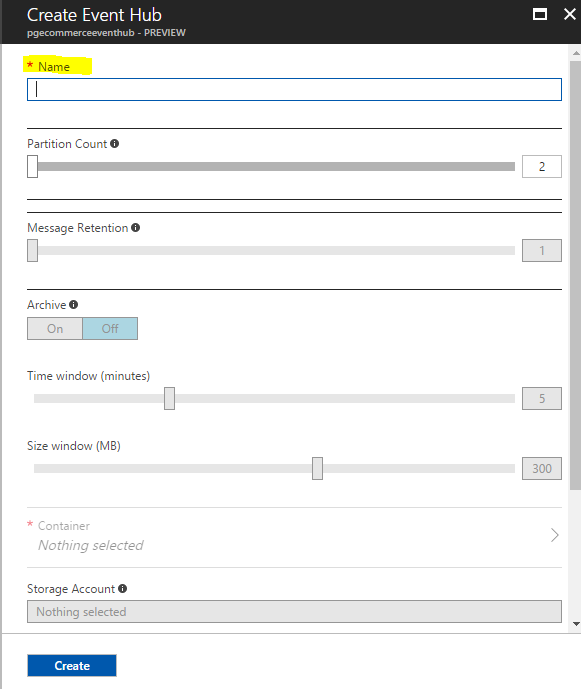


Note: Once deployment succeeded, click on the notification from portal for event-hub to navigate to below screen.

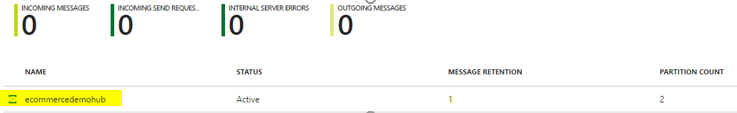
1. Once successfully created, will get below screen. Click on ‘Event Hub’ button to create event hub.



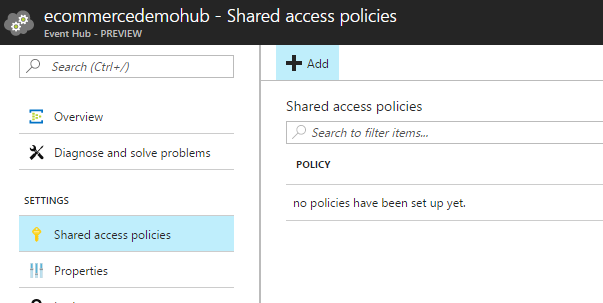
1. Enter ‘Name’ as ‘ecommercedemohub’ and click ‘Create’.



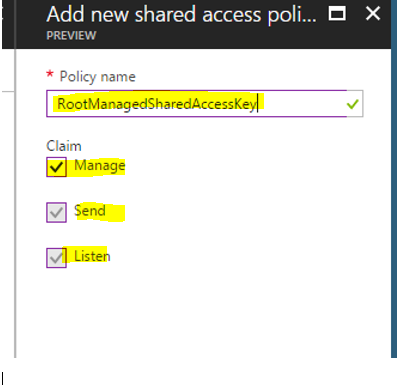
1. Click on created event hub from below screen.



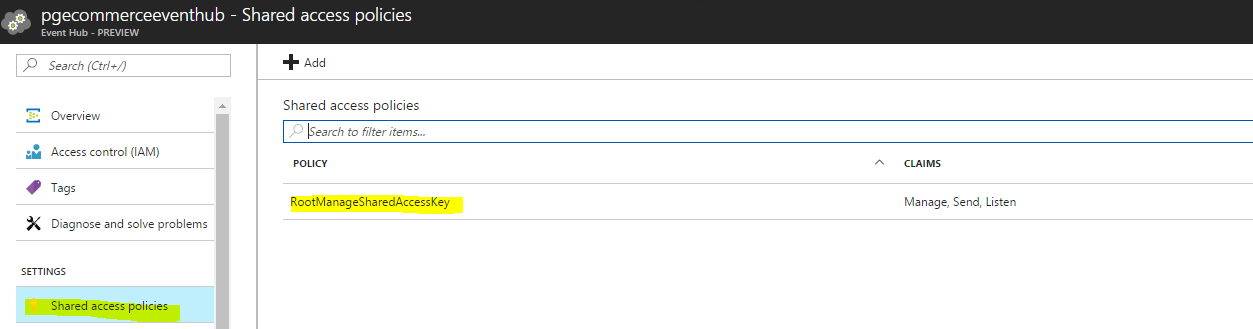
1. Add Shared access policies by following below screens. Click on Add to create policy.

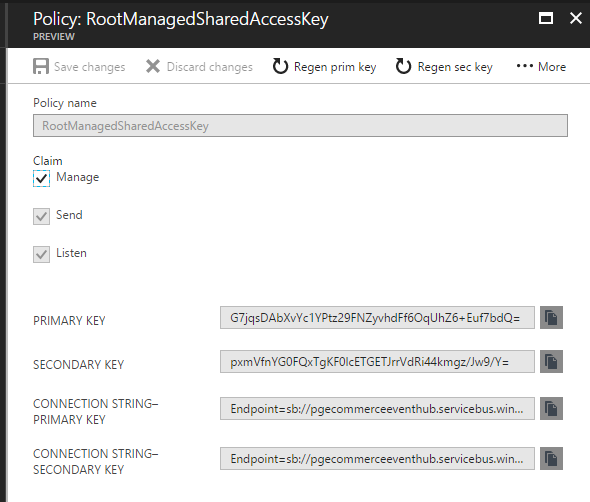


1. Click ‘Create’ in below screen.



1. From below screen navigate to Share Access Policies:





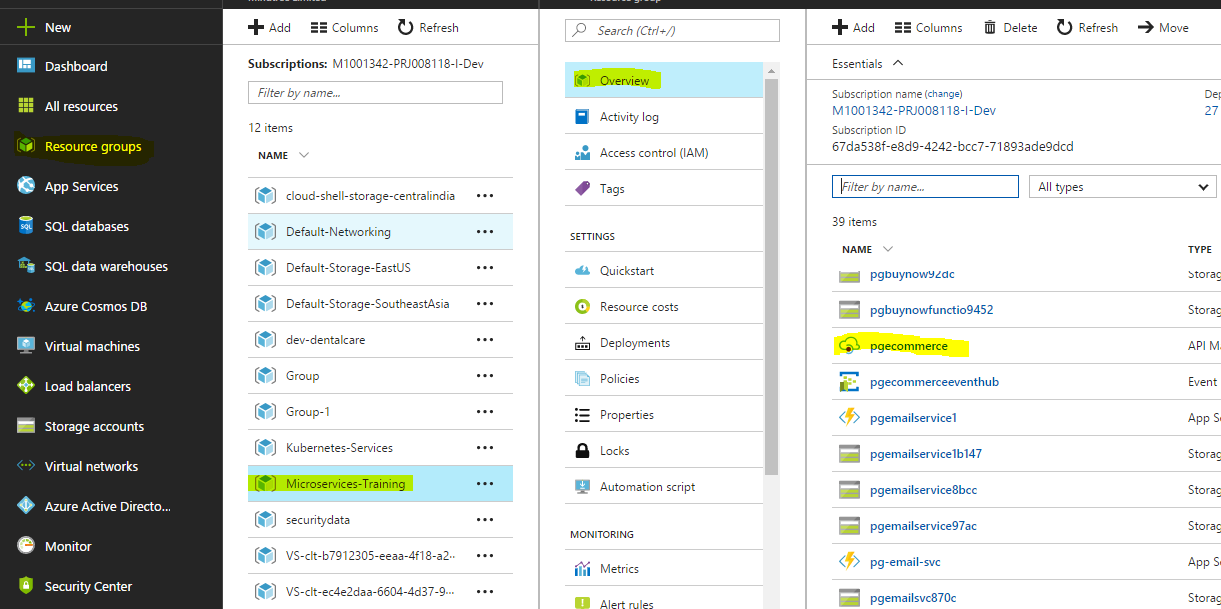
1. Note down the primary Key and Connectionstring.

## Create Logger for APi management

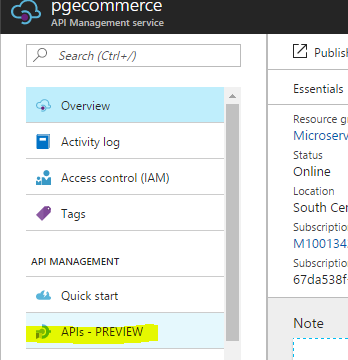
1. Pre-requisite, download POSTMAN tool. If already downloaded skip this step.

<https://www.google.co.in/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0ahUKEwiHnM-oqbzUAhXGpo8KHT_VD5QQFggnMAA&url=https%3A%2F%2Fchrome.google.com%2Fwebstore%2Fdetail%2Fpostman%2Ffhbjgbiflinjbdggehcddcbncdddomop%3Fhl%3Den&usg=AFQjCNE_Yq59TT1ZExzJ68FTldg4ho_lGw&sig2=xOVBGj1L_jaZQCJoRTt4jA>

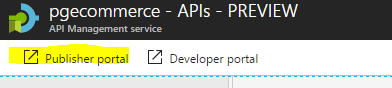
1. Login to Azure Portal; Navigate to respective resource group -> API Management service (pgecommerce).



1. Click on ‘APIs Preview’



1. Navigate to publisher portal by clicking ‘Publisher Portal’.



1. Open POSTMAN tool and create a PUT request using below URL:

<https://pgecommerce.management.azure-api.net/loggers/pgorderlogger?api-version=2014-02-14-preview>

Note down the logger name (highlighted) in the above URL.

* Specify the request body of the PUT with template. Crosscheck the event hub name and connectionstring. Note to remove entity path from event hub connection string.

{

"type" : "AzureEventHub",

"description" : "PG order logger",

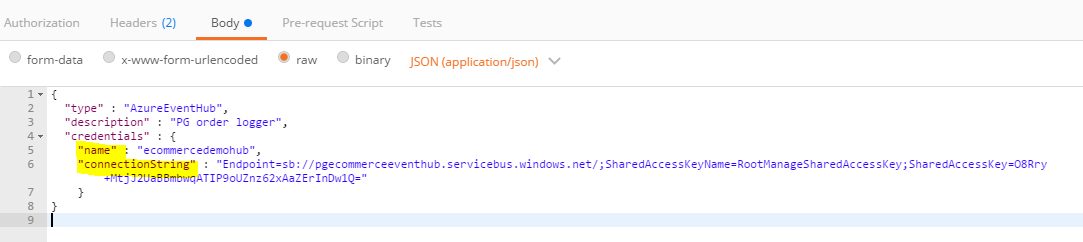
"credentials" : {

"name" : "ecommercedemohub",

"connectionString" : "Endpoint=sb://pgecommerceeventhub.servicebus.windows.net/;SharedAccessKeyName=RootManageSharedAccessKey;SharedAccessKey=O8Rry+MtjJ2UaBBmbwqATIP9oUZnz62xAaZErInDw1Q="

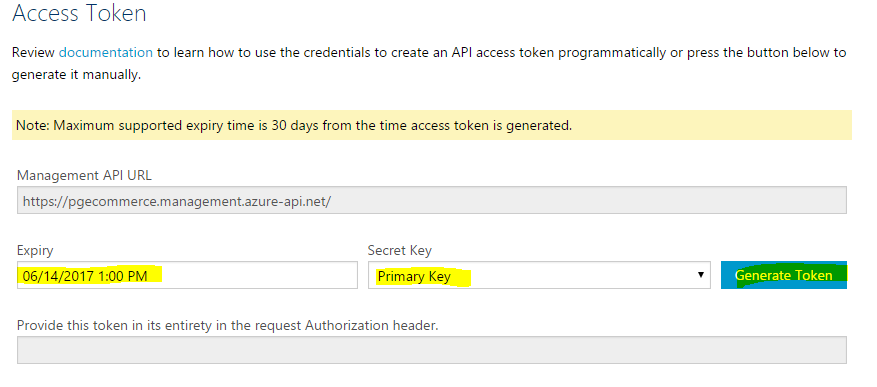
}

}

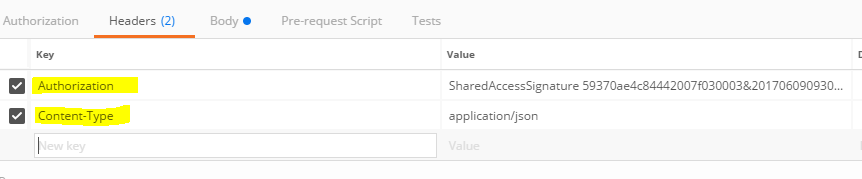


* Header of PUT call:
* Content-Type : application/json
* Authorization : SharedAccessSignature of API Management.

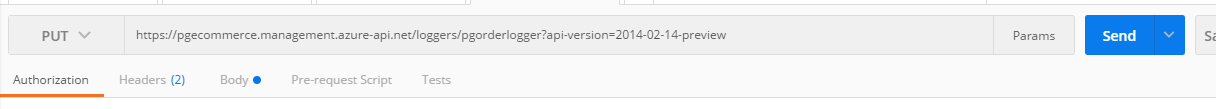
To get Shared Access Signature of API Management, navigate to API Management **publisher portal** -> **Security.** Have below highlighted setting and click Generate token. Copy this token to above PUT request header.



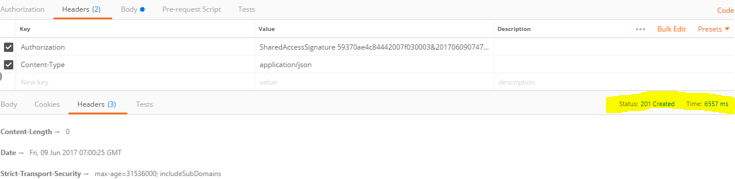
* Below screen is the header of the PUT request.



* Click on Send button.

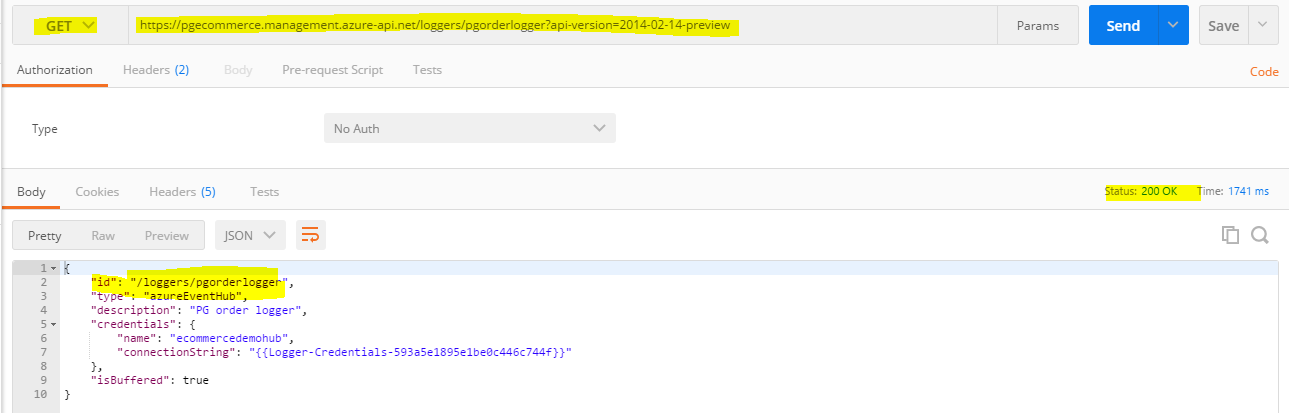


* On success, 201 message should display.



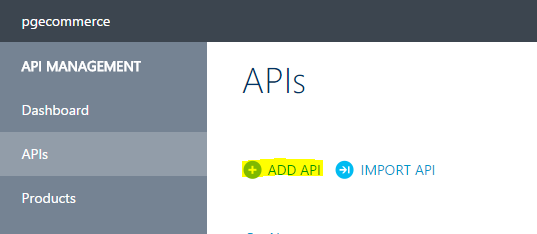
* To check further, make a GET call with same URL and click on ‘Send’. Message 200 should come along with body as highlighted in below screenshot.

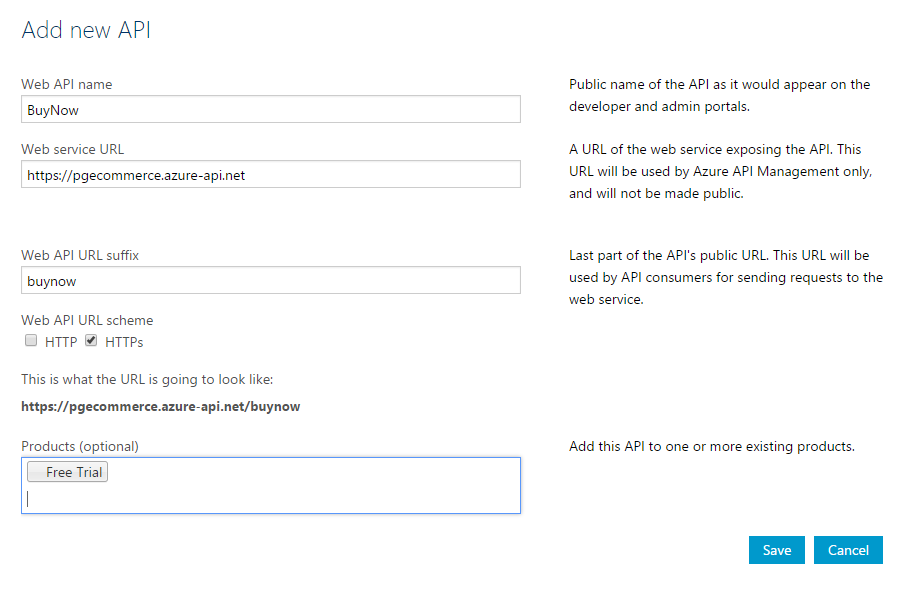
<https://pgecommerce.management.azure-api.net/loggers/pgorderlogger?api-version=2014-02-14-preview>



## Create BuyNOw Api

1. This API will aggregate the result from Order and Inventory with POST as method. The final response will have aggregated result. In addition, same API will be used for log event into event hub.
2. Login to Azure Portal and navigate to Publisher Portal of API Management Service.
3. Click on Add new API manually and enter below details. Click on ‘Save’.

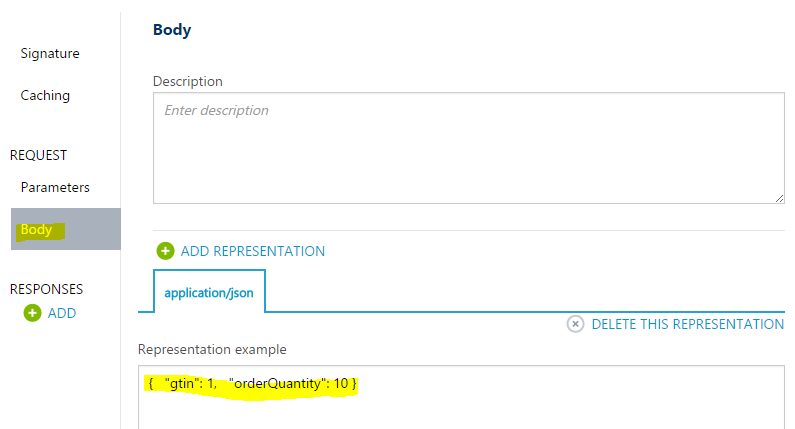




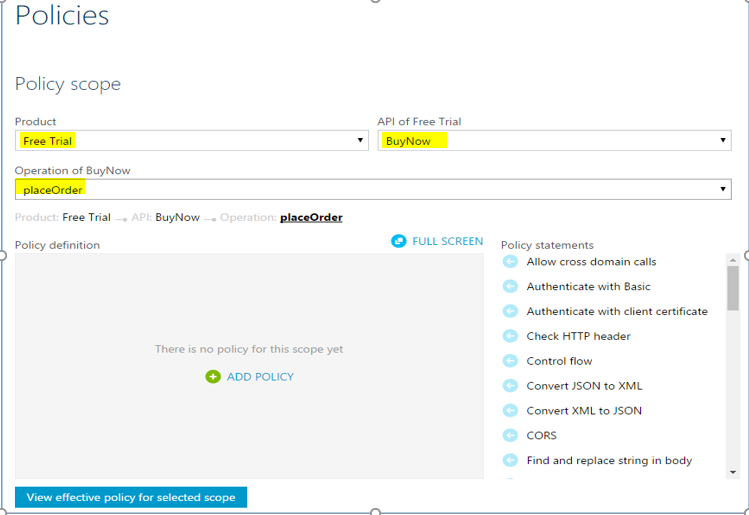
1. Add operation as:



1. Click on ‘Body’ and add ‘application/json’ as ADD REPRESENTATION. Click on ‘Save’



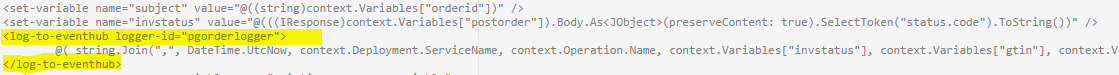
1. Go to Policies from left menu and click ADD POLICY.



1. Update the policy as attached in text file and click ‘Save’. Also cross-verify the endpoints for Order and Inventory in the Send-Request policy in the below text file before updating.

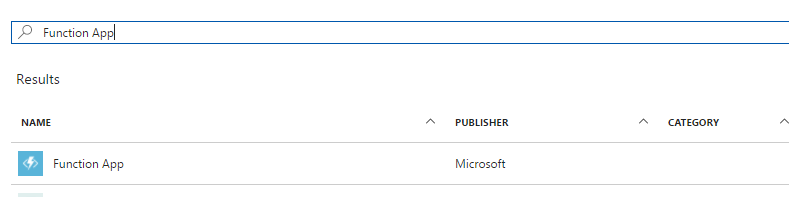


1. In the above policy, added updated policy for log into event hub too. Below highlighted screen which is responsible for log into event hub. Crosscheck the logger-id as created through POSTMAN tool.

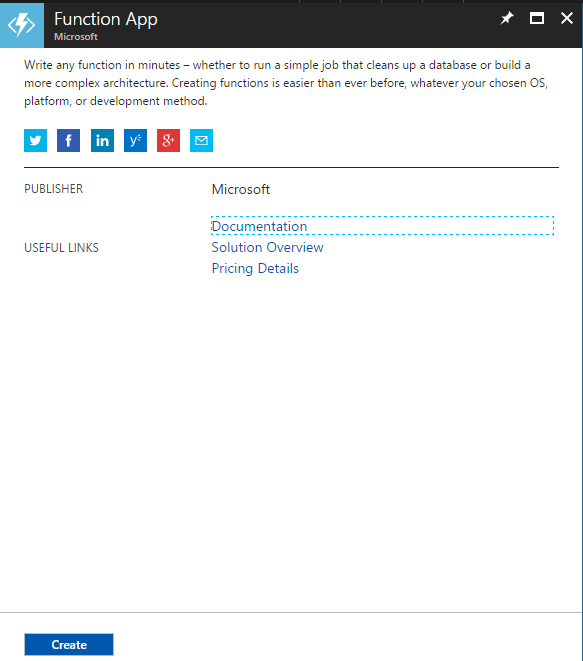


## Create azure function for event trigger

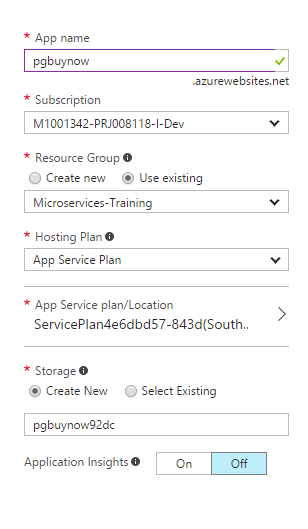
1. Login to Azure Portal; Navigate to the respective resource group and click ‘Add’.
2. Search for ‘Function App’.



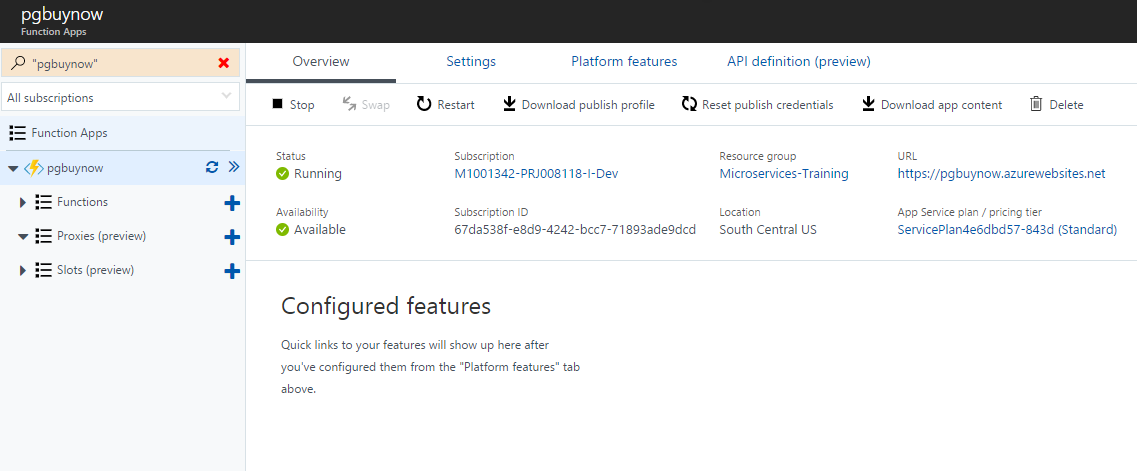
1. Select Function App and click ‘Create’.



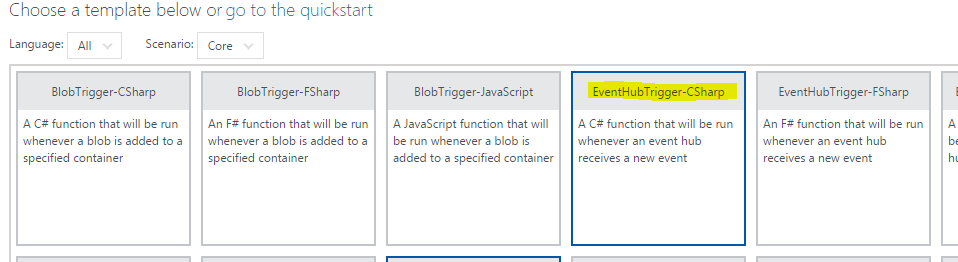
1. Enter App Name and click create.



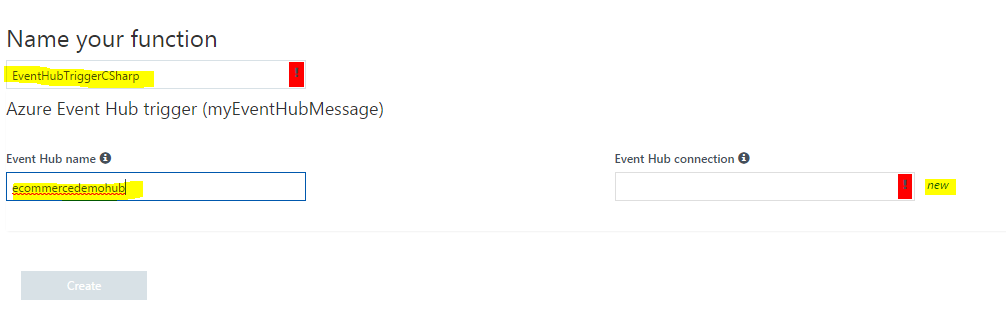
1. Once created, below screen should come. Else, click on Deployment Succeeded notification from Azure Portal.



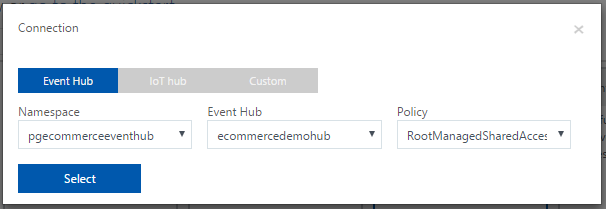
1. Click on ‘Functions’ -> New Function.
2. Choose template as EventHubTrigger-CSharp.



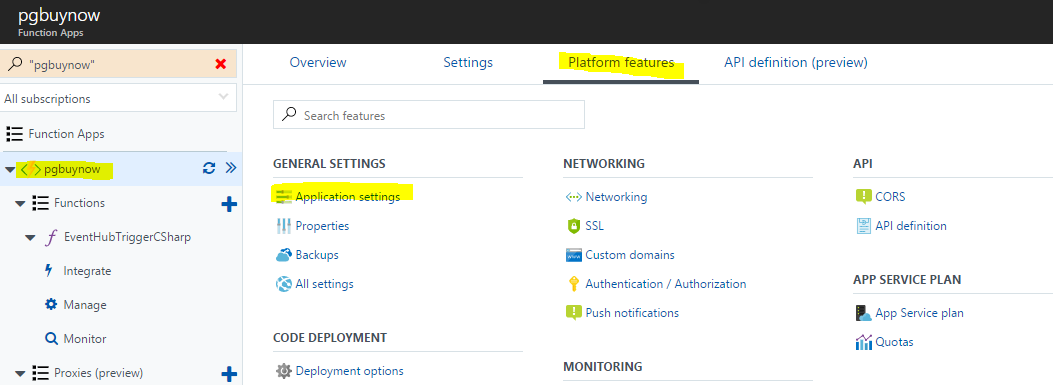
1. Enter highlighted fields and click Create.



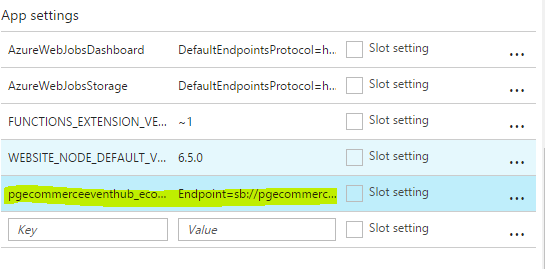
1. For Event hub connection, click on ‘new’ and select event hub created earlier in this document. Click Select.



1. Once function created. There is a known bug in Azure function for event hub connectionstring. To fix this, follow below steps:
   1. Navigate to pgbuynow -> Platform features -> Application Settings



* 1. Make sure newly added app setting for event hub does not carry entity path in the end of the connecting string.

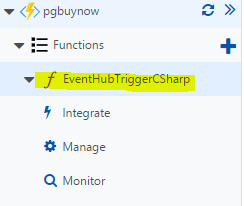


Your connection format should be similar to:

Endpoint=sb://pgecommerceeventhub.servicebus.windows.net/;SharedAccessKeyName=RootManagedSharedAccessKey;SharedAccessKey=G7jqsDAbXvYc1YPtz29FNZyvhdFf6OqUhZ6+Euf7bdQ=

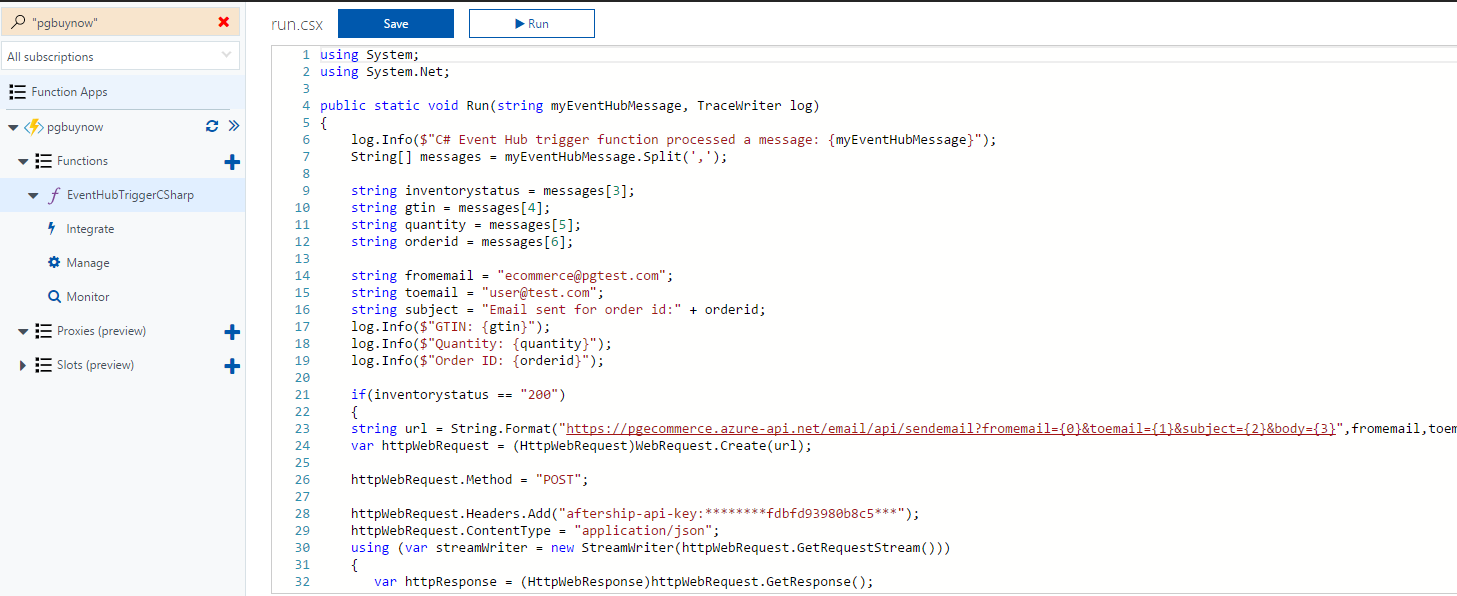
* 1. If any changes done, click ‘Save’.

1. Navigate to EventHubTriggerCSharp function.

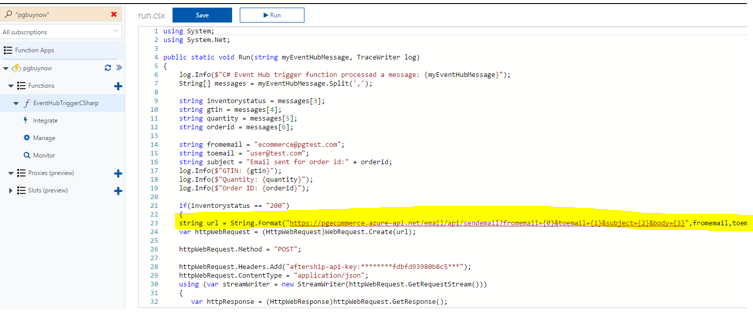


1. Copy paste the attached text file code into Editor. Click Save.



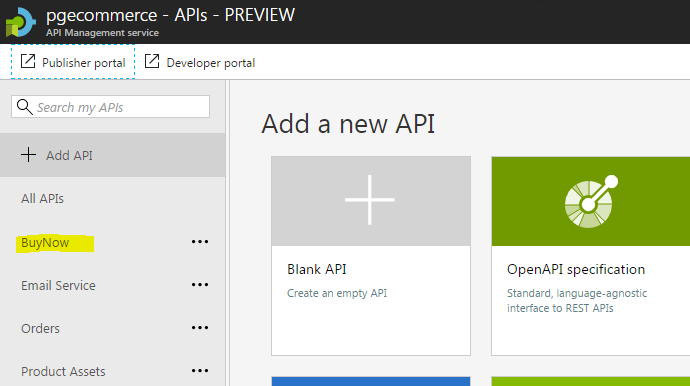


Note: in above code re-verify Email service API URL.

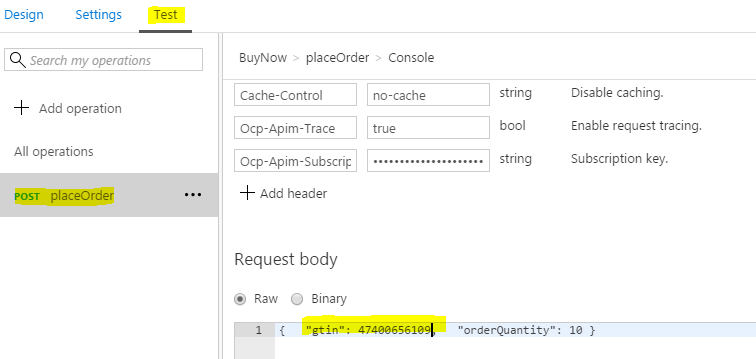


## Verify Aggregation and event hub trigger

1. Once policy updated, verify this API works fine. Navigate to the respective resource group -> API Management Service ->APIs Preview -> BuyNow API.



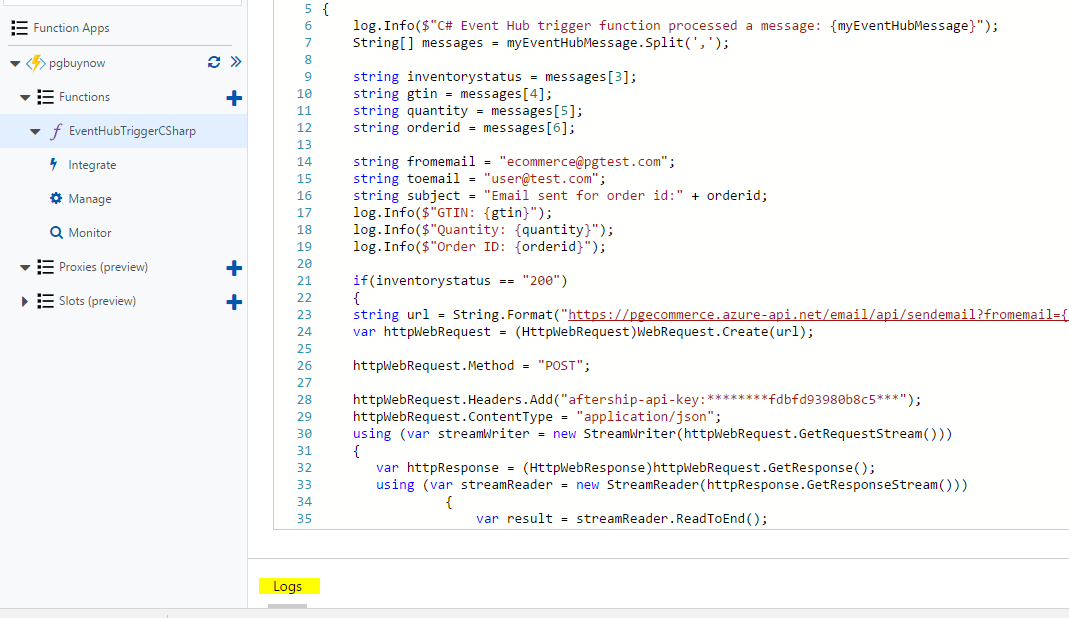
1. Select the below highlighted fields and enter valid GTIN under Test tab. Click on ‘Send’ button.



1. Response should have below body. This screen shows aggregation works fine between Order and Inventory APIs.



1. Verify Event hub trigger for email service. Navigate to Azure functions created in this document and click below ‘Logs’ button.



1. Output of logs should have below highlights. This shows email sent asynchronously post successful inventory.

