

CST8921 – Cloud Industry Trends

Title

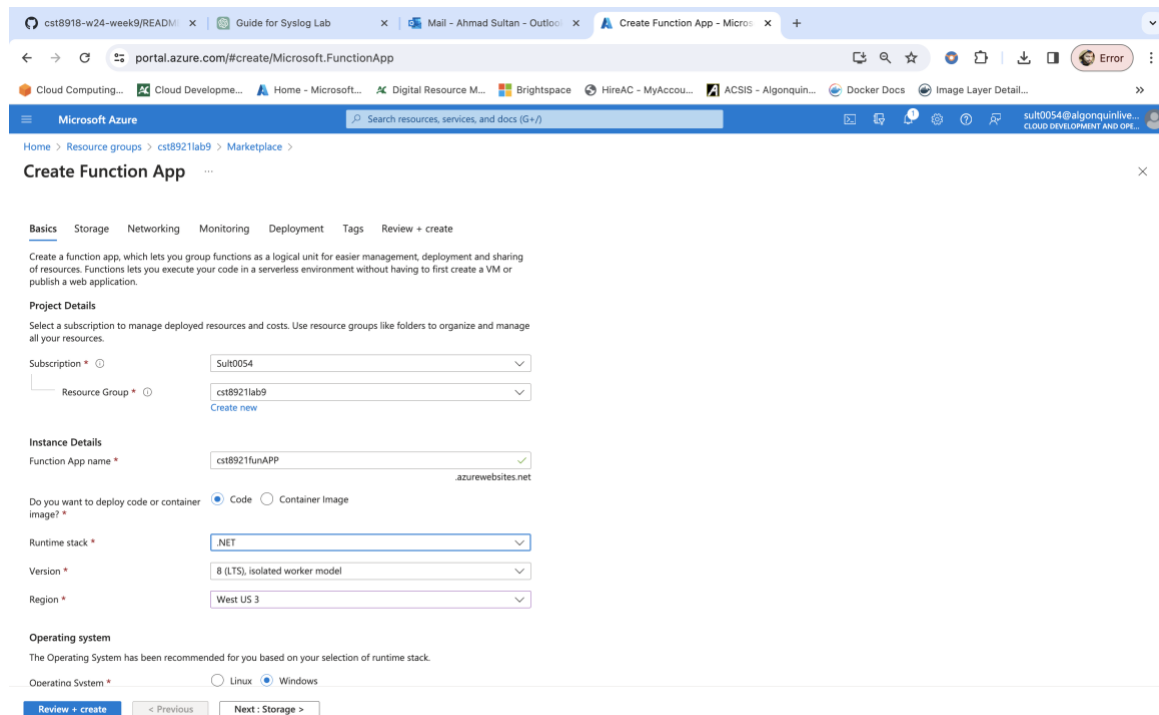
Exploring Serverless Computing in Cloud Environments

Summary

This lab introduced students to serverless computing services using Azure Functions and Azure Event Grid, emphasizing the elimination of infrastructure management tasks and focusing on code. The hands-on experience aimed to familiarize students with creating, testing, and monitoring serverless applications, specifically through the integration of Azure Functions with Azure Event Grid for event-driven architectures. The lab covered the entire process, from setting up a function app and event grid trigger to sending and receiving events, showcasing the agility and cost-effectiveness of serverless technologies.

Steps taken

1. Create a function app
2. Choose .NET as runtime stack



portal.azure.com/#create/Microsoft.FunctionApp

Microsoft Azure

Home > Resource groups > cst8921lab9 > Marketplace >

Create Function App

Basics Storage Networking Monitoring Deployment Tags Review + create

Create a function app, which lets you group functions as a logical unit for easier management, deployment and sharing of resources. Functions lets you execute your code in a serverless environment without having to first create a VM or publish a web application.

Project Details

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * Sult0054

Resource Group * cst8921lab9

Create new

Instance Details

Function App name * cst8921funAPP

Do you want to deploy code or container image? * ☒ Code ☐ Container Image

Runtime stack * .NET

Version * 8 (LTS, isolated worker model)

Region * West US 3

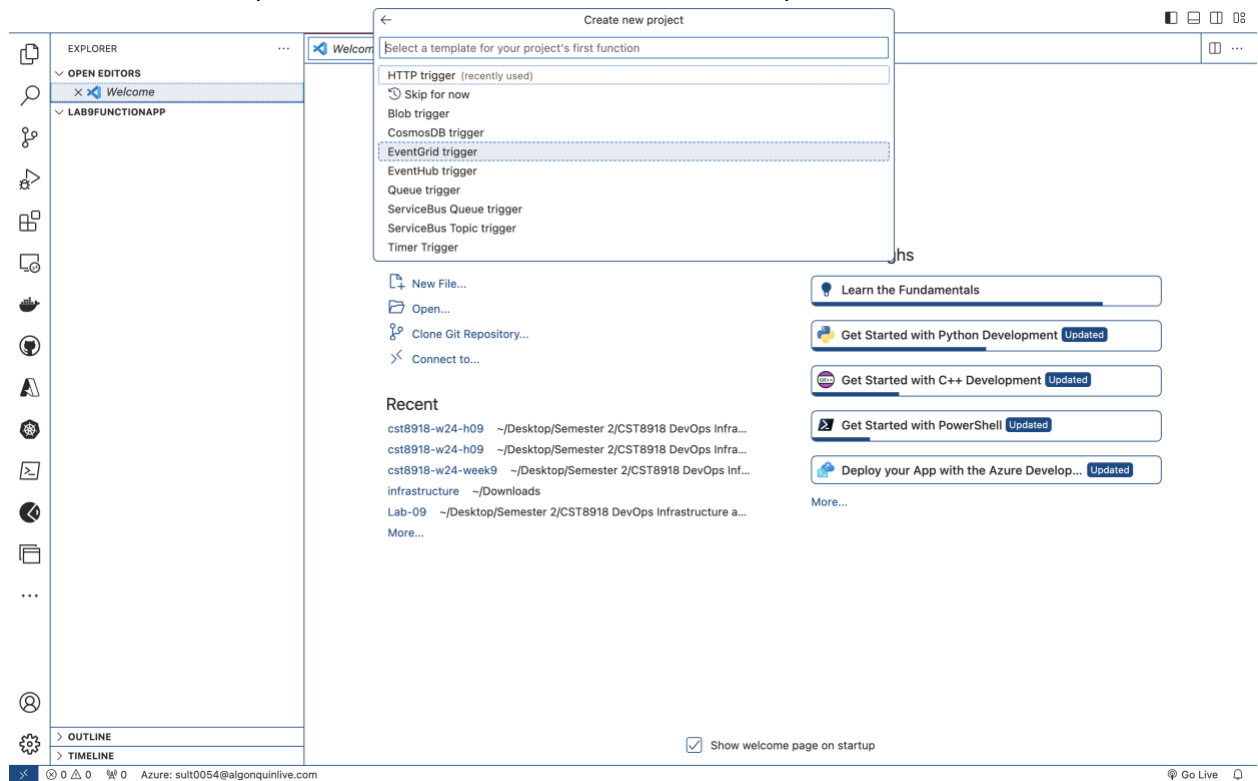
Operating system

The Operating System has been recommended for you based on your selection of runtime stack.

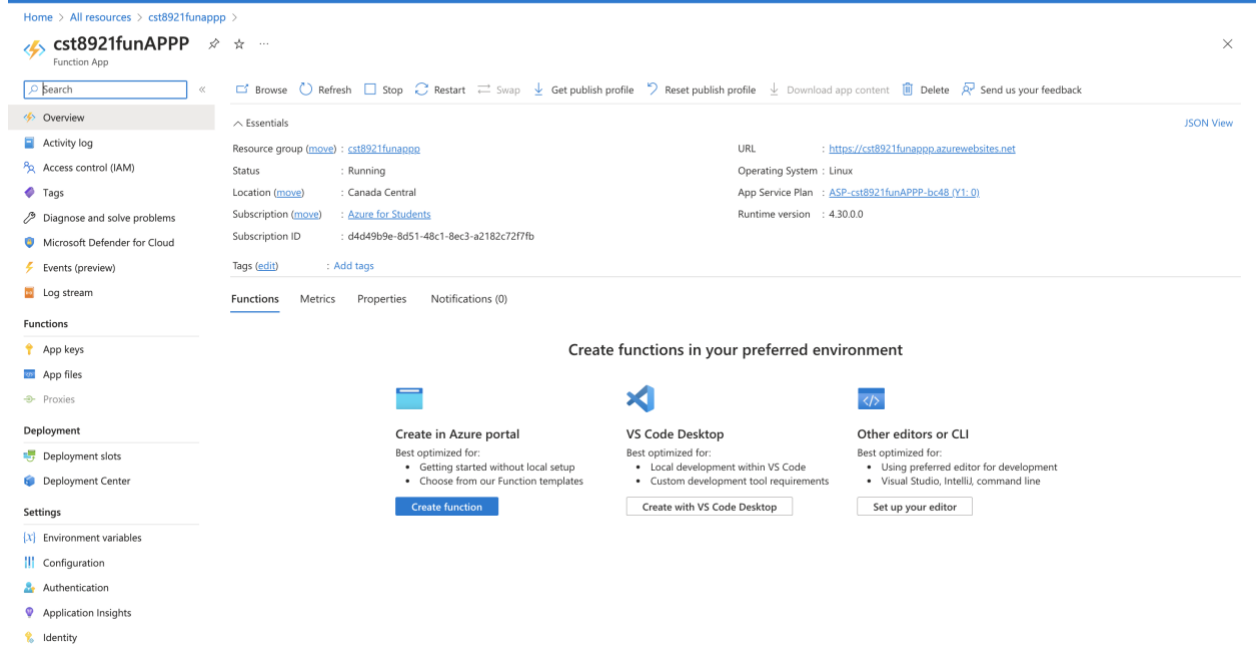
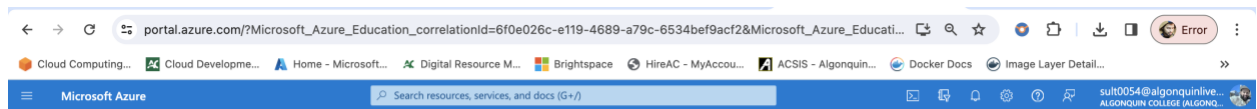
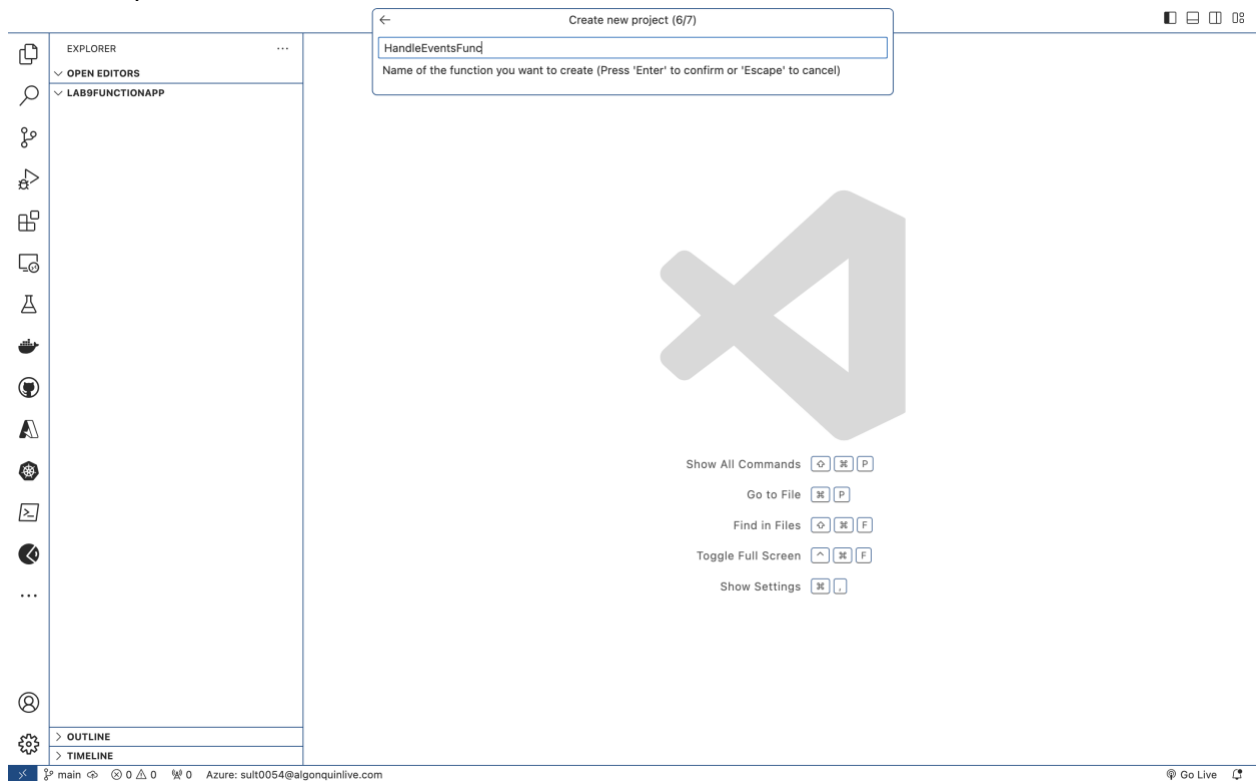
Operating System * ☐ Linux ☒ Windows

Review + create < Previous Next: Storage >

3. Create a function and in template section choose Azure event grid trigger
There was no option to create the function in the azure portal so I used VS Code

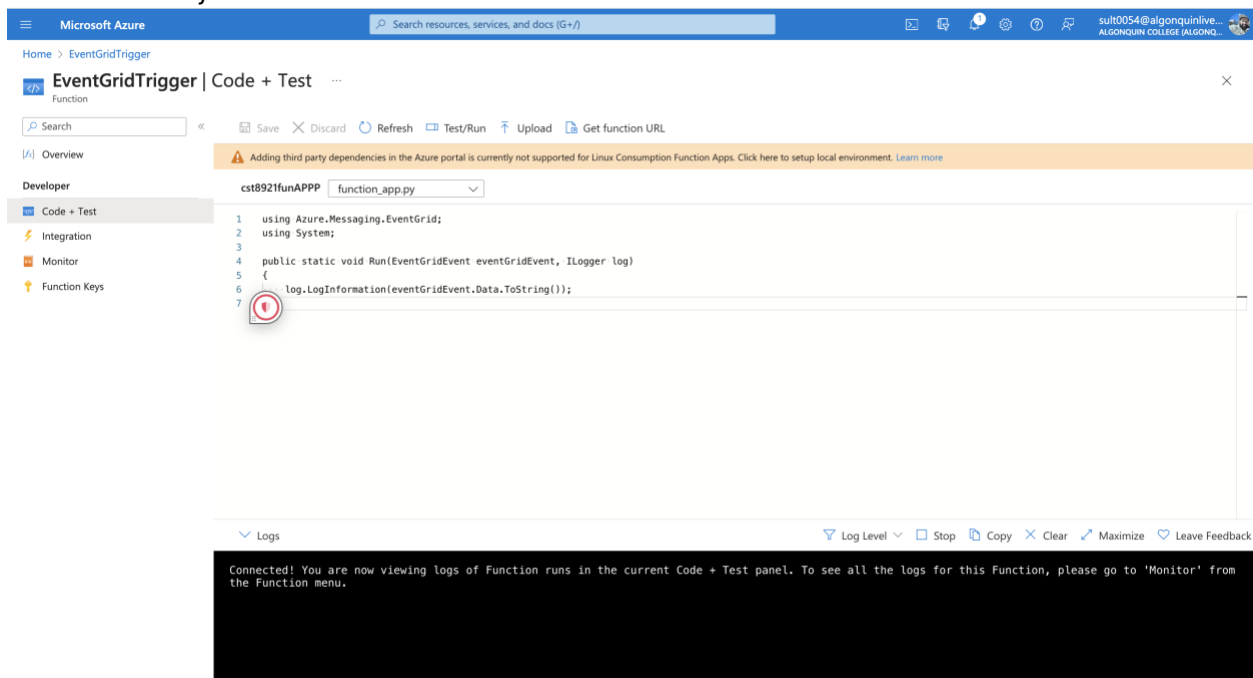


- In the Template details section in the bottom pane, enter a name for the function. In this example, it's HandleEventsFun



5. On the function page for handleeventsfunc, select code+test
6. Replace the code with the following code and test the function

```
#r "Azure.Messaging.EventGrid"
#r "System.Memory.Data"
using Azure.Messaging.EventGrid;
using System;
public static void Run(EventGridEvent eventGridEvent, ILogger log)
{
    log.LogInformation(eventGridEvent.Data.ToString());
}
```



Microsoft Azure

Home > EventGridTrigger

EventGridTrigger | Code + Test

Search resources, services, and docs (G+/I)

Save Discard Refresh Test/Run Upload Get function URL

Adding third party dependencies in the Azure portal is currently not supported for Linux Consumption Function Apps. Click here to setup local environment. [Learn more](#)

Developer

Code + Test

Integration

Monitor

Function Keys

cs8921funAPPP function_app.py

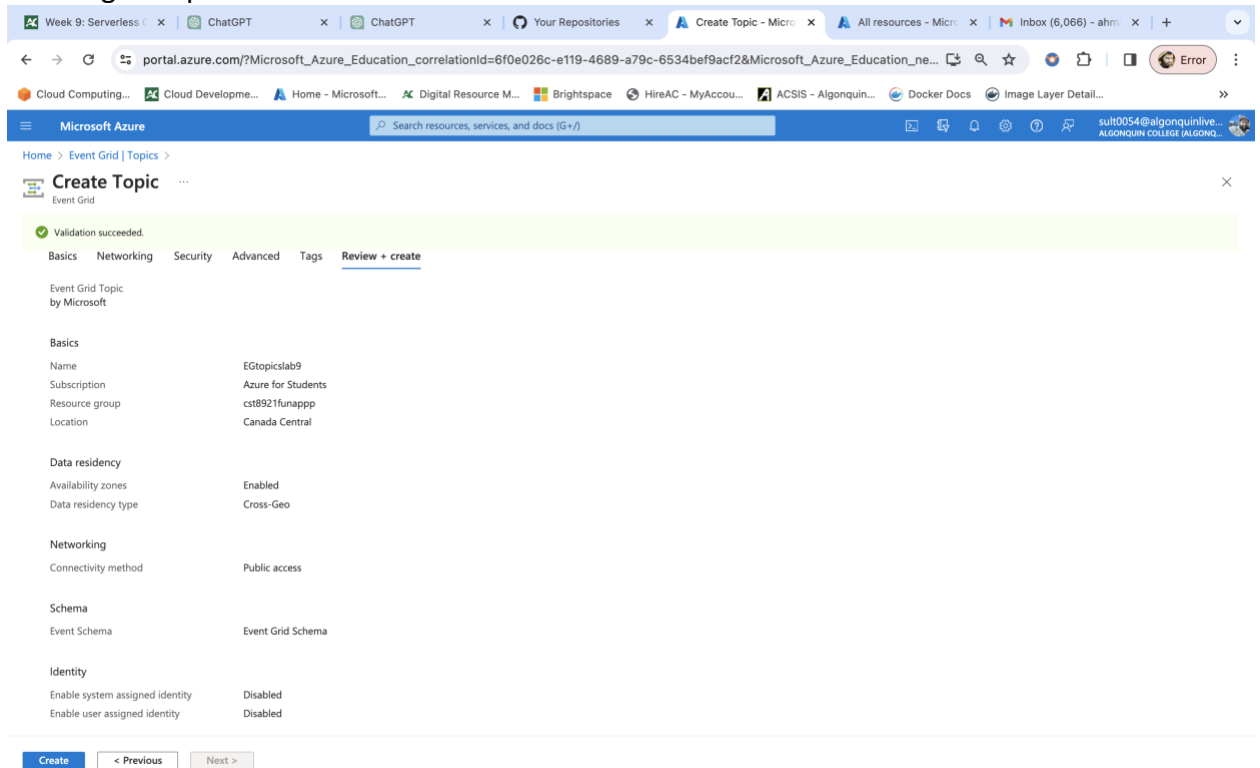
```
1 using Azure.Messaging.EventGrid;
2 using System;
3
4 public static void Run(EventGridEvent eventGridEvent, ILogger log)
5 {
6     log.LogInformation(eventGridEvent.Data.ToString());
7 }
```

Logs

Log Level Stop Copy Clear Maximize Leave Feedback

Connected! You are now viewing logs of Function runs in the current Code + Test panel. To see all the logs for this Function, please go to 'Monitor' from the Function menu.

7. Select monitor to see the received event information. Create a custom topic in the events grid topic



portal.azure.com/?Microsoft_Azure_Education_correlationId=6f0e026c-e119-4689-a79c-6534bef9acf2&Microsoft_Azure_Education_ne...

Cloud Computing... Cloud Developme... Home - Microsoft... Digital Resource M... Brightspace HireAC - MyAccou... ACSIS - Algonquin... Docker Docs Image Layer Detail...

Microsoft Azure Search resources, services, and docs (G+)

Home > Event Grid | Topics >

Create Topic Event Grid

Validation succeeded.

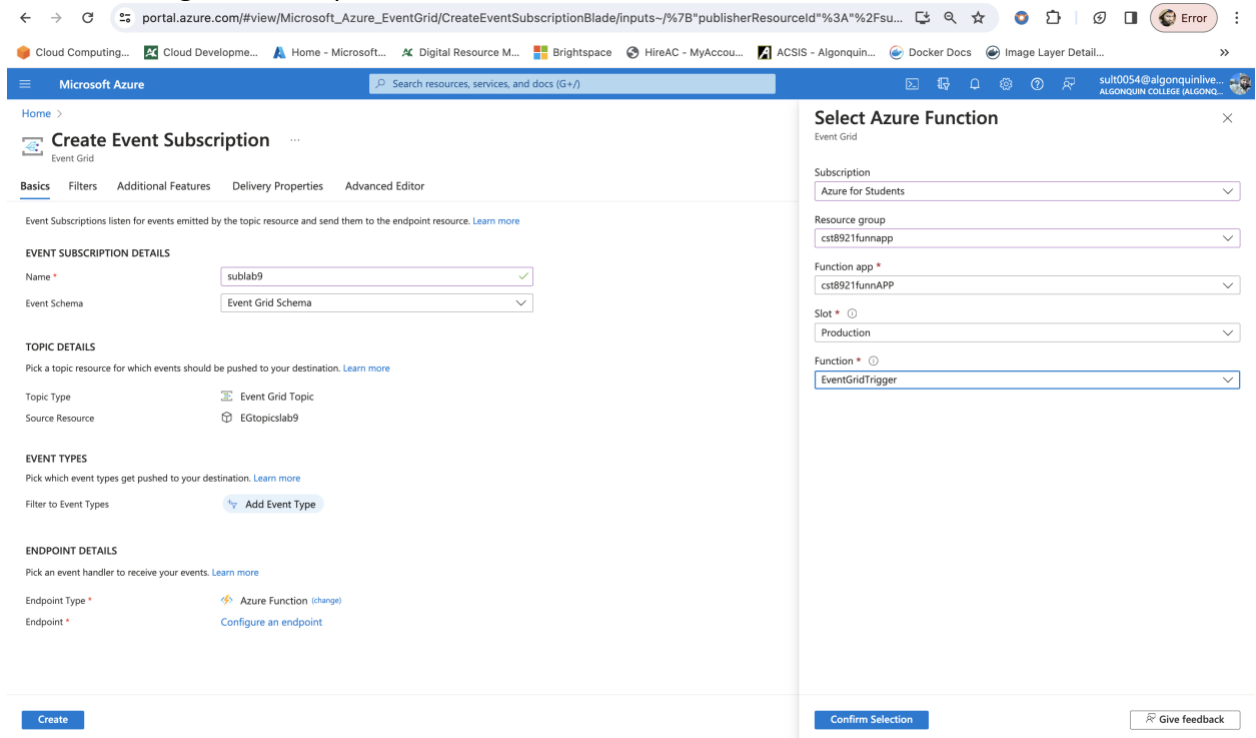
Basics Networking Security Advanced Tags Review + create

Event Grid Topic by Microsoft

| | |
|---------------------------------|--------------------|
| Basics | |
| Name | EGtopicslab9 |
| Subscription | Azure for Students |
| Resource group | cst8921funapp |
| Location | Canada Central |
| Data residency | |
| Availability zones | Enabled |
| Data residency type | Cross-Geo |
| Networking | |
| Connectivity method | Public access |
| Schema | |
| Event Schema | Event Grid Schema |
| Identity | |
| Enable system assigned identity | Disabled |
| Enable user assigned identity | Disabled |

Create < Previous Next >

8. Subscribe to the custom topic by selecting event subscription in event grid topic page
9. On the Create Event Subscription page, follow these steps:
10. Enter a name for the event subscription.
11. Select Azure Function for the Endpoint type.
12. Choose Configure an endpoint.



13. For the function endpoint, select the Azure Subscription and Resource Group your Function App is in and then select the Function App and function you created earlier. Select Confirm Selection.
14. On the Create Event Subscription page, switch to the Advanced Features tab, and set values for Max events per batch and Preferred batch size in kilobytes.
15. Batching can give you high-throughput. For Max events per batch, set maximum number of events that a subscription will include in a batch. Preferred batch size sets the preferred upper bound of batch size in kilo bytes, but can be exceeded if a single event is larger than this threshold.

The screenshot shows the Microsoft Azure portal interface for configuring an Event Subscription. The top navigation bar includes the Algonquin College logo and a user profile. The breadcrumb trail indicates the path: Home > All resources > EGtopicslab9 > Event Subscriptions. The main content area is titled 'sublab9 Event Subscription' and includes a 'Save' button and a 'Give feedback' link. The configuration section is titled 'Customize how many times and for how long event delivery will be retried. Learn more'. It contains several settings: 'Max Event Delivery Attempts' set to 30, 'Event Time to Live' set to 1 day, 'DEAD-LETTERING' with an unchecked 'Enable dead-lettering' checkbox, 'EVENT SUBSCRIPTION EXPIRATION TIME' with an unchecked 'Enable expiration time' checkbox, 'BATCHING' with 'Max events per batch' set to 20 and 'Preferred batch size in kilobytes' set to 64, and 'LABELS' with an 'Add Label' button.

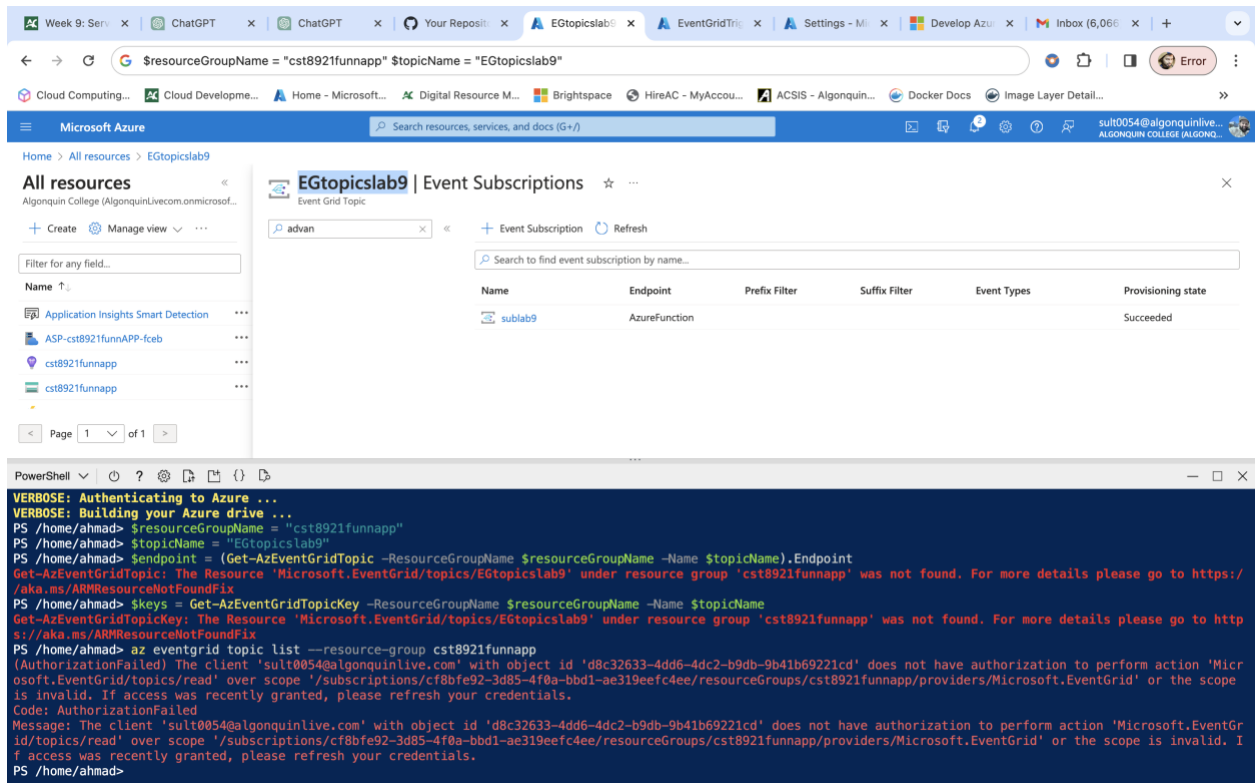
16. Send an event to your topic – Now, let's trigger an event to see how Event Grid distributes the message to your endpoint. Use either Azure CLI or PowerShell to send a test event to your custom topic. Typically, an application or Azure service would send the event data.

17. In powershell, set the following variables and update the topic name and resource group name:

```
$resourceGroupName = "cst8921funnapp"
$topicName = "EGtopicslab9"
```

18. Run the following commands to get the endpoint and the keys for the topic:

```
$endpoint = (Get-AzEventGridTopic -ResourceGroupName $resourceGroupName -Name
$topicName).Endpoint
$keys = Get-AzEventGridTopicKey -ResourceGroupName $resourceGroupName -Name
$topicName
```



Microsoft Azure

Home > All resources > EGtopicslab9

All resources
Algonquin College (AlgonquinLive.com.onmicrosoft...)

+ Create Manage view ...

Filter for any field...

Application Insights Smart Detection ...

ASP-cst8921funnAPP-fceb ...

cst8921funnapp ...

cst8921funnapp ...

Page 1 of 1

EGtopicslab9 | Event Subscriptions

advan + Event Subscription Refresh

Search to find event subscription by name...

| Name | Endpoint | Prefix Filter | Suffix Filter | Event Types | Provisioning state |
|---------|---------------|---------------|---------------|-------------|--------------------|
| sublab9 | AzureFunction | | | | Succeeded |

PowerShell

```
VERBOSE: Authenticating to Azure ...
VERBOSE: Building your Azure drive ...
PS /home/ahmad> $resourceGroupName = "cst8921funnapp"
PS /home/ahmad> $topicName = "EGtopicslab9"
PS /home/ahmad> $endpoint = (Get-AzEventGridTopic -ResourceGroupName $resourceGroupName -Name $topicName).Endpoint
Get-AzEventGridTopic: The Resource 'Microsoft.EventGrid/topics/EGtopicslab9' under resource group 'cst8921funnapp' was not found. For more details please go to https://aka.ms/ARMResourceNotFoundFix
PS /home/ahmad> $keys = Get-AzEventGridTopicKey -ResourceGroupName $resourceGroupName -Name $topicName
Get-AzEventGridTopicKey: The Resource 'Microsoft.EventGrid/topics/EGtopicslab9' under resource group 'cst8921funnapp' was not found. For more details please go to https://aka.ms/ARMResourceNotFoundFix
PS /home/ahmad> az eventgrid topic list --resource-group cst8921funnapp
(AuthorizationFailed) The client 'sult0054@algonquinlive.com' with object id 'd8c32633-4dd6-4dc2-b9db-9b41b69221cd' does not have authorization to perform action 'Microsoft.EventGrid/topics/read' over scope '/subscriptions/cf8bfe92-3d85-4f0a-bbd1-ae319eefc4ee/resourceGroups/cst8921funnapp/providers/Microsoft.EventGrid' or the scope is invalid. If access was recently granted, please refresh your credentials.
Code: AuthorizationFailed
Message: The client 'sult0054@algonquinlive.com' with object id 'd8c32633-4dd6-4dc2-b9db-9b41b69221cd' does not have authorization to perform action 'Microsoft.EventGrid/topics/read' over scope '/subscriptions/cf8bfe92-3d85-4f0a-bbd1-ae319eefc4ee/resourceGroups/cst8921funnapp/providers/Microsoft.EventGrid' or the scope is invalid. If access was recently granted, please refresh your credentials.
PS /home/ahmad>
```

19. Prepare the event

\$eventID = Get-Random 99999

#Date format should be SortableDateTimePattern (ISO 8601)

\$eventDate = Get-Date -Format s

#Construct body using Hashtable

```
$htbody = @{
    id= $eventID
    eventType="recordInserted"
    subject="myapp/vehicles/motorcycles"
    eventTime= $eventDate
    data= @{
        make="Ducati"
        model="Monster"
    }
    dataVersion="1.0"
}
```

#Use ConvertTo-Json to convert event body from Hashtable to JSON Object

#Append square brackets to the converted JSON payload since they are expected in the event's JSON payload syntax

\$body = "["+(ConvertTo-Json \$htbody)+"]"

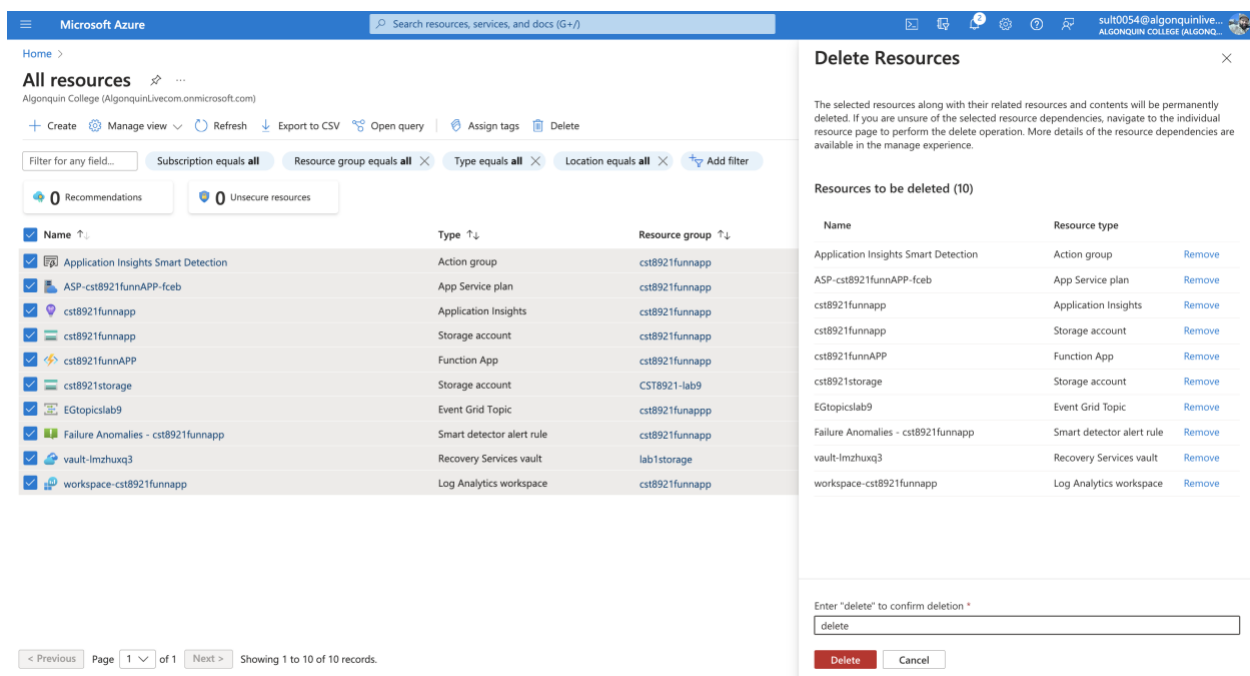
20. Use Invoke webrequest to send the event

Invoke-WebRequest -Uri \$endpoint -Method POST -Body \$body -Headers @{"aeg-sas-key" = \$keys.Key1}

21. Verify that function received the event

- On the monitor page of Azure function, see the invocation and details

22. Clean up the resources



The screenshot shows the Microsoft Azure portal interface. On the left, the 'All resources' page is visible, displaying a list of resources with columns for Name, Type, and Resource group. On the right, the 'Delete Resources' dialog is open, showing a list of 10 resources to be deleted. The resources are listed with their names and types, and each has a 'Remove' link. At the bottom of the dialog, there is a confirmation step where the user enters 'delete' to confirm the deletion.

| Name | Resource type | Action |
|--------------------------------------|---------------------------|------------------------|
| Application Insights Smart Detection | Action group | Remove |
| ASP-cst8921funnAPP-fceb | App Service plan | Remove |
| cst8921funnapp | Application Insights | Remove |
| cst8921funnapp | Storage account | Remove |
| cst8921funnAPP | Function App | Remove |
| cst8921storage | Storage account | Remove |
| EGtopicslab9 | Event Grid Topic | Remove |
| Failure Anomalies - cst8921funnapp | Smart detector alert rule | Remove |
| vault-lmzhuxq3 | Recovery Services vault | Remove |
| workspace-cst8921funnapp | Log Analytics workspace | Remove |

Challenges

- Authorization Issues:** Encountering permission-related errors when attempting to access or manage Azure resources, highlighting the importance of proper access rights and subscription management. I tried many things and also tried changing the subscription but nothing worked so I think WE do not have enough permissions on our azure subscriptions. Could not complete last step.

2. **Resource Management:** Difficulties in locating or verifying the existence of specific Azure resources (e.g., Event Grid topics), which may stem from naming inconsistencies, resource deletion, or operating in the wrong Azure subscription context.
3. **Technical Configuration:** Complexity in configuring event subscriptions and endpoints, as well as crafting and sending events correctly, requiring careful attention to Azure documentation and syntax accuracy.

Conclusion

The lab effectively demonstrated the value of serverless computing in modern cloud architectures by allowing students to engage directly with Azure Functions and Event Grid. Despite facing challenges such as authorization issues and resource management complexities, students gained practical insights into developing event-driven applications without the overhead of server management. This hands-on experience not only solidified their understanding of serverless concepts but also prepared them for leveraging these technologies in real-world scenarios to enhance scalability and cost-efficiency.