CST8912 – Cloud Solution Architecture

Graded Lab Activity #10

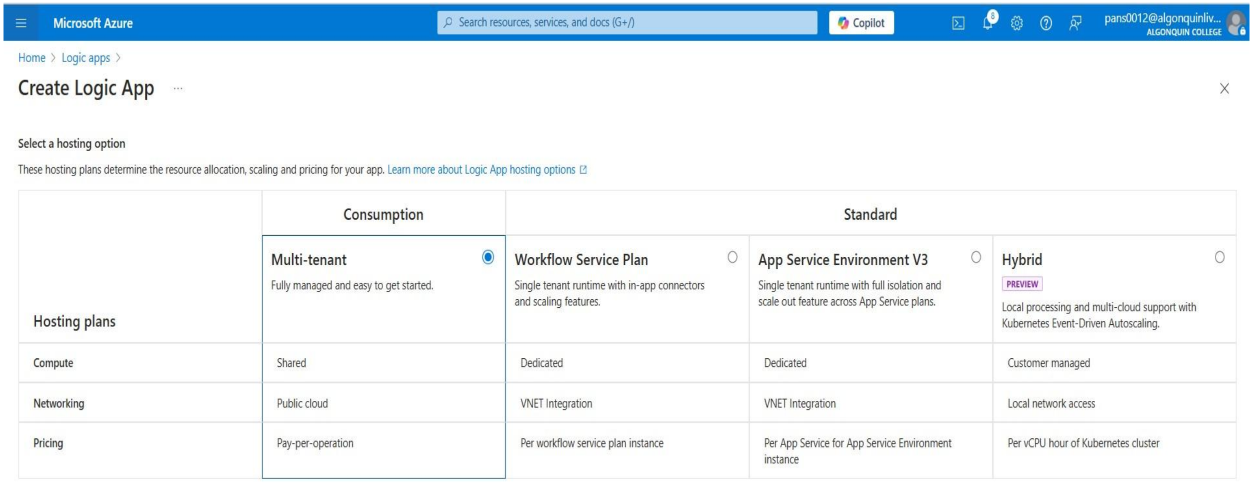
Introduction:

Azure Logic Apps is the PaaS (Platform as a Service) offering from Microsoft Azure. Logic Apps helps us to define workflows and build powerful solutions with the help of connectors, triggers, and actions.

Azure Logic Apps is a cloud platform where you can create and run automated workflows with little to no code. By using the visual designer and selecting from prebuilt operations, you can quickly build a workflow that integrates and manages your apps, data, services, and systems.

Task 1:

1. Create logic app (choose consumption based plan) and sql database instance in Canada central region

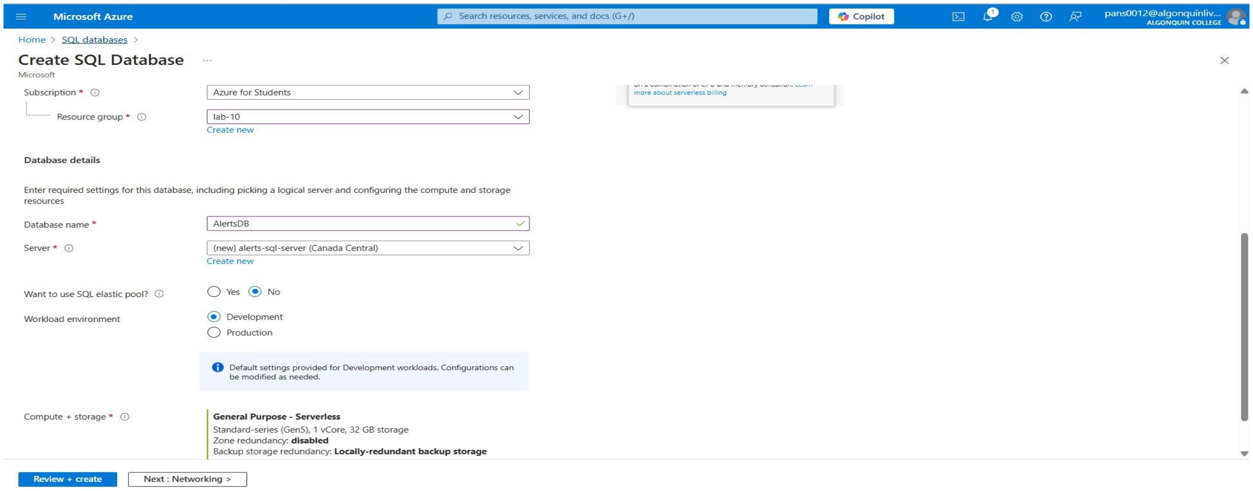


Create Logic App

Select a hosting option

These hosting plans determine the resource allocation, scaling and pricing for your app. Learn more about Logic App hosting options

	Consumption	Standard		
Hosting plans	Multi-tenant ● Fully managed and easy to get started.	Workflow Service Plan ○ Single tenant runtime with in-app connectors and scaling features.	App Service Environment V3 ○ Single tenant runtime with full isolation and scale out feature across App Service plans.	Hybrid ○ PREVIEW Local processing and multi-cloud support with Kubernetes Event-Driven Autoscaling.
Compute	Shared	Dedicated	Dedicated	Customer managed
Networking	Public cloud	VNET Integration	VNET Integration	Local network access
Pricing	Pay-per-operation	Per workflow service plan instance	Per App Service for App Service Environment instance	Per vCPU hour of Kubernetes cluster



Home > SQL databases >

Create SQL Database ···

Microsoft

more about serverless billing

Subscription * ⓘ Azure for Students

Resource group * ⓘ lab-10

Create new

Database details

Enter required settings for this database, including picking a logical server and configuring the compute and storage resources

Database name * AlertsDB ✓

Server * ⓘ (new) alerts-sql-server (Canada Central)

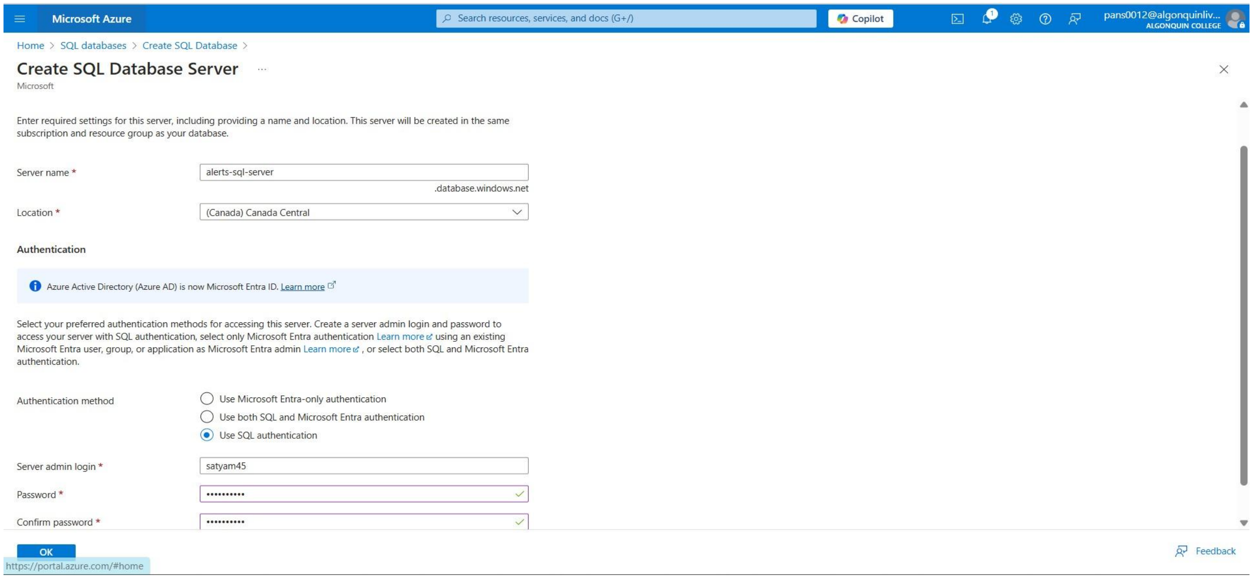
Create new

Want to use SQL elastic pool? ⓘ ○ Yes ● No

Workload environment ● Development ○ Production

ⓘ Default settings provided for Development workloads. Configurations can be modified as needed.

Compute + storage * ⓘ **General Purpose - Serverless**
Standard-series (Gen5), 1 vCore, 32 GB storage
Zone redundancy: **disabled**
Backup storage redundancy: **Locally-redundant backup storage**



Home > SQL databases > Create SQL Database >

Create SQL Database Server ···

Microsoft

Enter required settings for this server, including providing a name and location. This server will be created in the same subscription and resource group as your database.

Server name * alerts-sql-server .database.windows.net

Location * (Canada) Canada Central

Authentication

ⓘ Azure Active Directory (Azure AD) is now Microsoft Entra ID. Learn more

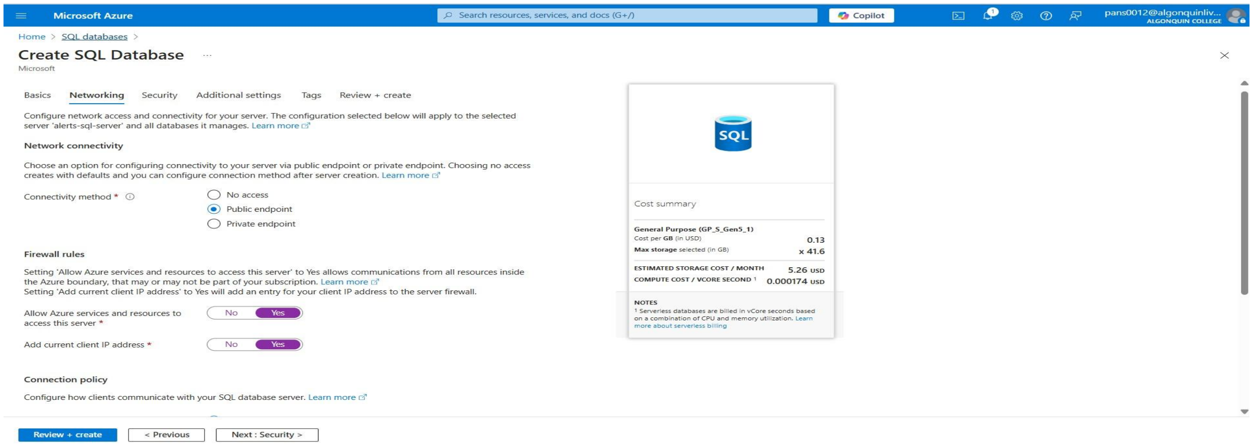
Select your preferred authentication methods for accessing this server. Create a server admin login and password to access your server with SQL authentication, select only Microsoft Entra authentication Learn more using an existing Microsoft Entra user, group, or application as Microsoft Entra admin Learn more , or select both SQL and Microsoft Entra authentication.

Authentication method
○ Use Microsoft Entra-only authentication
○ Use both SQL and Microsoft Entra authentication
● Use SQL authentication

Server admin login * satyam45

Password * •••••••••• ✓

Confirm password * •••••••••• ✓



Home > SQL databases >

Create SQL Database ···

Microsoft

Basics **Networking** Security Additional settings Tags Review + create

Configure network access and connectivity for your server. The configuration selected below will apply to the selected server 'alerts-sql-server' and all databases it manages. Learn more

Network connectivity

Choose an option for configuring connectivity to your server via public endpoint or private endpoint. Choosing no access creates with defaults and you can configure connection method after server creation. Learn more

Connectivity method * ⓘ
○ No access
● Public endpoint
○ Private endpoint

Firewall rules

Setting 'Allow Azure services and resources to access this server' to Yes allows communications from all resources inside the Azure boundary, that may or may not be part of your subscription. Learn more
Setting 'Add current client IP address' to Yes will add an entry for your client IP address to the server firewall.

Allow Azure services and resources to access this server * No [✓ Yes]

Add current client IP address * No [✓ Yes]

Connection policy

Configure how clients communicate with your SQL database server. Learn more

Cost summary

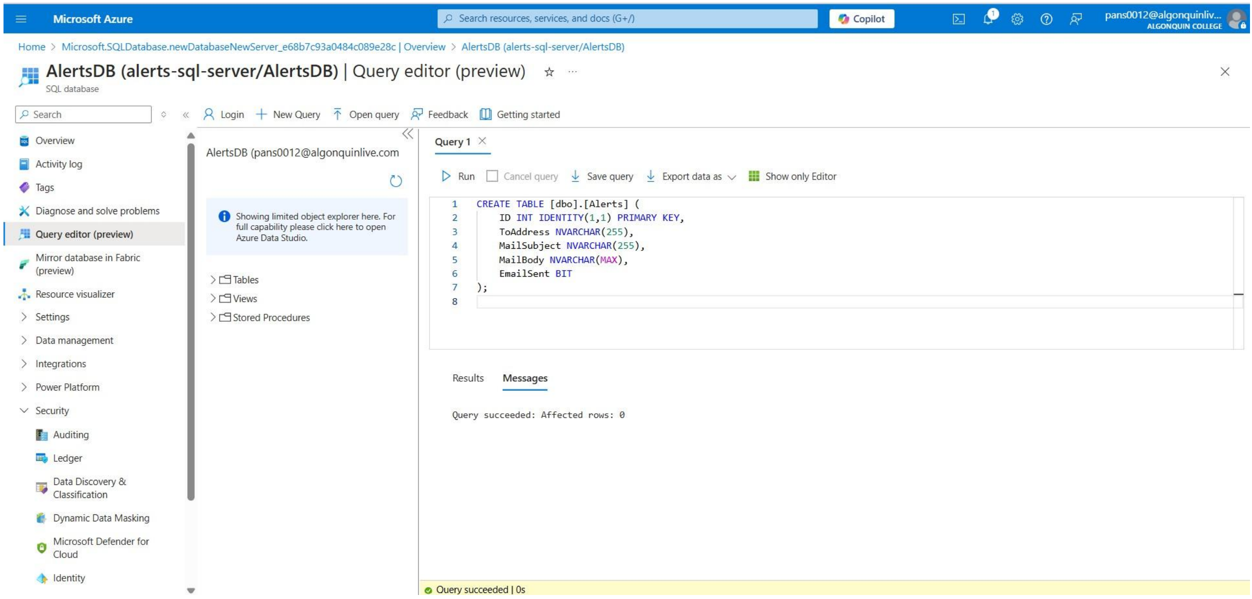
General Purpose (GP_S_Gen5_1)

Cost per **GB** (in USD)	0.13
Max storage selected (in GB)	x 41.6
ESTIMATED STORAGE COST / MONTH	5.26 USD
COMPUTE COST / VCORE SECOND [1]	0.000174 USD

NOTES

[1] Serverless databases are billed in vCore seconds based on a combination of CPU and memory utilization. Learn more about serverless billing

Create alerts table in sql database using query editor


Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
pans0012@algonquinliv...
ALGONQUIN COLLEGE
Home > Microsoft.SQLDatabase.newDatabaseNewServer_e68b7c93a0484c089e28c | Overview > AlertsDB (alerts-sql-server/AlertsDB)
AlertsDB (alerts-sql-server/AlertsDB) | Query editor (preview)
SQL database
Search
Login
New Query
Open query
Feedback
Getting started
Overview
Activity log
Tags
Diagnose and solve problems
Query editor (preview)
Mirror database in Fabric (preview)
Resource visualizer
Settings
Data management
Integrations
Power Platform
Security
Auditing
Ledger
Data Discovery & Classification
Dynamic Data Masking
Microsoft Defender for Cloud
Identity
AlertsDB (pans0012@algonquinlive.com
Showing limited object explorer here. For full capability please click here to open Azure Data Studio.
Tables
Views
Stored Procedures
Query 1
Run
Cancel query
Save query
Export data as
Show only Editor
Results
Messages
Query succeeded | 0s

```sql
CREATE TABLE [dbo].[Alerts] (
    ID INT IDENTITY(1,1) PRIMARY KEY,
    ToAddress NVARCHAR(255),
    MailSubject NVARCHAR(255),
    MailBody NVARCHAR(MAX),
    EmailSent BIT
);
```

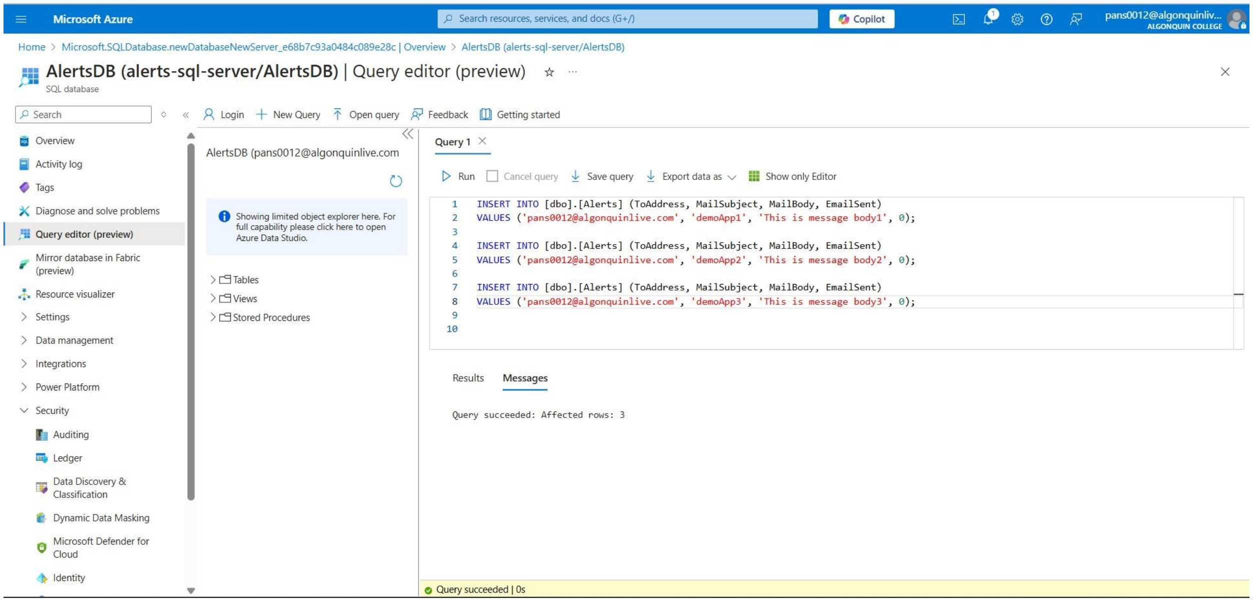
```
Query succeeded: Affected rows: 0
```

2. Insert records into the Alerts table using the query

   Insert into [dbo].[Alerts] (ToAddress,MailSubject,MailBody,EmailSent)
   values ('youremail','demoApp1','This is message body1',0)
   go

   Insert into [dbo].[Alerts] (ToAddress,MailSubject,MailBody,EmailSent)
   values ('youremail','demoApp2','This is message body2',0)
   go

   Insert into [dbo].[Alerts] (ToAddress,MailSubject,MailBody,EmailSent)
   values ('youremail','demoApp3','This is message body3',0)
   go

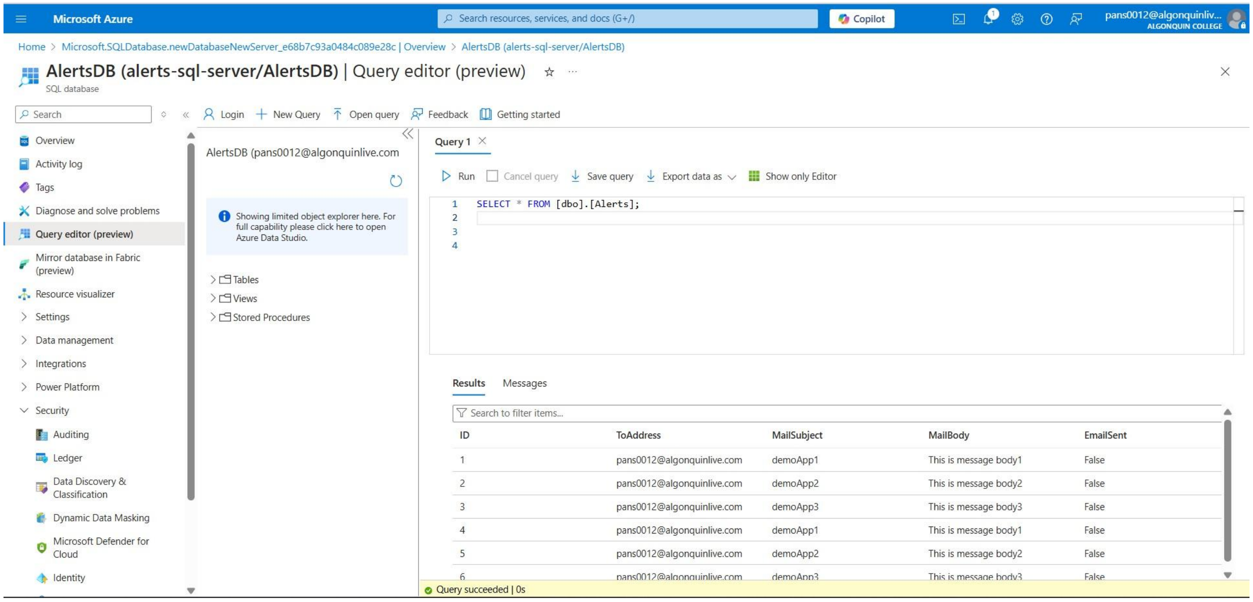


```
INSERT INTO [dbo].[Alerts] (ToAddress, MailSubject, MailBody, EmailSent)
VALUES ('pans0012@algonquinlive.com', 'demoApp1', 'This is message body1', 0);

INSERT INTO [dbo].[Alerts] (ToAddress, MailSubject, MailBody, EmailSent)
VALUES ('pans0012@algonquinlive.com', 'demoApp2', 'This is message body2', 0);

INSERT INTO [dbo].[Alerts] (ToAddress, MailSubject, MailBody, EmailSent)
VALUES ('pans0012@algonquinlive.com', 'demoApp3', 'This is message body3', 0);
```

```
Query succeeded: Affected rows: 3
```

Query succeeded | 0s

3. Select rows from db.Alerts table to verify the records inserted in the table.

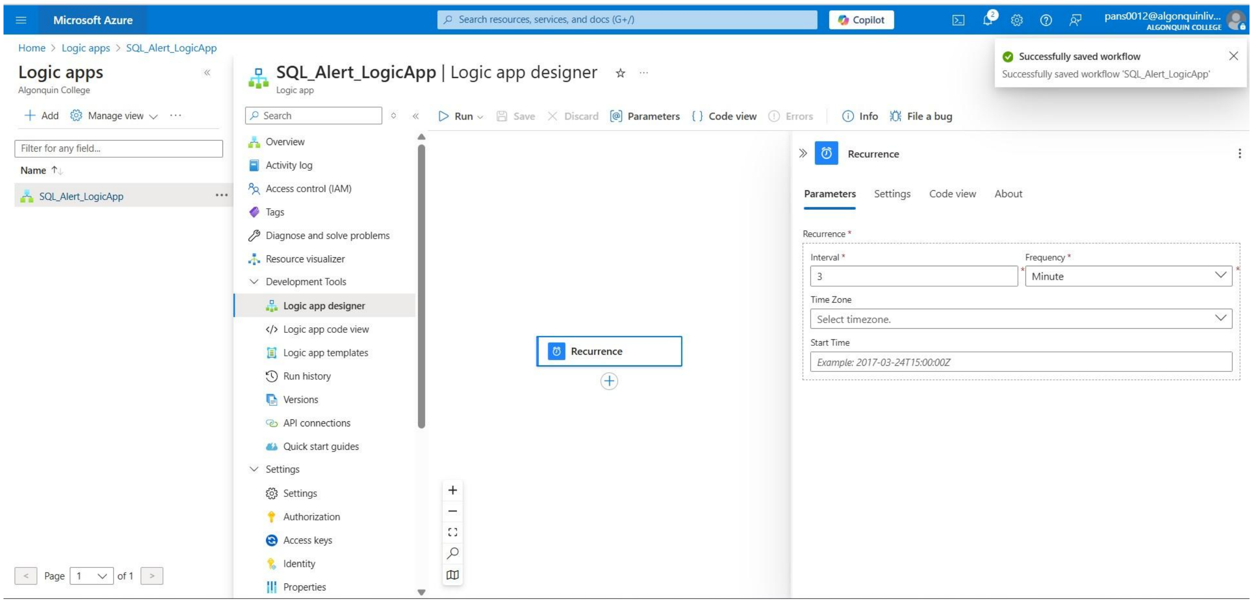


```
SELECT * FROM [dbo].[Alerts];
```

ID	ToAddress	MailSubject	MailBody	EmailSent
1	pans0012@algonquinlive.com	demoApp1	This is message body1	False
2	pans0012@algonquinlive.com	demoApp2	This is message body2	False
3	pans0012@algonquinlive.com	demoApp3	This is message body3	False
4	pans0012@algonquinlive.com	demoApp1	This is message body1	False
5	pans0012@algonquinlive.com	demoApp2	This is message body2	False
6	pans0012@algonquinlive.com	demoApp3	This is message body3	False

Query succeeded | 0s

4. Go to logic app created in lab
5. Use recurrence trigger and define values for interval (3) and frequency (minute)



Successfully saved workflow

Successfully saved workflow 'SQL_Alert_LogicApp'

Recurrence *

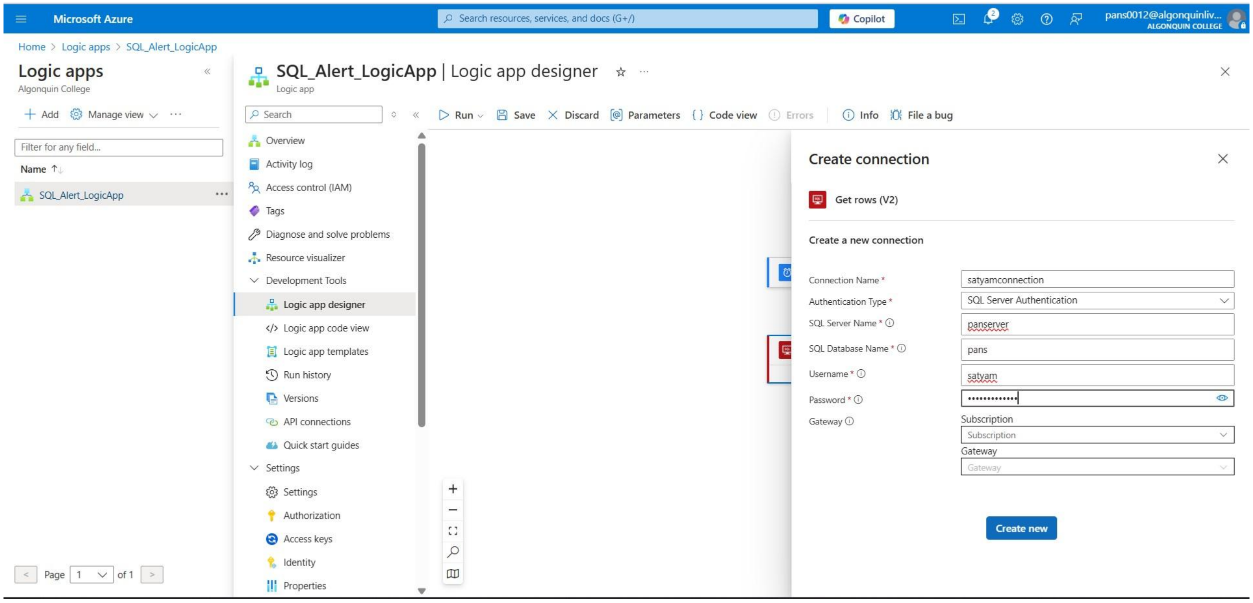
Interval * 3

Frequency * Minute

Time Zone: Select timezone.

Start Time: Example: 2017-03-24T15:00:00Z

6. Add new step named “sql server” , use “get rows” as action.



Create connection

Get rows (V2)

Create a new connection

Connection Name * satyamconnection

Authentication Type * SQL Server Authentication

SQL Server Name * panserver

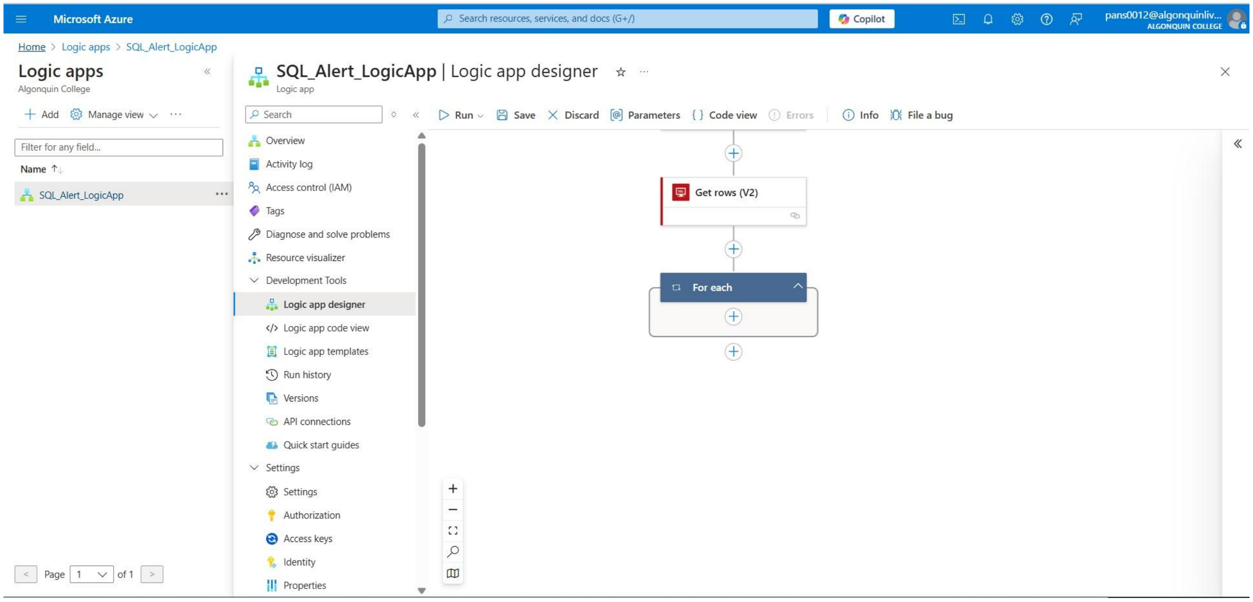
SQL Database Name * pans

Username * satyam

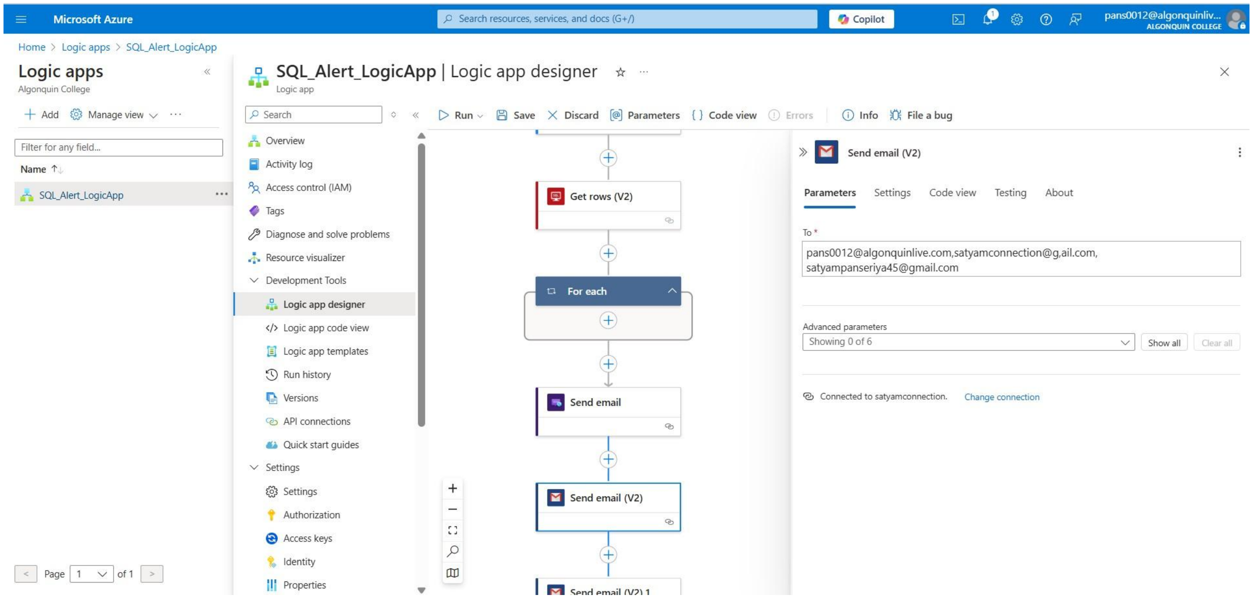
Password * •••••••••••••

Gateway: Subscription: Subscription; Gateway: Gateway

7. Enter the credentials (In background, connectors are getting created). Enter your server name (FQDN), database name, username and password.
8. Add a new step ‘For-Each’ in the Logic App.



9. Now add a 'Send mail' Action.
10. Enter the details from Dynamic content (refer to values from columns defined in the alerts table)



To *
pans0012@algonquinlive.com,satyamconnection@g,ail.com, satyampanseriya45@gmail.com

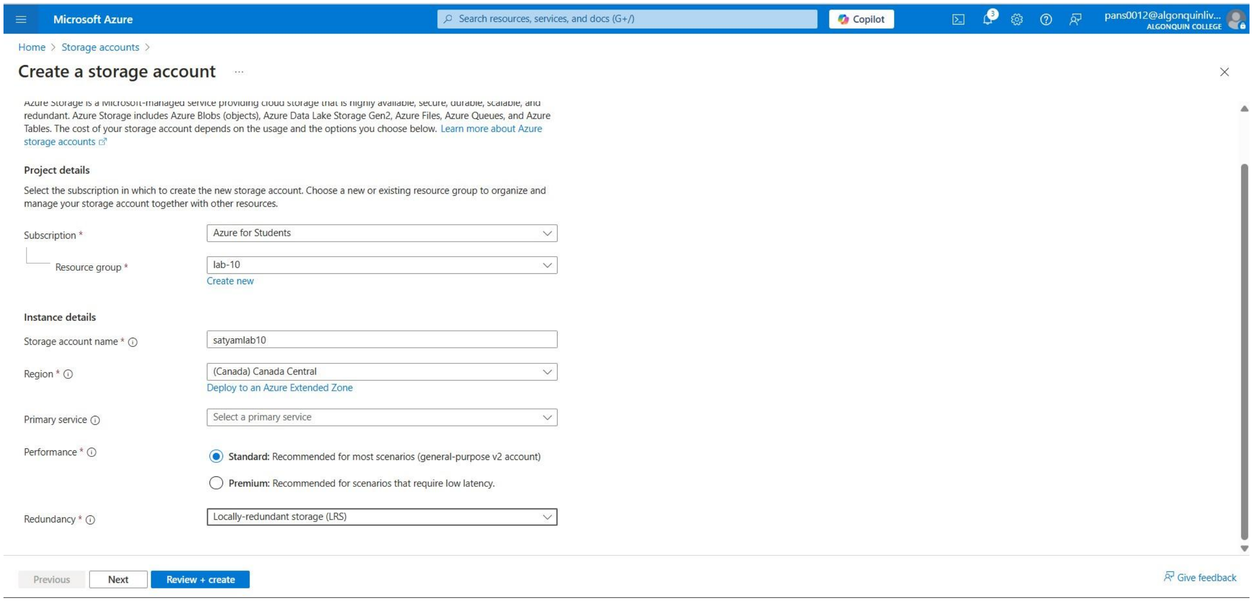
Advanced parameters
Showing 0 of 6

Connected to satyamconnection.

11. Save the Logic App.
12. Wait for sometime and you will receive an email.

Task 2: Design a logic to trigger an email notification in your outlook when the file to a specific folder does not gets uploaded by specific time.

1. Create a storage account in Canada central region


Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
pans0012@algonquinliv...
ALGONQUIN COLLEGE
Home > Storage accounts >

Create a storage account

Azure Storage is a Microsoft-managed service providing cloud storage that is highly available, secure, durable, scalable, and redundant. Azure Storage includes Azure Blobs (objects), Azure Data Lake Storage Gen2, Azure Files, Azure Queues, and Azure Tables. The cost of your storage account depends on the usage and the options you choose below. Learn more about Azure storage accounts

Project details

Select the subscription in which to create the new storage account. Choose a new or existing resource group to organize and manage your storage account together with other resources.

Subscription * Azure for Students

Resource group * lab-10

Create new

Instance details

Storage account name * satyamlab10

Region * (Canada) Canada Central

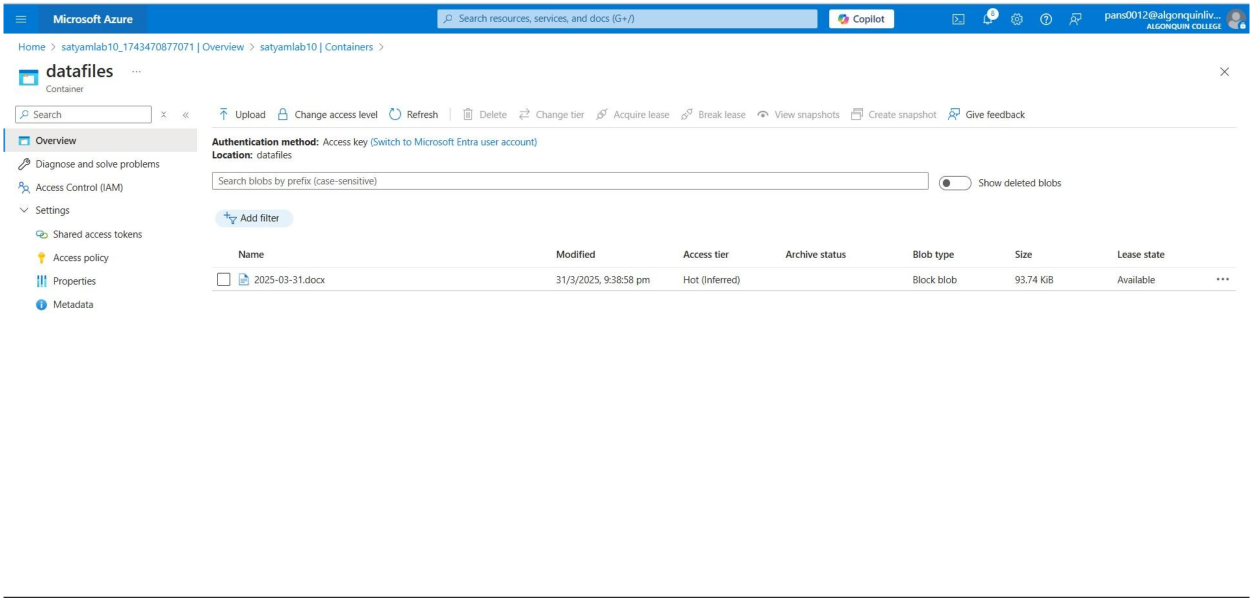
Deploy to an Azure Extended Zone

Primary service: Select a primary service

Performance *
● Standard: Recommended for most scenarios (general-purpose v2 account)
○ Premium: Recommended for scenarios that require low latency.

Redundancy * Locally-redundant storage (LRS)

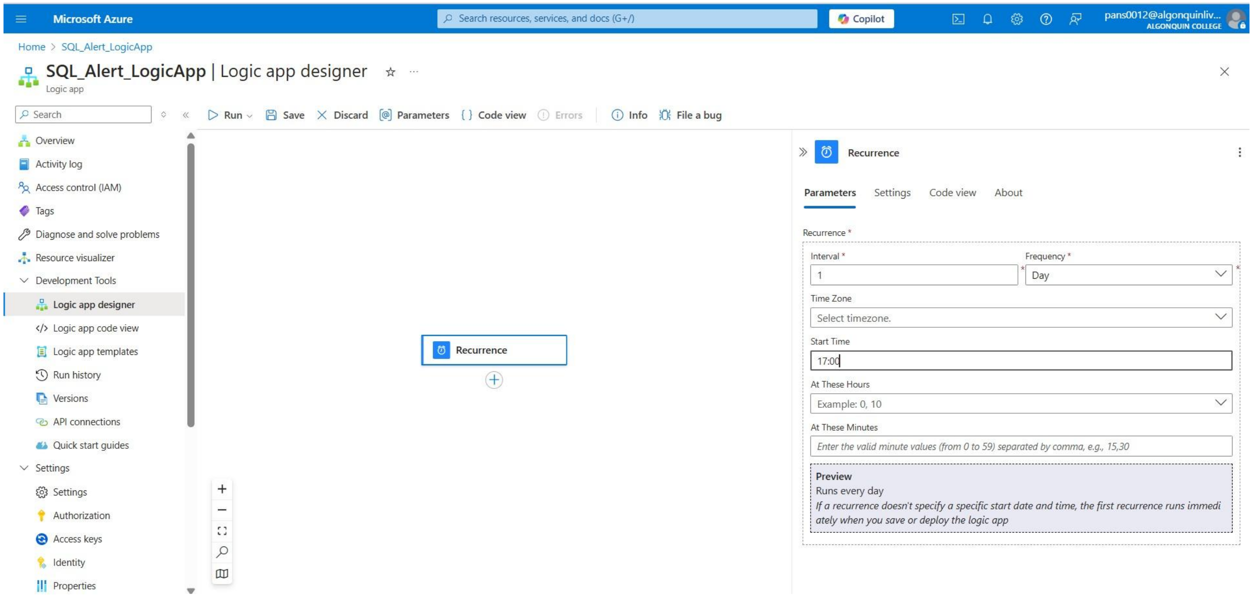
2. Create a sample container and within that container create folder in format of "yyyy-mm-dd"



Authentication method: Access key (Switch to Microsoft Entra user account)
Location: datafiles

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
2025-03-31.docx	31/3/2025, 9:38:58 pm	Hot (Inferred)		Block blob	93.74 KiB	Available

3. Create a trigger to schedule this logic everyday at 6pm



Recurrence *

Interval *: 1

Frequency *: Day

Time Zone: Select timezone.

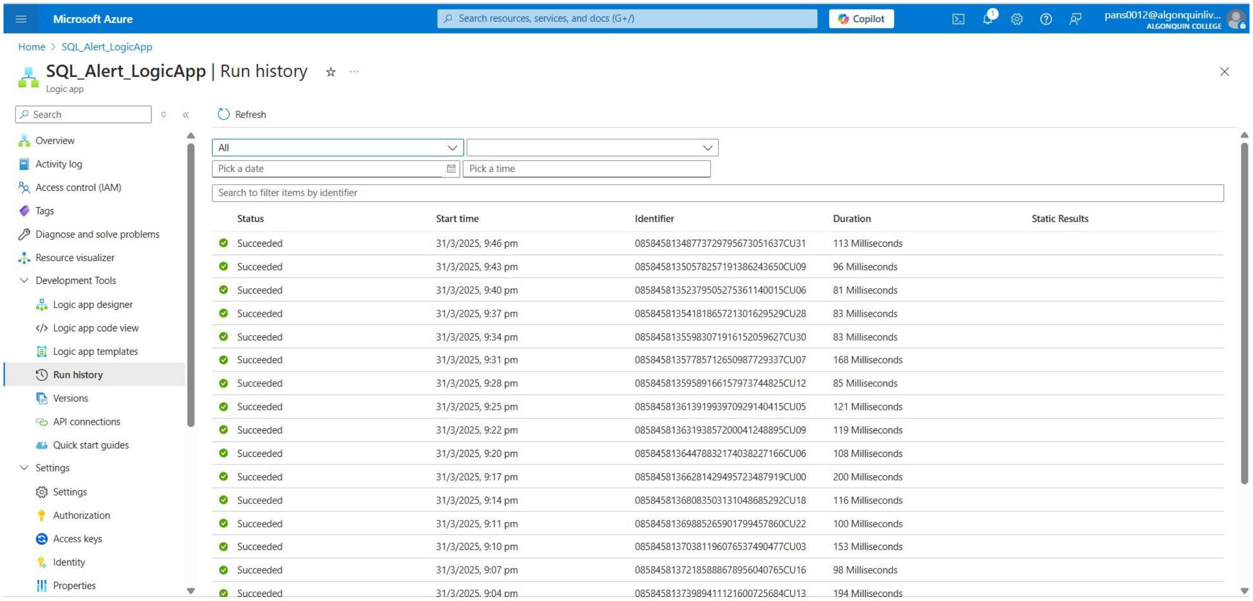
Start Time: 17:00

At These Hours: Example: 0, 10

At These Minutes: *Enter the valid minute values (from 0 to 59) separated by comma, e.g., 15,30*

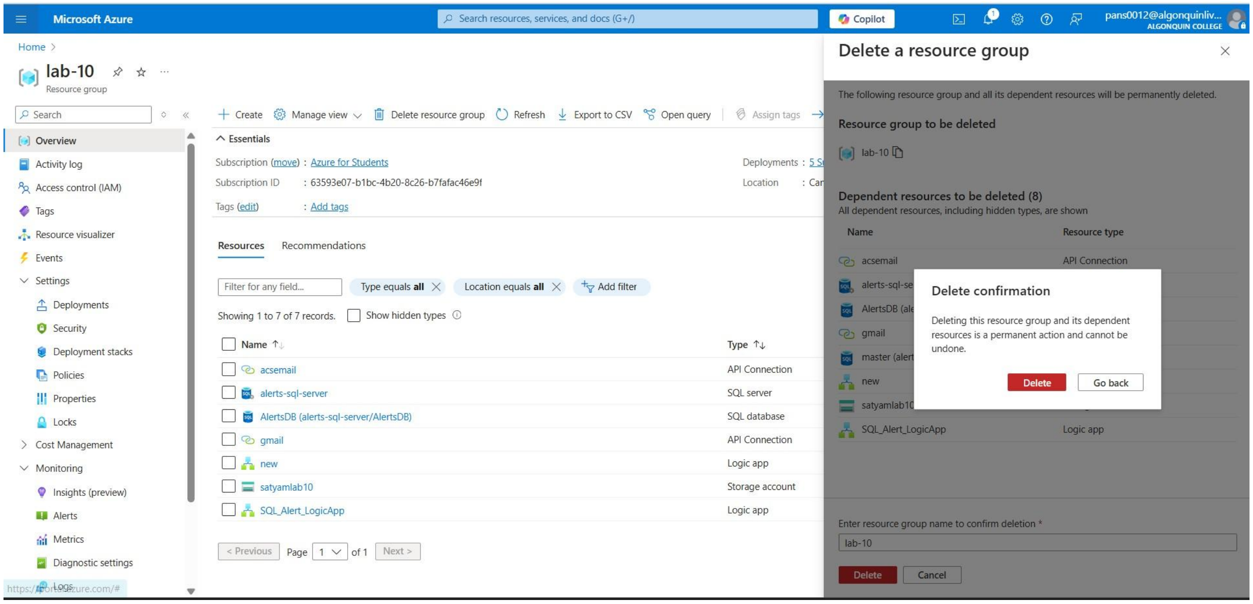
Preview
Runs every day
If a recurrence doesn't specify a specific start date and time, the first recurrence runs immedi ately when you save or deploy the logic app

Task 3: Monitor workflows in azure logic apps



Status	Start time	Identifier	Duration	Static Results
Succeeded	31/3/2025, 9:46 pm	08584581348773729795673051637CU31	113 Milliseconds	
Succeeded	31/3/2025, 9:43 pm	08584581350578257191386243650CU09	96 Milliseconds	
Succeeded	31/3/2025, 9:40 pm	08584581352379505275361140015CU06	81 Milliseconds	
Succeeded	31/3/2025, 9:37 pm	08584581354181865721301629529CU28	83 Milliseconds	
Succeeded	31/3/2025, 9:34 pm	08584581355983071916152059627CU30	83 Milliseconds	
Succeeded	31/3/2025, 9:31 pm	08584581357785712650987729337CU07	168 Milliseconds	
Succeeded	31/3/2025, 9:28 pm	08584581359589166157973744825CU12	85 Milliseconds	
Succeeded	31/3/2025, 9:25 pm	08584581361391993970929140415CU05	121 Milliseconds	
Succeeded	31/3/2025, 9:22 pm	08584581363193857200041248895CU09	119 Milliseconds	
Succeeded	31/3/2025, 9:20 pm	08584581364478832174038227166CU06	108 Milliseconds	
Succeeded	31/3/2025, 9:17 pm	08584581366281429495723487919CU00	200 Milliseconds	
Succeeded	31/3/2025, 9:14 pm	08584581368083503131048685292CU18	116 Milliseconds	
Succeeded	31/3/2025, 9:11 pm	08584581369885265901799457860CU22	100 Milliseconds	
Succeeded	31/3/2025, 9:10 pm	08584581370381196076537490477CU03	153 Milliseconds	
Succeeded	31/3/2025, 9:07 pm	08584581372185888678956040765CU16	98 Milliseconds	
Succeeded	31/3/2025, 9:04 pm	08584581373989411121600725684CU13	194 Milliseconds	

Task4: Clean all the resources created during this lab and record all the steps with screenshots in the lab report.



Subscription (move) : Azure for Students

Subscription ID : 63593e07-b1bc-4b20-8c26-b7fafac46e9f

Tags (edit) : Add tags

Deployments : 5 S

Location : Car

Showing 1 to 7 of 7 records. □ Show hidden types

Name	Type
acsemail	API Connection
alerts-sql-server	SQL server
AlertsDB (alerts-sql-server/AlertsDB)	SQL database
gmail	API Connection
new	Logic app
satyamlab10	Storage account
SQL_Alert_LogicApp	Logic app

Delete a resource group

The following resource group and all its dependent resources will be permanently deleted.

Resource group to be deleted

lab-10

Dependent resources to be deleted (8)

All dependent resources, including hidden types, are shown

Name	Resource type
acsemail	API Connection
alerts-sql-se	
AlertsDB (ale	
gmail	
master (alert	
new	
satyamlab10	
SQL_Alert_LogicApp	Logic app

Delete confirmation

Deleting this resource group and its dependent resources is a permanent action and cannot be undone.

Enter resource group name to confirm deletion *

lab-10