

Output and Screenshots

Create a SQL Database

Azure services



Give the required details

Create SQL Database ...

Microsoft

Basics Networking Security Additional settings Tags Review + create

Create a SQL database with your preferred configurations. Complete the Basics tab then go to Review + Create to provision with smart defaults, or visit each tab to customize. [Learn more](#)

Project details

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

Subscription * ⓘ

Resource group * ⓘ [Create new](#)

Database details

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

Database name * ✓

Server * ⓘ [Create new](#)

Want to use SQL elastic pool? ⓘ ☐ Yes ☒ No

Workload environment ☒ Development ☐ Production

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


Compute + storage * ⓘ **Basic**

Review + create

Next : Networking >

Choose the tier

Configure ...

 Feedback

Service and compute tier

Select from the available tiers based on the needs of your workload. The vCore model provides a wide range of configuration controls and offers Hyperscale and Serverless to automatically scale your database based on your workload needs. Alternately, the DTU model provides set price/performance packages to choose from for easy configuration. [Learn more](#)

Service tier

Basic (For less demanding workloads) 

[Compare service tiers](#)

DTUs [Compare DTU options](#)

5 (Basic)

Data max size (GB)

 2



Cost summary

Basic (Basic)

Cost per DTU (in --)

--

DTUs selected


x 5

ESTIMATED COST / MONTH

-- ..

PLEASE CONTACT YOUR RESELLER

Apply

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 * 


Basic

2 GB storage

[Configure database](#)

Backup storage redundancy

Choose how your PITR and LTR backups are replicated. Geo restore or ability to recover from regional outage is only available when geo-redundant storage is selected.

Backup storage redundancy 

☒ Locally-redundant backup storage
☐ Zone-redundant backup storage
☐ Geo-redundant backup storage

Review + create

Next : Networking >

Click Create

Create SQL Database ...

Microsoft

Basics Networking Security Additional settings Tags Review + create

Product details

SQL database
by Microsoft
[Terms of use](#) | [Privacy policy](#)

Estimated cost per month

-- --

Terms

By clicking "Create", I (a) agree to the legal terms and privacy statement(s) associated with the Marketplace offering(s) listed above; (b) authorize Microsoft to bill my current payment method for the fees associated with the offering(s), with the same billing frequency as my Azure subscription; and (c) agree that Microsoft may share my contact, usage and transactional information with the provider(s) of the offering(s) for support, billing and other transactional activities. Microsoft does not provide rights for third-party offerings. For additional details see [Azure Marketplace Terms](#).

Basics

Subscription	Simplilearn HOL 100500
Resource group	ODL-azure-1105040
Region	East US
Database name	BlogTrackerDb
Server	(new) newsqlserver3058
Authentication method	SQL authentication
Server admin login	admin123
Compute + storage	Basic: 2 GB storage
Backup storage redundancy	Locally-redundant backup storage

Networking

Allow Azure services and resources to access this server	No
Private endpoint	None
Minimum TLS version	1.2

Create

< Previous

[Download a template for automation](#)

Deployment complete

[Home](#) >



Microsoft.SQLDatabase.newDatabaseNewServer_225c0b90020a4279b013f | Overview

Deployment

Search

Delete Cancel Redeploy Download Refresh

Overview

Inputs

Outputs

Template

✓ Your deployment is complete

Deployment name : Microsoft.SQLDatabase.newDatabaseNewServer_225c0b90020a4279b013f
Subscription : Simplilearn HOL 100500
Resource group : ODL-azure-1105040

Start time : 9/26/2023, 10:13:44 AM
Correlation ID : 6c2c8709-0645-4303-a55e-8ca3c8829b59

> Deployment details

∨ Next steps

Go to resource

Set firewall


Public access Private access Connectivity


Public network access

Public Endpoints allow access to this resource through the internet using a public IP address. An application or resource that is granted access with the following Public network access

☐ Disable

☒ Selected networks

 Connections from the IP addresses configured in the Firewall rules section below will have access to this database. By d

 Please save public network access value before adding new virtual networks.

Virtual networks

Allow virtual networks to connect to your resource using service endpoints. [Learn more](#)

[+](#) Add a virtual network rule

Rule	Virtual network	Subnet	Address range	Endpoint status	Resource group	Subscription	State
------	-----------------	--------	---------------	-----------------	----------------	--------------	-------

Firewall rules

Allow certain public internet IP addresses to access your resource. [Learn more](#)

[+](#) Add your client IPv4 address (49.37.217.234) [+](#) Add a firewall rule

Rule name	Start IPv4 address	End IPv4 address	
ClientIPAddress_2023-9-26_10-17-20	49.37.217.0	49.37.217.234	

Exceptions

☒ Allow Azure services and resources to access this server 

Save

Discard

Use the SQL authentication credentials

Query editor (preview) is a tool to run SQL queries against Azure SQL Database in the Azure portal. It is designed for lightweight querying and object exploration in your database. For more information and troubleshooting, [Learn more](#)



Welcome to SQL Database Query Editor

SQL server authentication

Login *

admin123

Password *

OK

Microsoft Entra authentication

Continue as odl_user_1105040@simpilile...

OR

Query Window

BlogTrackerDb (admin123)

Showing limited object explorer here. For full capability please click here to open Azure Data Studio.

- Tables
- Views
- Stored Procedures

Query 1 X

Run Cancel query Save query Export data as Show only Editor

1

Results Messages

Search to filter items...

As I created the table already in SQL Server I used the same query to create this

BlogTrackerDb (admin123)

Showing limited object explorer here. For full capability please click here to open Azure Data Studio.

- Tables
 - dbo.AdminInfo
 - dbo.BlogInfo
 - dbo.EmpInfo
- Views
- Stored Procedures

Query 1 X

Run Cancel query Save query Export data as Show only Editor

```
1 CREATE TABLE [dbo].[BlogInfo](
2     [BlogInfoId] [int] IDENTITY(1,1) NOT NULL,
3     [Title] [nvarchar](50) NOT NULL,
4     [Subject] [nvarchar](50) NOT NULL,
5     [DateOfCreation] [datetime2](7) NOT NULL,
6     [BlogUrl] [nvarchar](max) NOT NULL,
7     [EmpEmailId] [nvarchar](max) NOT NULL,
8     CONSTRAINT [PK_BlogInfo] PRIMARY KEY CLUSTERED
9     (
10         [BlogInfoId] ASC
11     )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
12     ) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
13
14 CREATE TABLE [dbo].[AdminInfo](
```

Connection string

BlogTrackerDb (newsqserver3058/BlogTrackerDb) | Connection strings ☆ ...

ADO.NET JDBC ODBC PHP Go

ADO.NET (Microsoft Entra passwordless authentication)

Microsoft.Data.SqlClient Quickstart
Entity Framework Core Quickstart

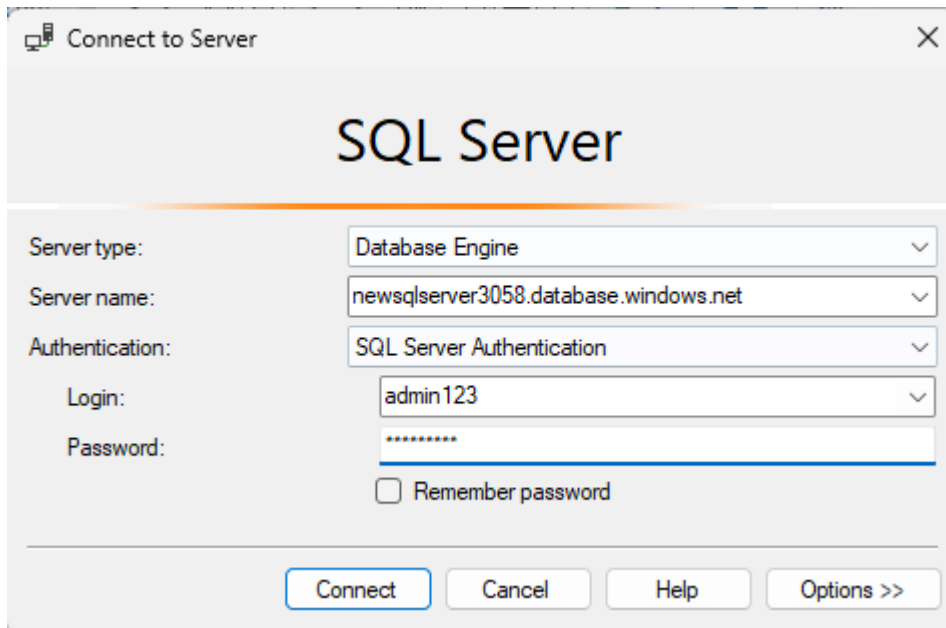
```
Server=tcp:newsqserver3058.database.windows.net,1433;Initial Catalog=BlogTrackerDb;Encrypt=True;TrustServerCertificate=False;Connection Timeout=30;Authentication="Active Directory Default"
```

ADO.NET (SQL authentication)

```
Server=tcp:newsqserver3058.database.windows.net,1433;Initial Catalog=BlogTrackerDb;Persist Security Info=False;User ID=admin123;Password={your_password};MultipleActiveResultSets=False;Encrypt=True;TrustServerCertificate=False
```

Download ADO.NET driver for SQL server

Connect the server name in the Azure SQL database



Connect to Server

SQL Server

Server type: Database Engine

Server name: newsqlserver3058.database.windows.net

Authentication: SQL Server Authentication

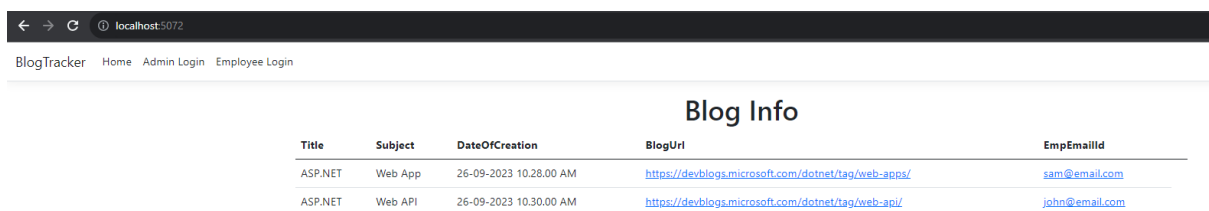
Login: admin123

Password:

☐ Remember password

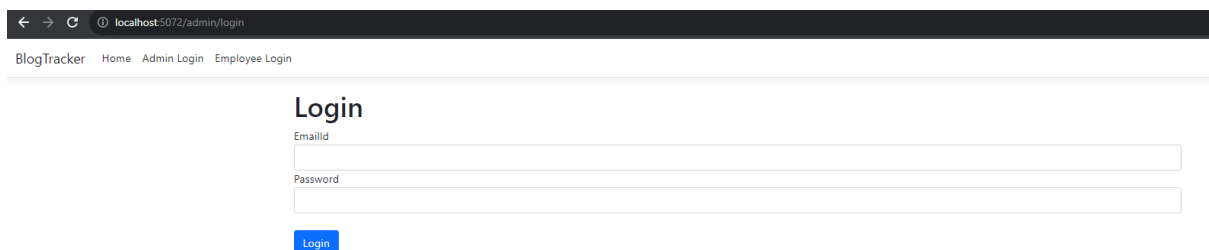
Connect Cancel Help Options >>

Blog Landing page



Title	Subject	DateOfCreation	BlogUrl	EmpEmailId
ASP.NET	Web App	26-09-2023 10:28:00 AM	https://devblogs.microsoft.com/dotnet/tag/web-apps/	sam@email.com
ASP.NET	Web API	26-09-2023 10:30:00 AM	https://devblogs.microsoft.com/dotnet/tag/web-api/	john@email.com

Admin Login page



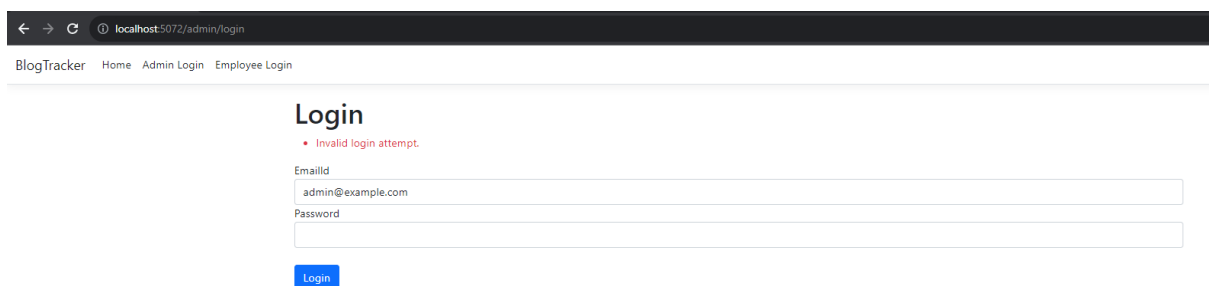
Login

EmailId

Password

Login

Incorrect password



Login

Invalid login attempt.

EmailId

admin@example.com

Password

Login

After putting correct password, adding new employee page

[←](#) [→](#) [↻](#) [localhost:5072/EmplInfoes](#)

[BlogTracker](#) [Home](#) [Admin Login](#) [Employee Login](#)

Index

[Create New](#) [Logout](#)

EmailId	Name	DateOfJoining	PassCode	
sam@email.com	Sam	26-09-2023 10:26:00 AM	1234	Edit Details Delete
john@email.com	John	26-09-2023 10:29:00 AM	4567	Edit Details Delete

Adding new employee

[←](#) [→](#) [↻](#) [localhost:5072/EmplInfoes/Create](#)

[BlogTracker](#) [Home](#) [Admin Login](#) [Employee Login](#)

Create EmplInfo

EmailId

arun@email.com

Name

Arun

DateOfJoining

26-09-2023 10:34 AM

PassCode

7890

[Create](#)

[Back to List](#)

After adding new Employee

[←](#) [→](#) [↻](#) [localhost:5072/EmplInfoes](#)

[BlogTracker](#) [Home](#) [Admin Login](#) [Employee Login](#)

Index

[Create New](#) [Logout](#)

EmailId	Name	DateOfJoining	PassCode	
sam@email.com	Sam	26-09-2023 10:26:00 AM	1234	Edit Details Delete
john@email.com	John	26-09-2023 10:29:00 AM	4567	Edit Details Delete
arun@email.com	Arun	26-09-2023 10:34:00 AM	7890	Edit Details Delete

Employee Login

[←](#) [→](#) [↻](#) [localhost:5072/EmplInfoes/EmployeeLogin](#)

[BlogTracker](#) [Home](#) [Admin Login](#) [Employee Login](#)

Employee Login

Email

Passcode

[Login](#)

[←](#) [→](#) [↺](#) [localhost:5072/BlogInfos/Create](#)

[BlogTracker](#) [Home](#) [Admin Login](#) [Employee Login](#)

Create

BlogInfo


Title

Azure

Subject

Web App Azure

DateOfCreation

26-09-2023 10:36 AM 

BlogUrl

https://azure.microsoft.com/en-us/blog/product/web-a

EmpEmailId

arun@email.com

Create

[Back to List](#)

[←](#)
[→](#)
[🔍 localhost:5072](#)

[BlogTracker](#)
[Home](#)
[Admin Login](#)
[Employee Login](#)

Blog Info

Title	Subject	DateOfCreation	BlogUrl	EmpEmailId
ASP.NET	Web App	26-09-2023 10:28:00 AM	https://devblogs.microsoft.com/dotnet/tag/web-apps/	sam@email.com
ASP.NET	Web API	26-09-2023 10:30:00 AM	https://devblogs.microsoft.com/en-us/blog/product/web-api/	john@email.com
Azure	Web App Azure	26-09-2023 10:36:00 AM	https://azure.microsoft.com/en-us/blog/product/web-apps/	arun@email.com

Test Explorer

Search (Ctrl+I)

Test run finished: 3 Tests (3 Passed, 0 Failed, 0 Skipped) run in 454 ms

0 Warnings 0 Errors

Test	Duration
BlogTracker (3)	235 ms
BlogTracker (3)	235 ms
TestClass (3)	235 ms
BlogInfoTest	7 ms
Create_ValidEmplInfo_RedirectsT...	228 ms
EmplInfoTest	< 1 ms

Group Summary

BlogTracker

Tests in group: 3


Total Duration: 235 ms

Outcomes

3 Passed


In the API after changing some values

```
[
  {
    "blogInfoId": 1,
    "title": "New Blog",
    "subject": "ASP.NET",
    "dateOfCreation": "2023-09-25T13:52:00",
    "blogUrl": "https://devblogs.microsoft.com/dotnet/category/aspnet/",
    "empEmailId": "sam@email.com"
  },
  {
    "blogInfoId": 2,
    "title": "Another Blog",
    "subject": "Web API",
    "dateOfCreation": "2023-09-25T19:08:00",
    "blogUrl": "https://devblogs.microsoft.com/dotnet/tag/web-api/",
    "empEmailId": "Shankar@email.com"
  },
  {
    "blogInfoId": 3,
    "title": "ASP.NET",
    "subject": "Web App",
    "dateOfCreation": "2023-09-26T09:13:00",
    "blogUrl": "https://devblogs.microsoft.com/dotnet/tag/web-apps/",
    "empEmailId": "arun@email.com"
  }
]
```

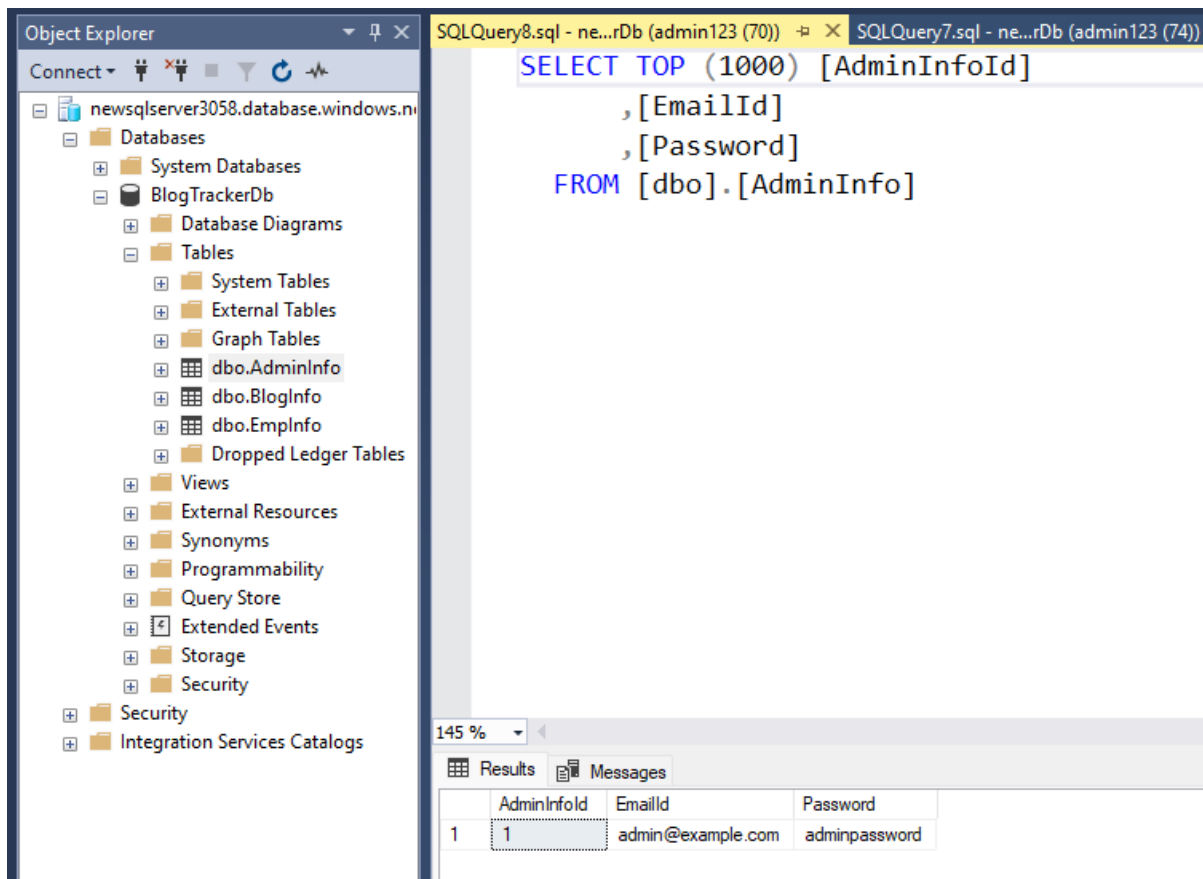
 Download

Response body

```
{
  "empInfoId": 1,
  "emailId": "sam@email.com",
  "name": "Sam",
  "dateOfJoining": "2023-09-25T13:23:00",
  "passCode": 1234
},
{
  "empInfoId": 2,
  "emailId": "john@email.com",
  "name": "John",
  "dateOfJoining": "2023-09-25T14:00:00",
  "passCode": 2345
},
{
  "empInfoId": 3,
  "emailId": "shankar@email.com",
  "name": "Shankar",
  "dateOfJoining": "2023-09-25T16:29:00",
  "passCode": 4567
},
{
  "empInfoId": 4,
  "emailId": "arun@email.com",
  "name": "Arun",
  "dateOfJoining": "2023-09-26T09:10:00",
  "passCode": 4567
}
```

 Download

In SQL server after connected with Azure SQL Database



The screenshot shows the SQL Server Enterprise Edition interface. On the left, the Object Explorer displays the database structure for 'newsqserver3058.database.windows.net'. The 'Databases' folder is expanded, showing 'System Databases', 'BlogTrackerDb', 'Database Diagrams', 'Tables', 'Views', 'External Resources', 'Synonyms', 'Programmability', 'Query Store', 'Extended Events', 'Storage', 'Security', and 'Integration Services Catalogs'. The 'Tables' folder is expanded, showing 'System Tables', 'External Tables', 'Graph Tables', 'dbo.AdminInfo', 'dbo.BlogInfo', 'dbo.EmplInfo', and 'Dropped Ledger Tables'. The 'dbo.AdminInfo' table is selected.

The main window shows a SQL query in the 'SQLQuery8.sql' file:

```
SELECT TOP (1000) [AdminInfoId]
,[EmailId]
,[Password]
FROM [dbo].[AdminInfo]
```

The query results are displayed in the 'Results' tab, showing a single row with the following data:

AdminInfoId	EmailId	Password
1	admin@example.com	adminpassword

Object Explorer

newsqlserver3058.database.windows.net

Databases

System Databases

BlogTrackerDb

Database Diagrams

Tables

System Tables

External Tables

Graph Tables

dbo.AdminInfo

dbo.BlogInfo

dbo.EmpInfo

Dropped Ledger Tables

Views

External Resources

Synonyms

Programmability

Query Store

Extended Events

Storage

Security

Integration Services Catalogs

SQLQuery8.sql - ne...rDb (admin123 (70))

SQLQuery7.sql - ne...rDb (admin123 (74))

```
SELECT TOP (1000) [EmpInfoId]
, [EmailId]
, [Name]
, [DateOfJoining]
, [PassCode]
FROM [dbo].[EmpInfo]
```

145 %

Results Messages

	EmpInfoId	EmailId	Name	DateOfJoining	PassCode
1	1	sam@email.com	Sam	2023-09-26 10:26:00.0000000	1234
2	2	john@email.com	John	2023-09-26 10:29:00.0000000	4567
3	3	arun@email.com	Arun	2023-09-26 10:34:00.0000000	7890

Object Explorer

newsqlserver3058.database.windows.net

Databases

System Databases

BlogTrackerDb

Database Diagrams

Tables

System Tables

External Tables

Graph Tables

dbo.AdminInfo

dbo.BlogInfo

dbo.EmpInfo

Dropped Ledger Tables

Views

External Resources

Synonyms

Programmability

Query Store

Extended Events

Storage

Security

Integration Services Catalogs

SQLQuery9.sql - ne...rDb (admin123 (51))

SQLQuery8.sql - ne...rDb (admin123 (70))

SQLQuery7.sql - ne...rDb (admin123 (74))

```
SELECT TOP (1000) [BlogInfoId]
, [Title]
, [Subject]
, [DateOfCreation]
, [BlogUrl]
, [EmpEmailId]
FROM [dbo].[BlogInfo]
```

145 %

Results Messages

	BlogInfoId	Title	Subject	DateOfCreation	BlogUrl	EmpEmailId
1	1	ASP.NET	Web App	2023-09-26 10:28:00.0000000	https://devblogs.microsoft.com/dotnet/tag/web-ap...	sam@email.com
2	2	ASP.NET	Web API	2023-09-26 10:30:00.0000000	https://devblogs.microsoft.com/dotnet/tag/web-api/	john@email.com
3	3	Azure	Web App Azure	2023-09-26 10:36:00.0000000	https://azure.microsoft.com/en-us/blog/product/w...	arun@email.com

Docker running

AppServiceLayer	8a413bb77227	appservicelayer.dev	Running	0%	32768.80	2 hours ago	
BlogTracker	c5907712d37b	blogtracker.dev	Running	0%	32769.80	2 hours ago	