leaplogic
an IMPETUS product

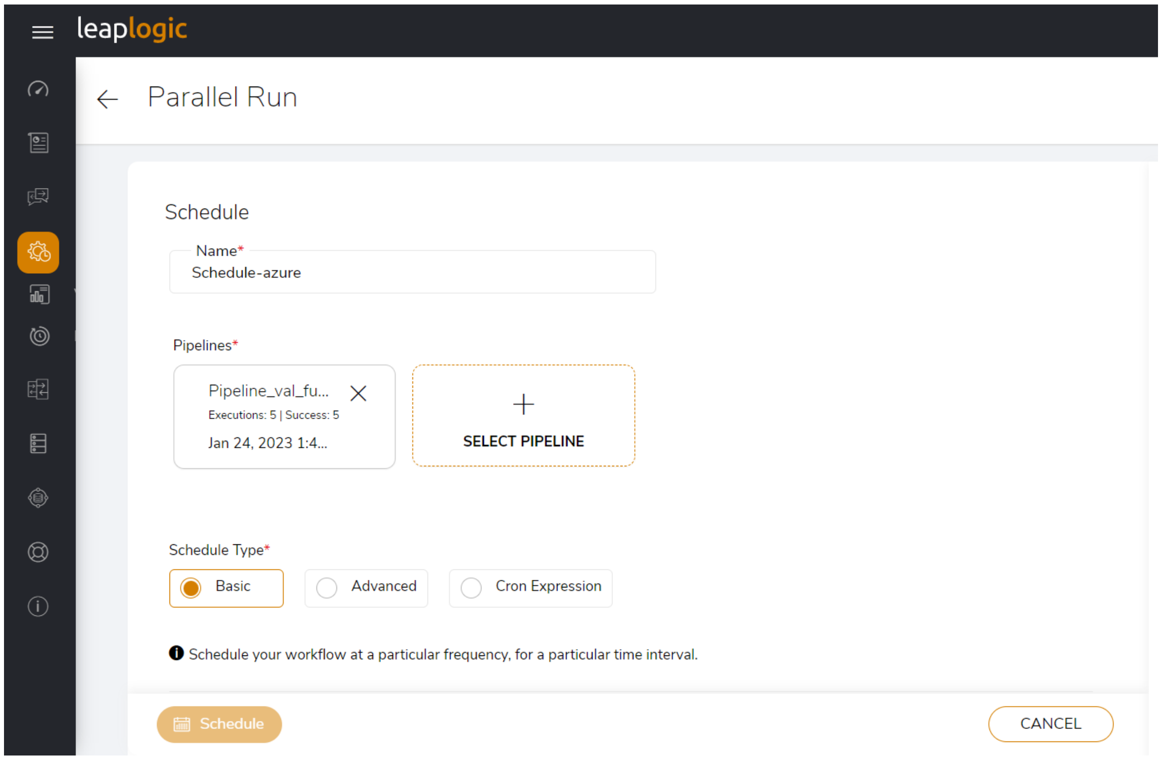
Pipeline Scheduling with Azure Functions

Contents

1. Automatic Push

This option allows LeapLogic to generate Azure Function codes dynamically and push that to your selected Azure cloud environment to generate Azure functions. It also allows to trigger the pipeline execution or schedule pipelines. You can provide the credentials of the respective cloud environment in the given format.

1. Go to Operationalization > Parallel Run.
2. Select the pipelines that need to be scheduled.



Name*: Schedule-azure

Pipelines*: Pipeline_val_fu... Executions: 5 | Success: 5 Jan 24, 2023 1:4...

Schedule Type*: [✓ Basic] Advanced Cron Expression

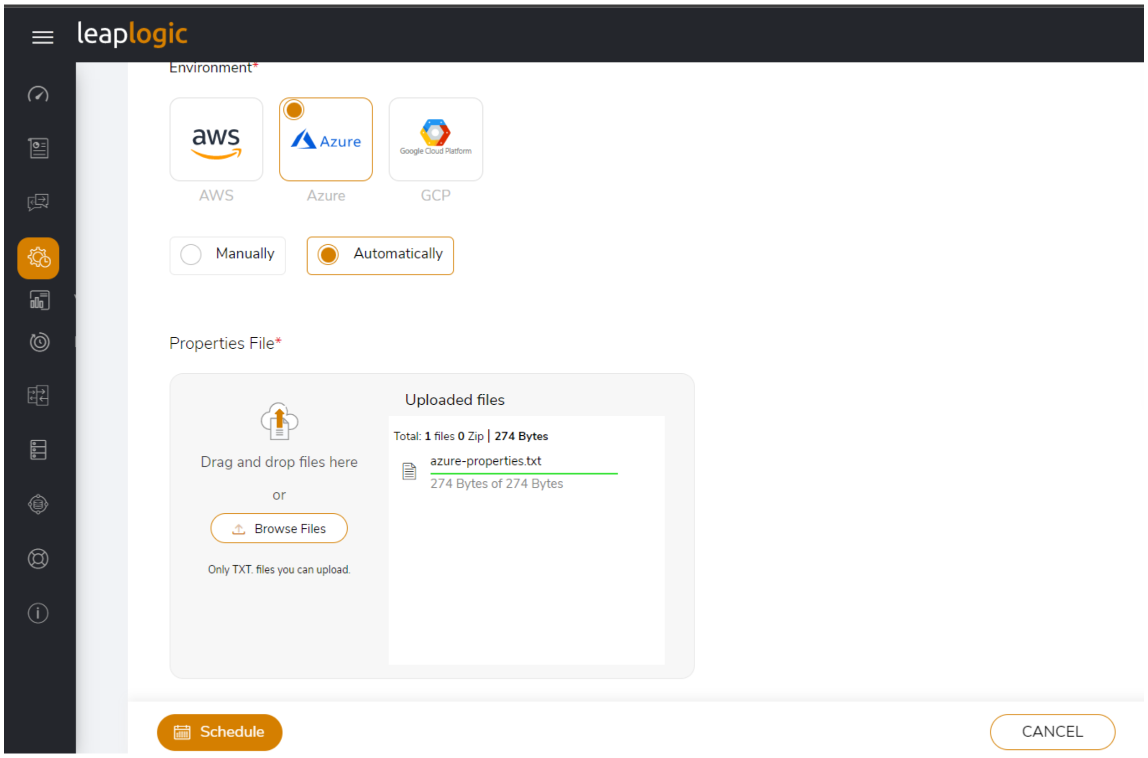
Schedule your workflow at a particular frequency, for a particular time interval.

3. Click **Advanced** as schedule type.
4. Select Environment for advance trigger. Select Azure.
5. Click **Automatically**. You can upload the properties file as per the below format.

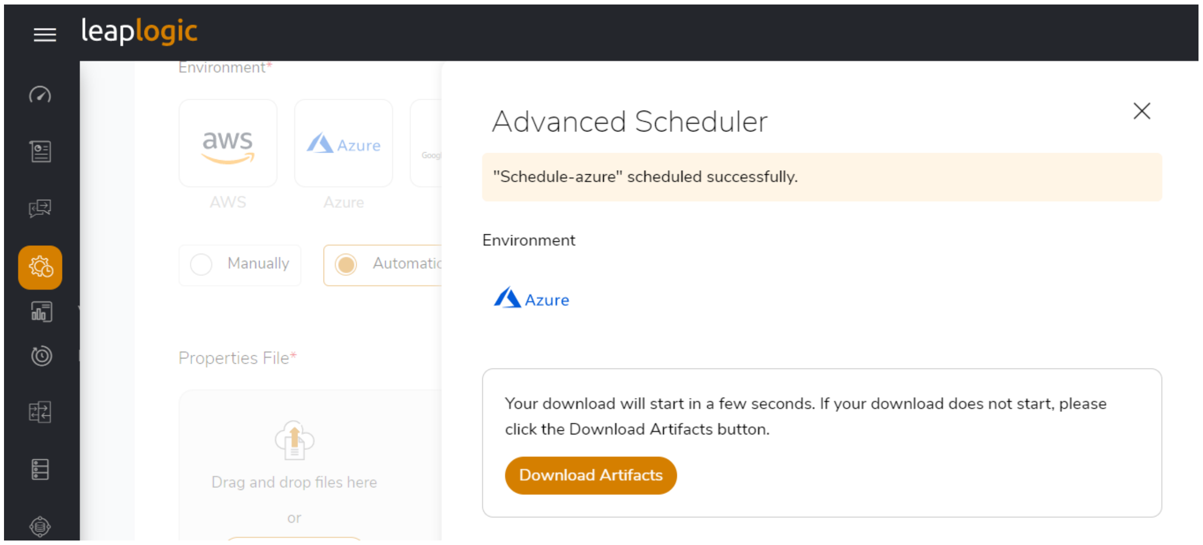
```
clientId=<Azure Client Id>
clientSecret=<Azure Client Secret>
tenantId=<Azure Tenant Id>
subscriptionId=<Azure Subscription Id>
resourceGroupName=<Resource group name under which Function App is created>
FunctionAppName=<Function App Name>
```

Note

To create an Azure Function app, refer to Creating Azure Function App.



6. Click **Schedule**. This generates the Azure functions on the cloud environment with the provided Azure environment details.



"Schedule-azure" scheduled successfully.

Environment

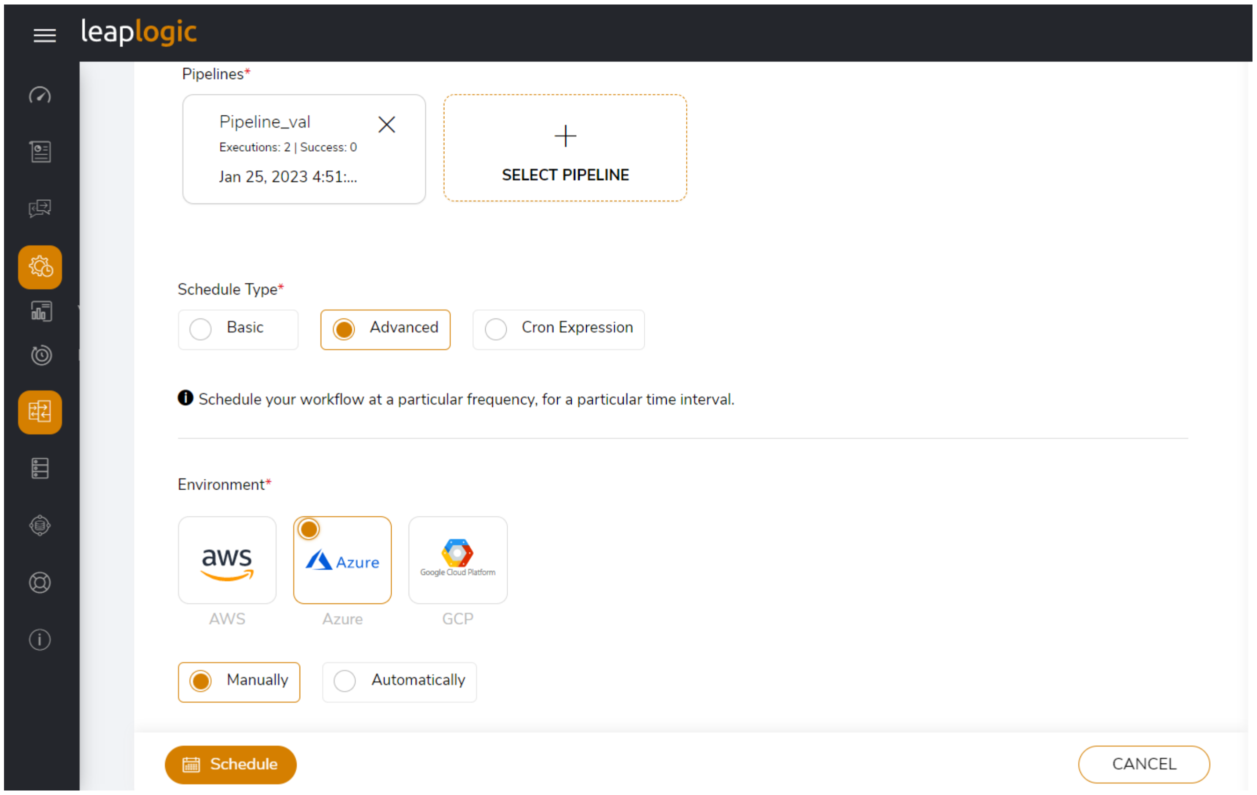
Azure

Your download will start in a few seconds. If your download does not start, please click the Download Artifacts button.

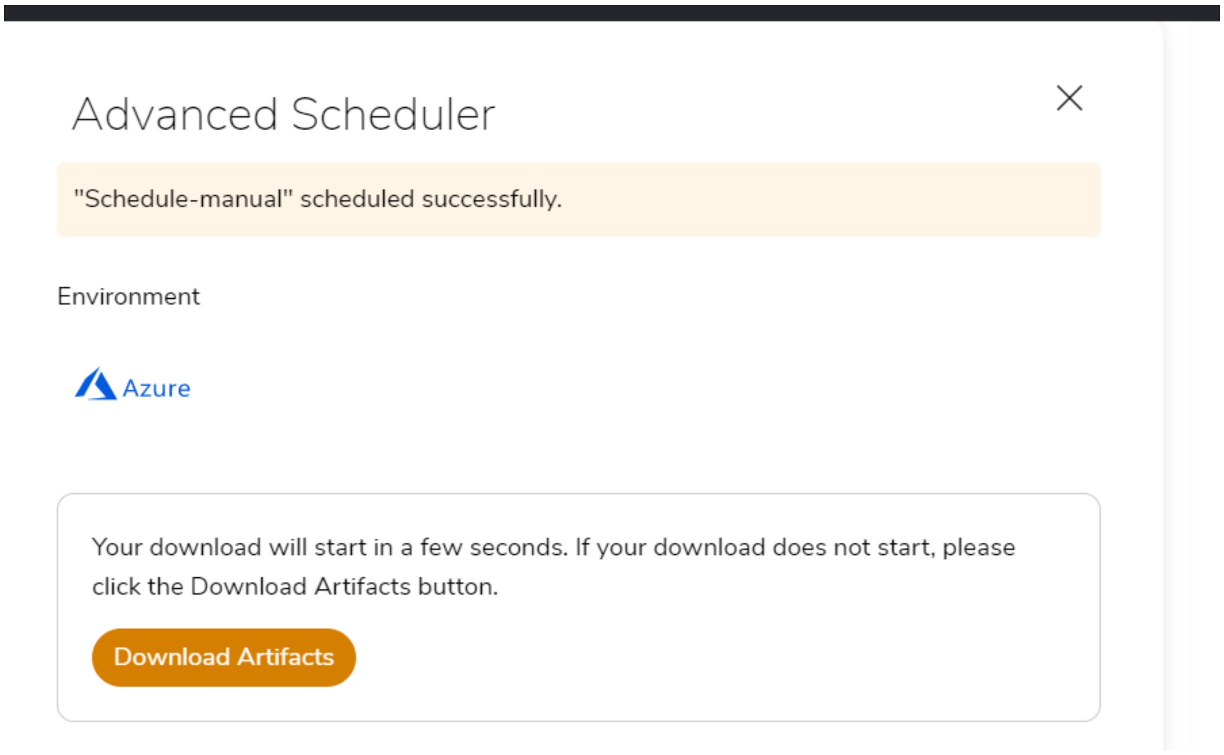
2. Manual

This option allows LeapLogic to generate the Azure function codes dynamically. You can download the generated code in zip format and generate the Azure functions manually.

1. Go to Operationalization > Parallel Run.
2. Select the pipelines to schedule.
3. Click **Advanced** as schedule type.
4. Select Environment for advance trigger and then select **Azure**.
5. Click **Manually.**



6. Click **Schedule**. The download artifact option appears.


Advanced Scheduler

"Schedule-manual" scheduled successfully.

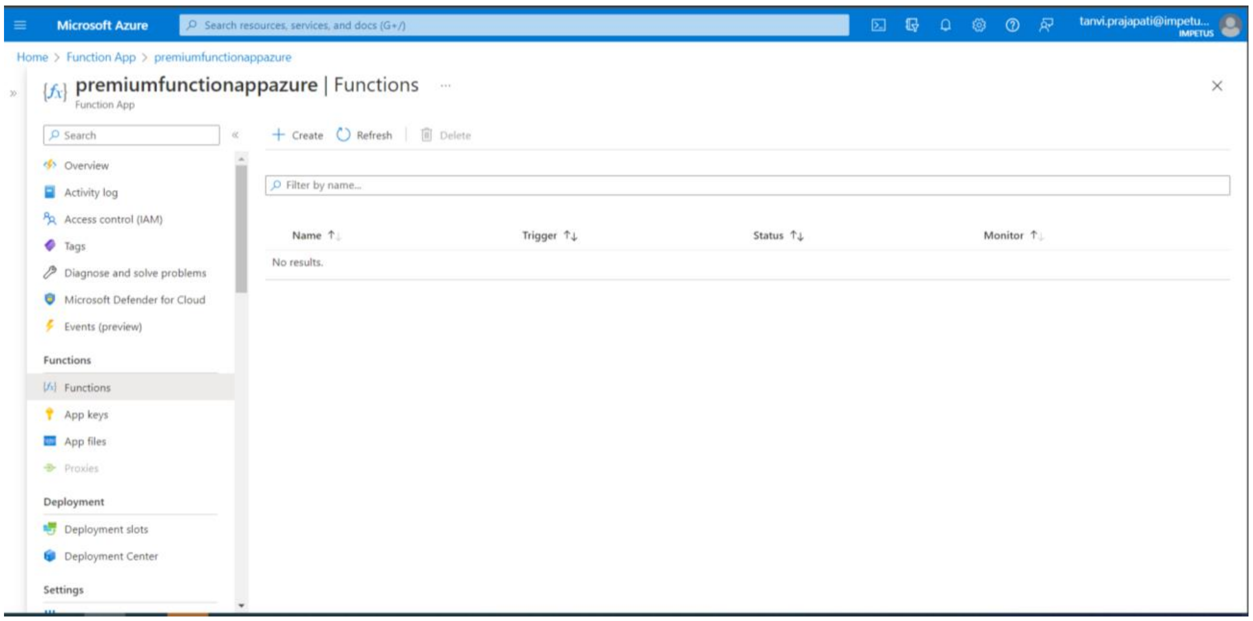
Environment

Your download will start in a few seconds. If your download does not start, please click the Download Artifacts button.

7. Download the zip file.

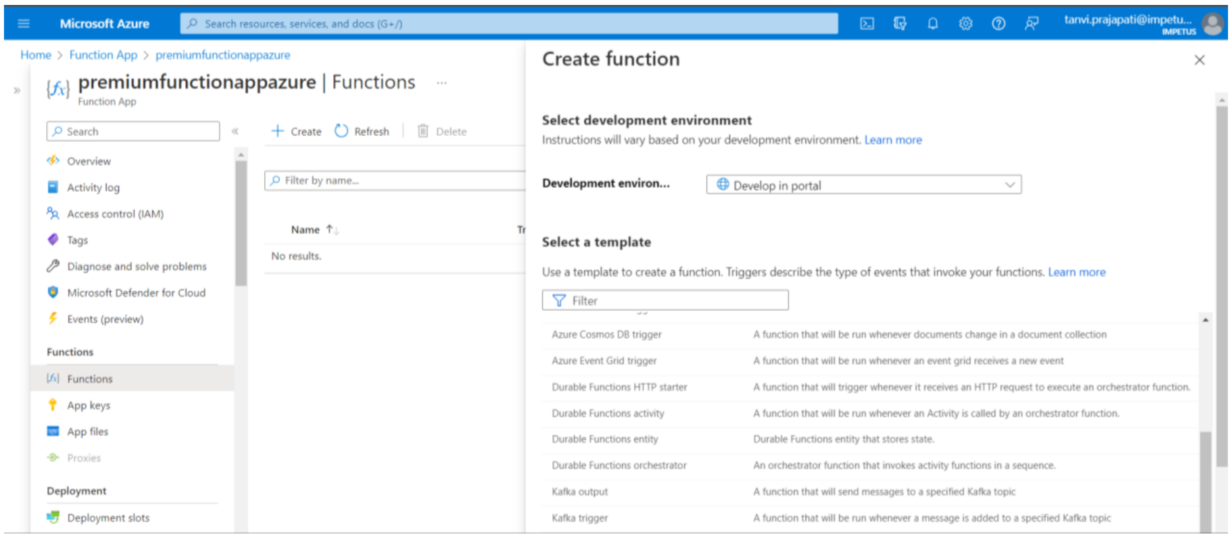
2.1 Creating Azure Functions Manually

1. Go to the Azure Function App console under which the functions need to be created.
2. Click **Functions** on the left navigation pane.
3. Click **Create.**



Name	Trigger	Status	Monitor
No results.			

4. In Development Environment, select **Develop in portal**.



Microsoft Azure | Search resources, services, and docs (G+/) | tanvi.prajapati@impetu... IMPETUS

Home > Function App > premiumfunctionappazure

premiumfunctionappazure | Functions ...
Function App

Create function

Select development environment

Instructions will vary based on your development environment. Learn more

Development environ... Develop in portal

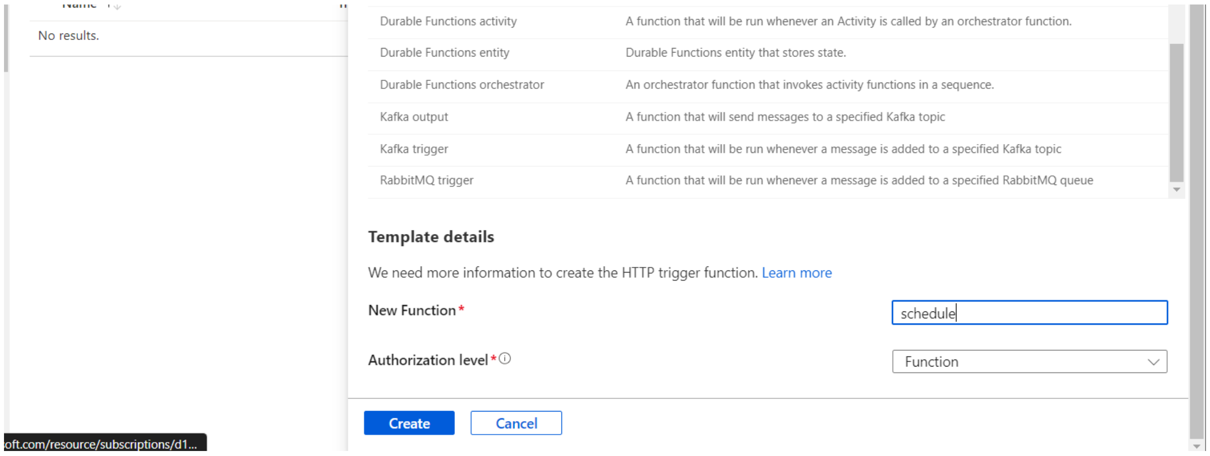
Select a template

Use a template to create a function. Triggers describe the type of events that invoke your functions. Learn more

Filter

Azure Cosmos DB trigger	A function that will be run whenever documents change in a document collection
Azure Event Grid trigger	A function that will be run whenever an event grid receives a new event
Durable Functions HTTP starter	A function that will trigger whenever it receives an HTTP request to execute an orchestrator function.
Durable Functions activity	A function that will be run whenever an Activity is called by an orchestrator function.
Durable Functions entity	Durable Functions entity that stores state.
Durable Functions orchestrator	An orchestrator function that invokes activity functions in a sequence.
Kafka output	A function that will send messages to a specified Kafka topic
Kafka trigger	A function that will be run whenever a message is added to a specified Kafka topic

5. Provide a New Function name same as the schedule name.



No results.

Durable Functions activity	A function that will be run whenever an Activity is called by an orchestrator function.
Durable Functions entity	Durable Functions entity that stores state.
Durable Functions orchestrator	An orchestrator function that invokes activity functions in a sequence.
Kafka output	A function that will send messages to a specified Kafka topic
Kafka trigger	A function that will be run whenever a message is added to a specified Kafka topic
RabbitMQ trigger	A function that will be run whenever a message is added to a specified RabbitMQ queue

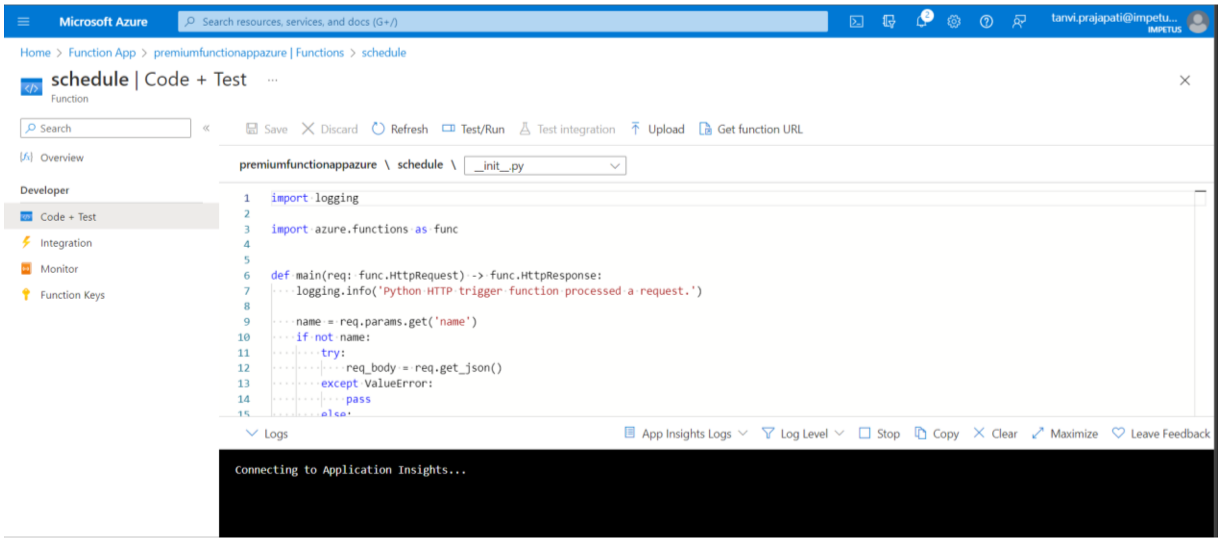
Template details

We need more information to create the HTTP trigger function. Learn more

New Function * schedule

Authorization level * Function

6. Extract the zip file downloaded at the time of the schedule.
7. Click **Upload** and then upload the __init__.py and Certificate.pem files.
8. Azure will not be able to parse the Certificate.pem file. Open the Certificate.pem file in notepad, copy the content, and paste it in the Certificate.pem file in the Azure functions.



```
import logging

import azure.functions as func


def main(req: func.HttpRequest) -> func.HttpResponse:
    logging.info('Python HTTP trigger function processed a request.')

    name = req.params.get('name')
    if not name:
        try:
            req_body = req.get_json()
        except ValueError:
            pass
        else:
```

```
Connecting to Application Insights...
```

This generates the Azure functions.

Note

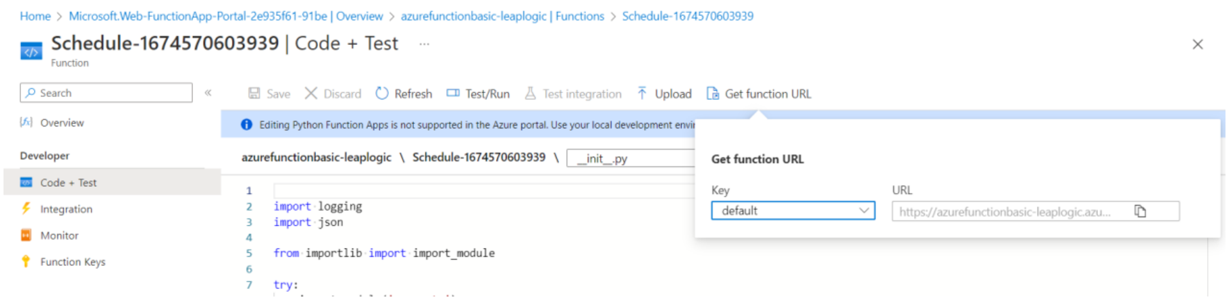
To create functions through the Azure portal, you need to use the Functions Premium Plan while creating Function App.

2.2 Using Azure Functions to Schedule/Execute Pipeline

You can now execute or schedule the pipelines as per your requirements by triggering Azure functions with appropriate JSON. You can also provide its credentials in JSON format to authorize/authenticate beforehand.

For example, execute an Azure function by triggering REST endpoint generated.

1. You can find the Trigger URL as shown below.


Home > Microsoft.Web-FunctionApp-Portal-2e935f61-91be | Overview > azurefunctionbasic-leaplogic | Functions > Schedule-1674570603939
Schedule-1674570603939 | Code + Test
Function
Search
Overview
Developer
Code + Test
Integration
Monitor
Function Keys
Save
Discard
Refresh
Test/Run
Test integration
Upload
Get function URL
azurefunctionbasic-leaplogic \ Schedule-1674570603939 \ __init__.py
Get function URL
Key
default
URL
https://azurefunctionbasic-leaplogic.azu...

Editing Python Function Apps is not supported in the Azure portal. Use your local development envi

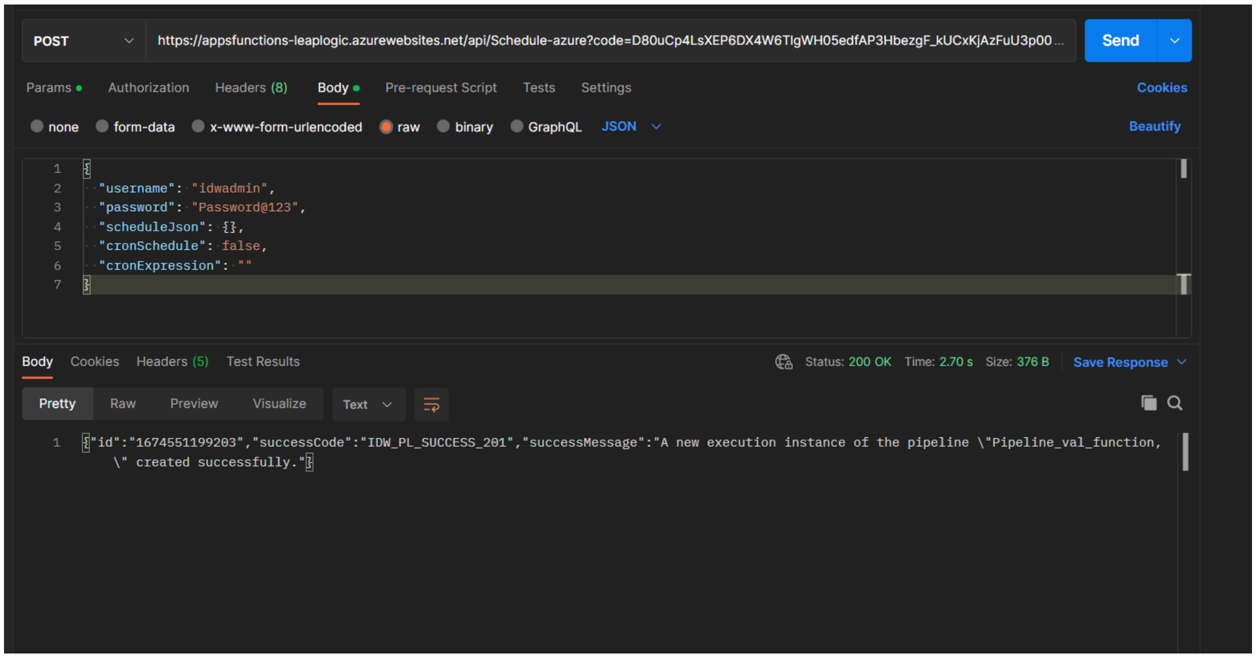
```

import logging
import json

from importlib import import_module

try:
```

2. Use this URL to trigger the pipeline with the appropriate JSON.


POST https://appsfunctions-leaplogic.azurewebsites.net/api/Schedule-azure?code=D80uCp4LsXEP6DX4W6TlgWH05edfAP3HbezgF_kUCxKjAzFuU3p00...

```
{
  "username": "idwadmin",
  "password": "Password@123",
  "scheduleJson": {},
  "cronSchedule": false,
  "cronExpression": ""
}
```

Status: 200 OK Time: 2.70 s Size: 376 B

```
{"id":"1674551199203","successCode":"IDW_PL_SUCCESS_201","successMessage":"A new execution instance of the pipeline \"Pipeline_val_function,
    \" created successfully."}
```

Note

For HTTPS deployment, upload the Certificate.pem file. The file is not readable until it is explicitly saved. So, you need to go to the created Azure function and then choose the Certificate.pem file and save it explicitly.

i. **JSON to execute pipelines**

```
{
 "username": "idwadmin",
 "password": "Password@123",
 "scheduleJson": {},
 "cronSchedule": false,
 "cronExpression": ""
}
```

ii. **JSON to schedule with basic details**

```
{
 "username": "idwadmin",
 "password": "Password@123",
 "scheduleJson": {
  "startDate": "2022-12-27",
  "endDate": "2022-12-27",
  "startTime": "18:05",
  "endTime": "18:06",
  "minutes": 0,
  "timezone": "Asia/Calcutta",
  "frequency": "NONE"
 },
  "cronSchedule": false,
  "cronExpression": ""
}
```

Note

Frequency can be NONE (Once), DAILY, WEEKLY, MONTHLY, YEARLY, and CUSTOM. CUSTOM allows you to specify an interval of minutes for the schedule.

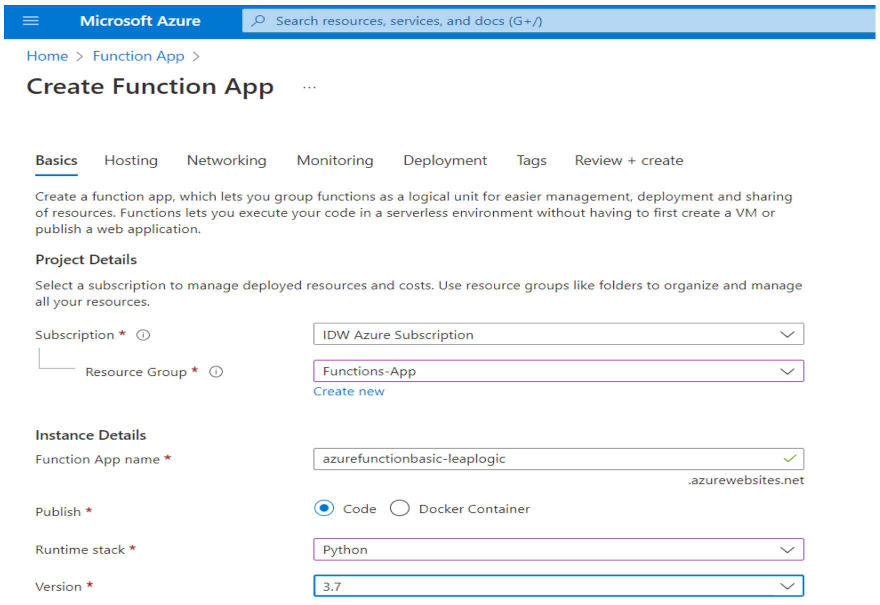
iii. **JSON to schedule with cron expression**

```
{
  "username": "idwadmin",
  "password": "Password@123",
  "scheduleJson": {},
  "cronSchedule": true,
  "cronExpression": "0 30 18 27 12 ? 2022"
}
```

3. You can also curl the API URL.
4. You can also integrate the API URL in this application.

2.3 Creating Azure Function App

1. Go to Function App in the Azure console.
2. In **Runtime stack**, select *Python*.
3. In **Version**, select *3.7* with the appropriate region.



Create Function App

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

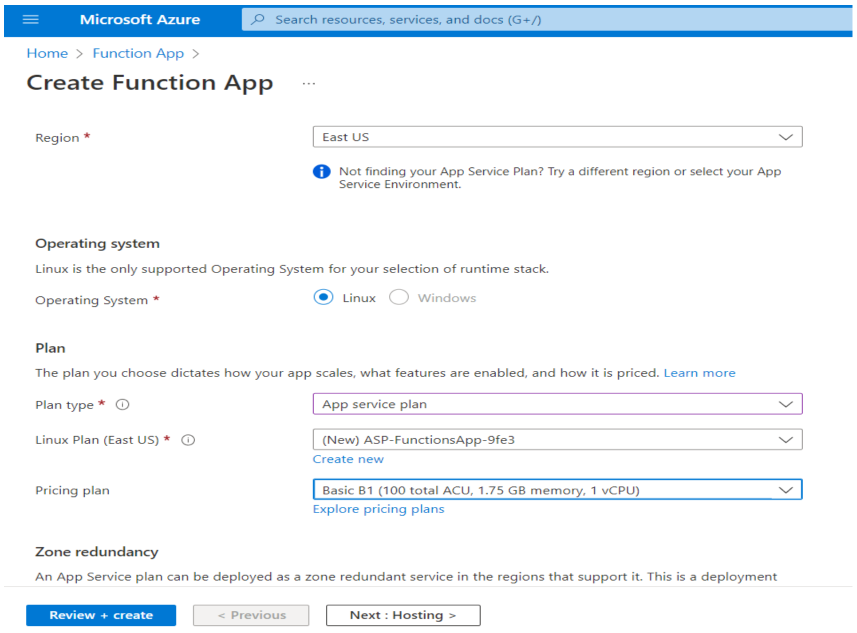
Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *: IDW Azure Subscription
Resource Group *: Functions-App
Create new

Instance Details

Function App name *: azurefunctionbasic-leaplogic .azurewebsites.net
Publish *: [✓ Code] Docker Container
Runtime stack *: Python
Version *: 3.7

4. In **Operating System**, select *Linux* and choose the appropriate plan type.



Create Function App

Region * East US

Not finding your App Service Plan? Try a different region or select your App Service Environment.

Operating system

Linux is the only supported Operating System for your selection of runtime stack.

Operating System * [✓ Linux] Windows

Plan

The plan you choose dictates how your app scales, what features are enabled, and how it is priced. Learn more

Plan type * App service plan

Linux Plan (East US) * (New) ASP-FunctionsApp-9fe3

Create new

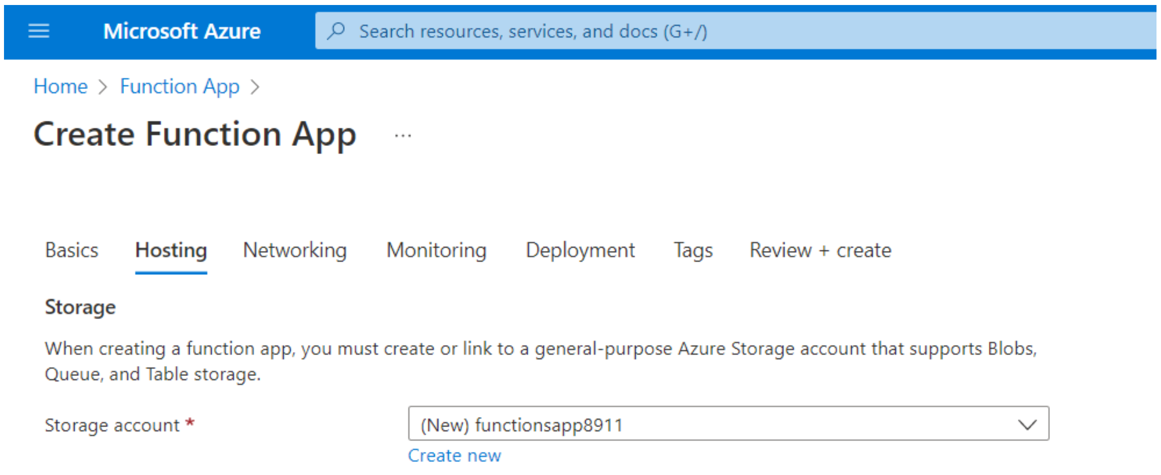
Pricing plan Basic B1 (100 total ACU, 1.75 GB memory, 1 vCPU)

Explore pricing plans

Zone redundancy

An App Service plan can be deployed as a zone redundant service in the regions that support it. This is a deployment

5. Click Next. In **Storage account**, select account.



Create Function App

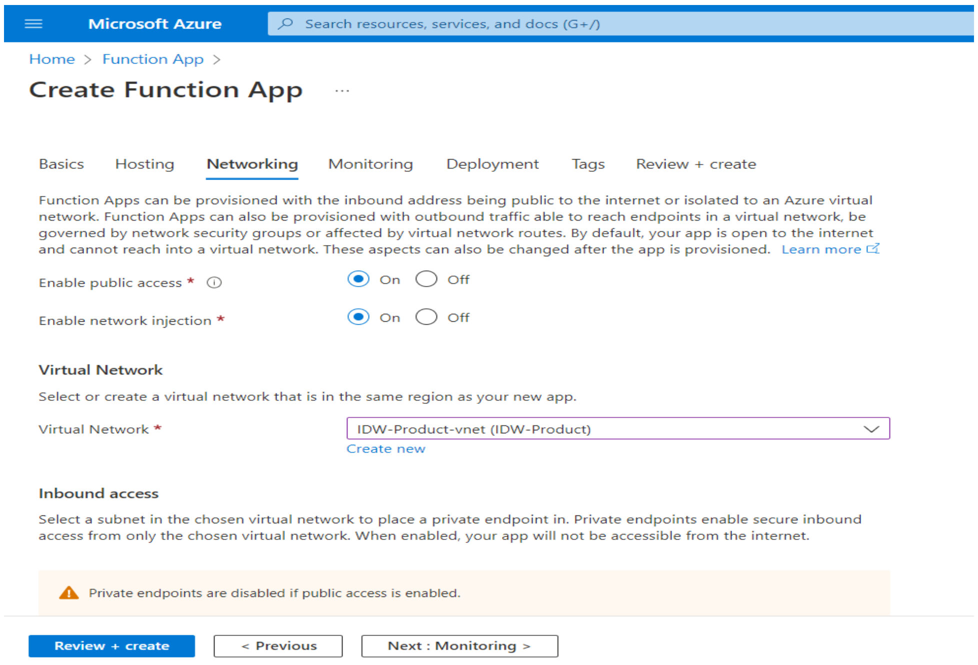
Storage

When creating a function app, you must create or link to a general-purpose Azure Storage account that supports Blobs, Queue, and Table storage.

Storage account * (New) functionsapp8911

Create new

6. In **Enable network injection**, select *On,* if LeapLogic is deployed in a closed network. And in **Virtual Network**, select the virtual network (it should be the same as the virtual machine instance's network). This is not required if LeapLogic is deployed in an open network.


Microsoft Azure
Search resources, services, and docs (G+/)
Home > Function App >

Create Function App

Basics Hosting Networking Monitoring Deployment Tags Review + create

Function Apps can be provisioned with the inbound address being public to the internet or isolated to an Azure virtual network. Function Apps can also be provisioned with outbound traffic able to reach endpoints in a virtual network, be governed by network security groups or affected by virtual network routes. By default, your app is open to the internet and cannot reach into a virtual network. These aspects can also be changed after the app is provisioned. Learn more

Enable public access * [✓ On] Off

Enable network injection * [✓ On] Off

Virtual Network

Select or create a virtual network that is in the same region as your new app.

Virtual Network * IDW-Product-vnet (IDW-Product)

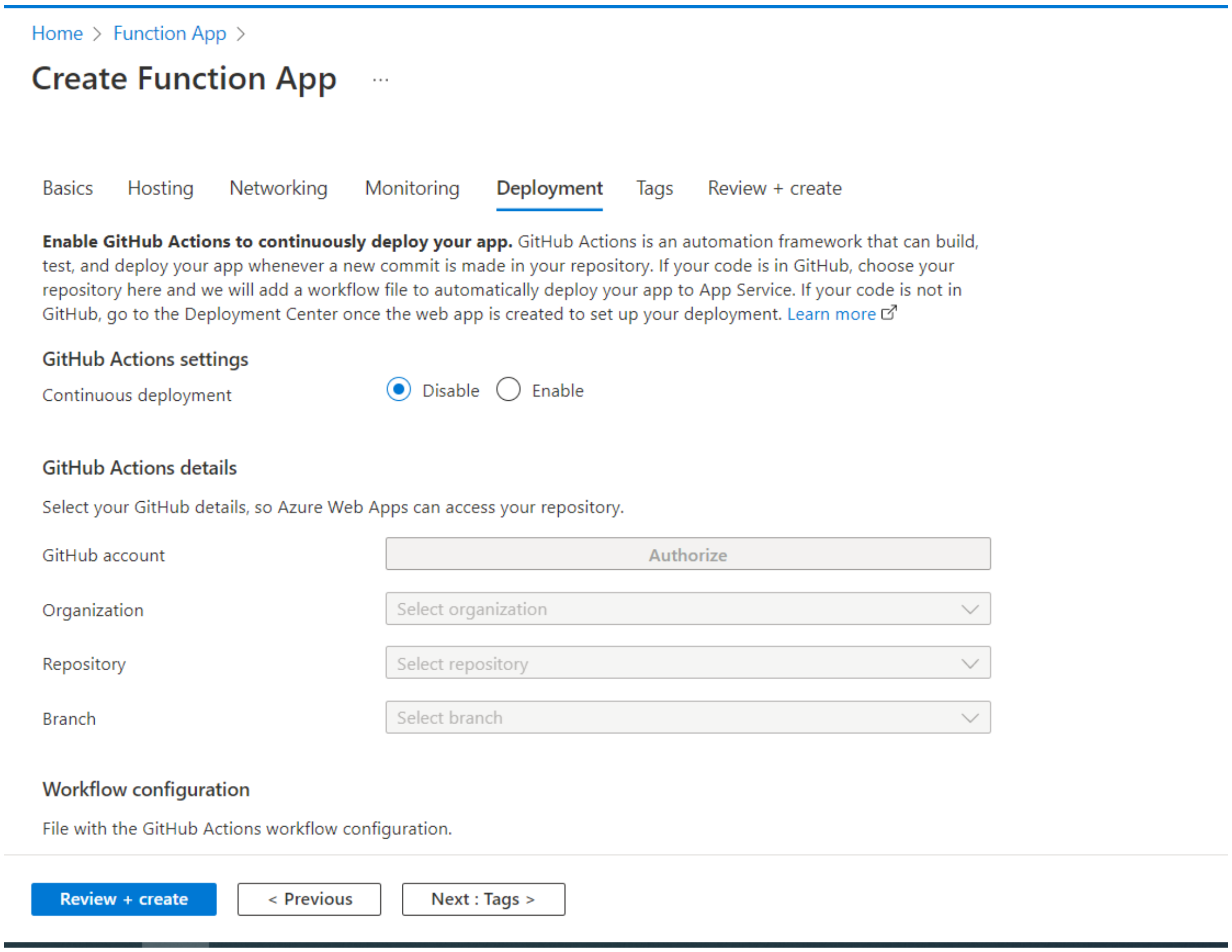
Create new

Inbound access

Select a subnet in the chosen virtual network to place a private endpoint in. Private endpoints enable secure inbound access from only the chosen virtual network. When enabled, your app will not be accessible from the internet.

Private endpoints are disabled if public access is enabled.

7. In **Monitoring** tab, click Next. No change is required in the Monitoring and Deployment tab.



Create Function App

Basics Hosting Networking Monitoring Deployment Tags Review + create

Enable GitHub Actions to continuously deploy your app. GitHub Actions is an automation framework that can build, test, and deploy your app whenever a new commit is made in your repository. If your code is in GitHub, choose your repository here and we will add a workflow file to automatically deploy your app to App Service. If your code is not in GitHub, go to the Deployment Center once the web app is created to set up your deployment. Learn more

GitHub Actions settings

Continuous deployment [✓ Disable] Enable

GitHub Actions details

Select your GitHub details, so Azure Web Apps can access your repository.

GitHub account Authorize

Organization Select organization

Repository Select repository

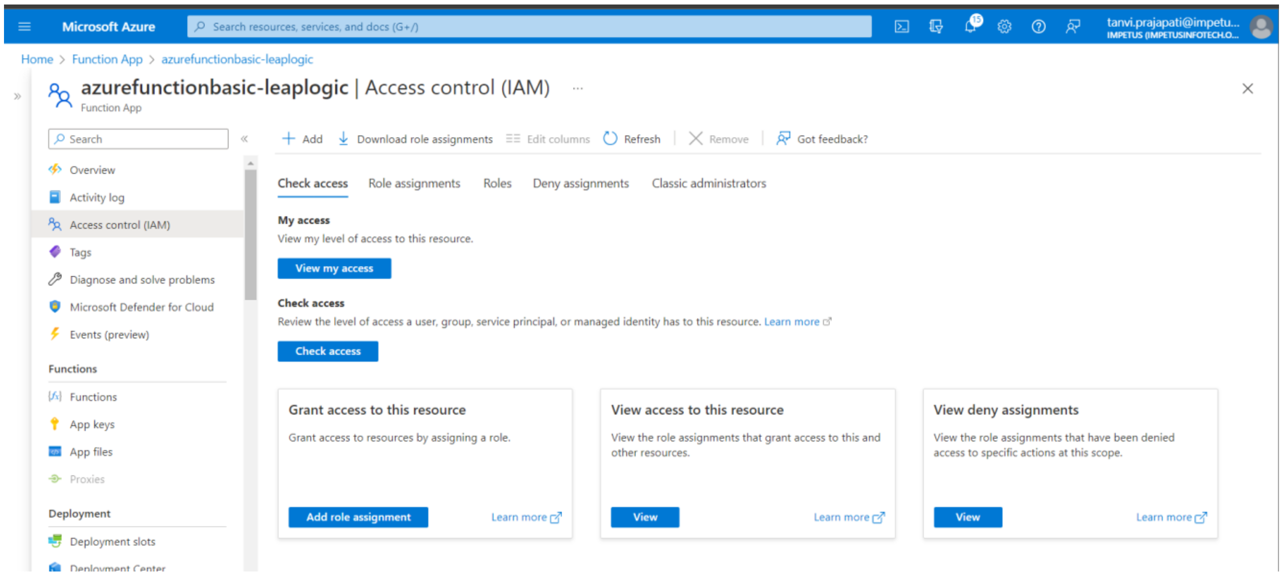
Branch Select branch

Workflow configuration

File with the GitHub Actions workflow configuration.

8. Click **Review + create** and then click **Create**. This will create the Function App.

9. Provide access to the Application from Access Control (IAM) (Application which is registered and whose client id and client secret id are used to create functions).



View my level of access to this resource.

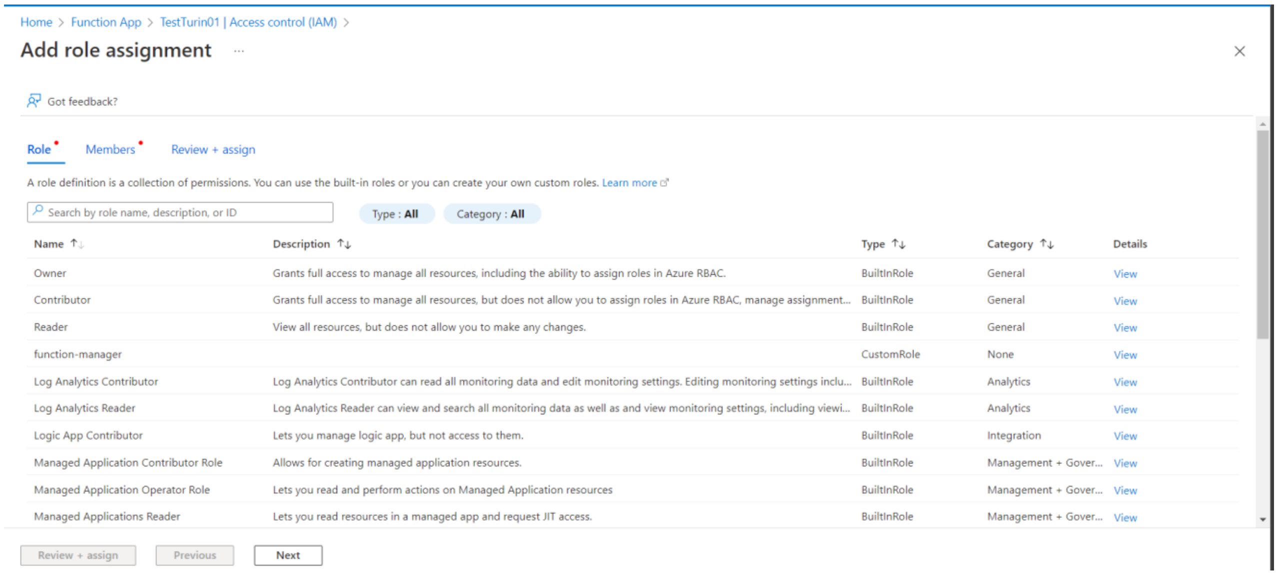
Review the level of access a user, group, service principal, or managed identity has to this resource. Learn more

Grant access to resources by assigning a role.

View the role assignments that grant access to this and other resources.

View the role assignments that have been denied access to specific actions at this scope.

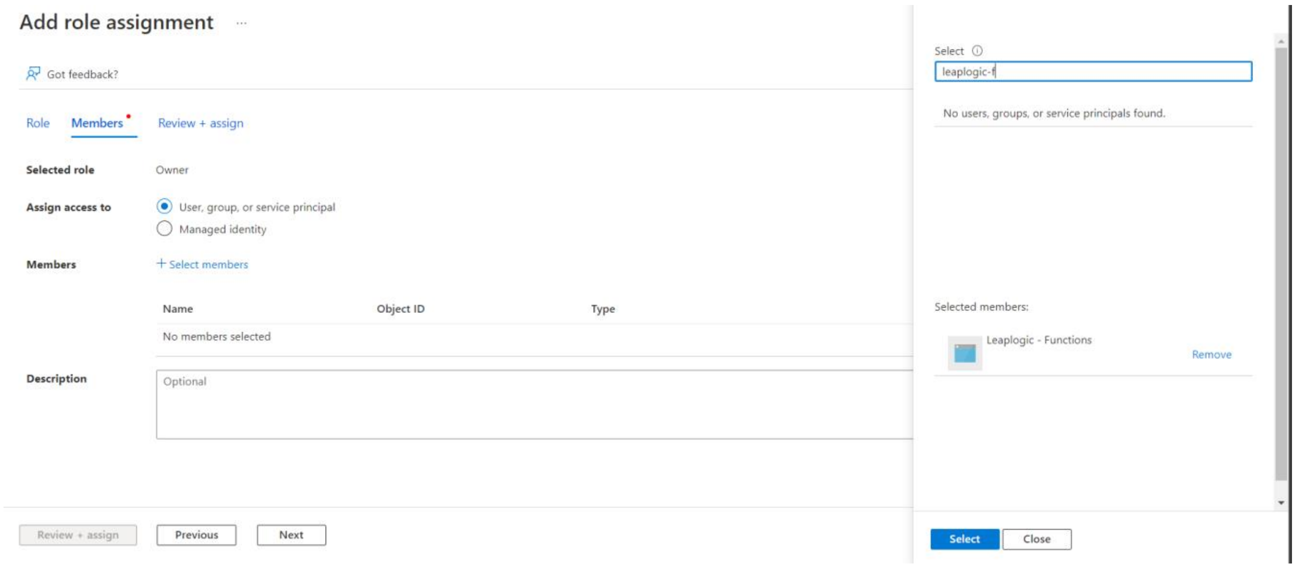
10. Click **Add role assignment**.
11. In the Role tab, select **Owner**.



A role definition is a collection of permissions. You can use the built-in roles or you can create your own custom roles. Learn more

Name	Description	Type	Category	Details
Owner	Grants full access to manage all resources, including the ability to assign roles in Azure RBAC.	BuiltInRole	General	View
Contributor	Grants full access to manage all resources, but does not allow you to assign roles in Azure RBAC, manage assignment...	BuiltInRole	General	View
Reader	View all resources, but does not allow you to make any changes.	BuiltInRole	General	View
function-manager		CustomRole	None	View
Log Analytics Contributor	Log Analytics Contributor can read all monitoring data and edit monitoring settings. Editing monitoring settings inclu...	BuiltInRole	Analytics	View
Log Analytics Reader	Log Analytics Reader can view and search all monitoring data as well as and view monitoring settings, including viewi...	BuiltInRole	Analytics	View
Logic App Contributor	Lets you manage logic app, but not access to them.	BuiltInRole	Integration	View
Managed Application Contributor Role	Allows for creating managed application resources.	BuiltInRole	Management + Gover...	View
Managed Application Operator Role	Lets you read and perform actions on Managed Application resources	BuiltInRole	Management + Gover...	View
Managed Applications Reader	Lets you read resources in a managed app and request JIT access.	BuiltInRole	Management + Gover...	View

12. Click **Select members** and add the Application registered to create functions.


Add role assignment

Got feedback?

Role | Members | Review + assign

Selected role: Owner

Assign access to: (● User, group, or service principal) (○ Managed identity)

Members: + Select members

Name	Object ID	Type
No members selected		

Description: Optional

Select: leaplogic-f

No users, groups, or service principals found.

Selected members:

Leaplogic - Functions — Remove

13. Click **Review + assign.**

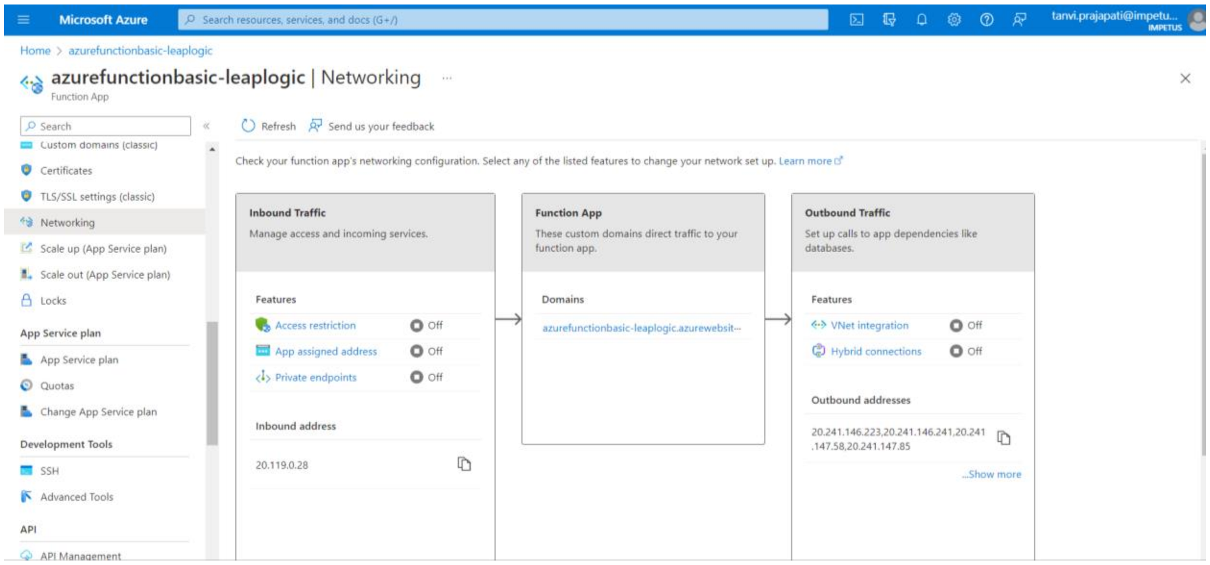
Note

1. If LeapLogic is deployed on an Azure Network, then it is available only in a closed network. The VNet Integration is required to allow functions to communicate with LeapLogic. If LeapLogic is deployed on an open network, then VNet Integration is not required.
2. To know how to register an application, refer to Register New Application.

2.4 Creating VNet Integration in Function App

To create a VNet Integration:

1. Open the Function App created using the above steps.
2. In left navigation pane, click **Networking**.



Home > azurefunctionbasic-leaplogic

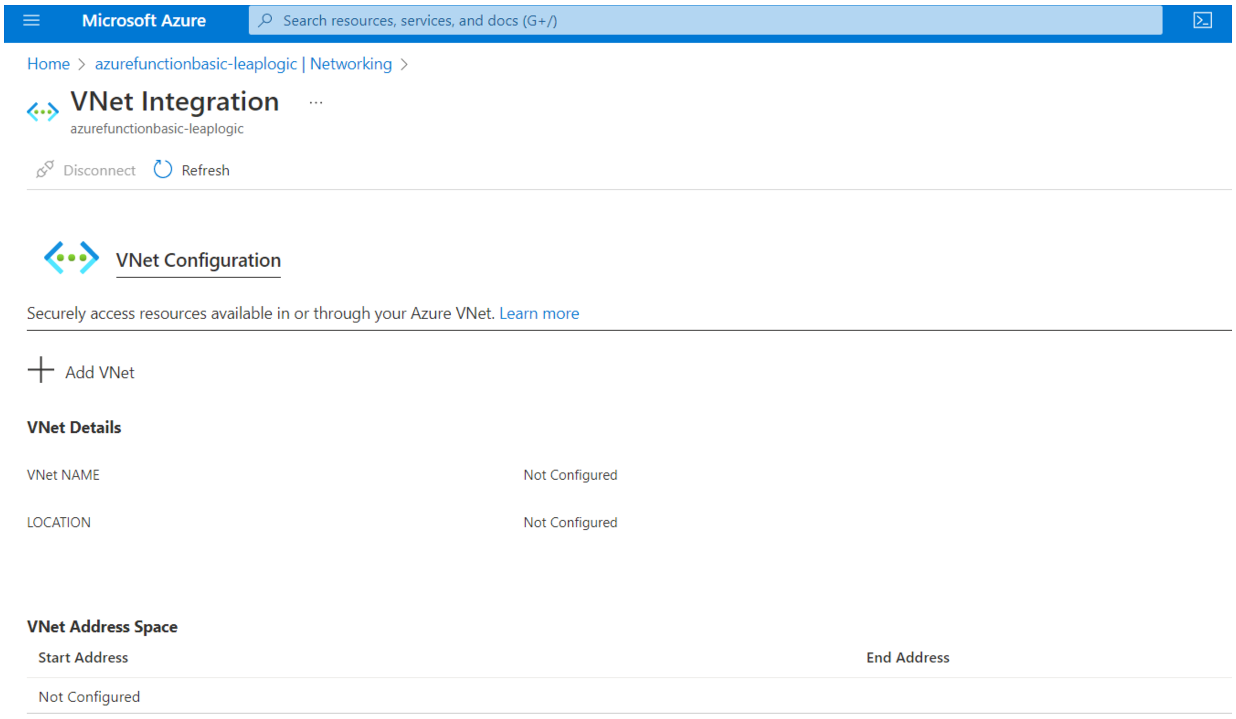
azurefunctionbasic-leaplogic | Networking

Function App

Check your function app's networking configuration. Select any of the listed features to change your network set up. Learn more

Inbound Traffic	Function App	Outbound Traffic
Manage access and incoming services.	These custom domains direct traffic to your function app.	Set up calls to app dependencies like databases.
Features: Access restriction Off; App assigned address Off; Private endpoints Off	Domains: azurefunctionbasic-leaplogic.azurewebsit...	Features: VNet integration Off; Hybrid connections Off
Inbound address: 20.119.0.28		Outbound addresses: 20.241.146.223,20.241.146.241,20.241.147.58,20.241.147.85

3. Click **VNet Integration.**



Home > azurefunctionbasic-leaplogic | Networking >

VNet Integration

azurefunctionbasic-leaplogic

VNet Configuration

Securely access resources available in or through your Azure VNet. Learn more

+ Add VNet

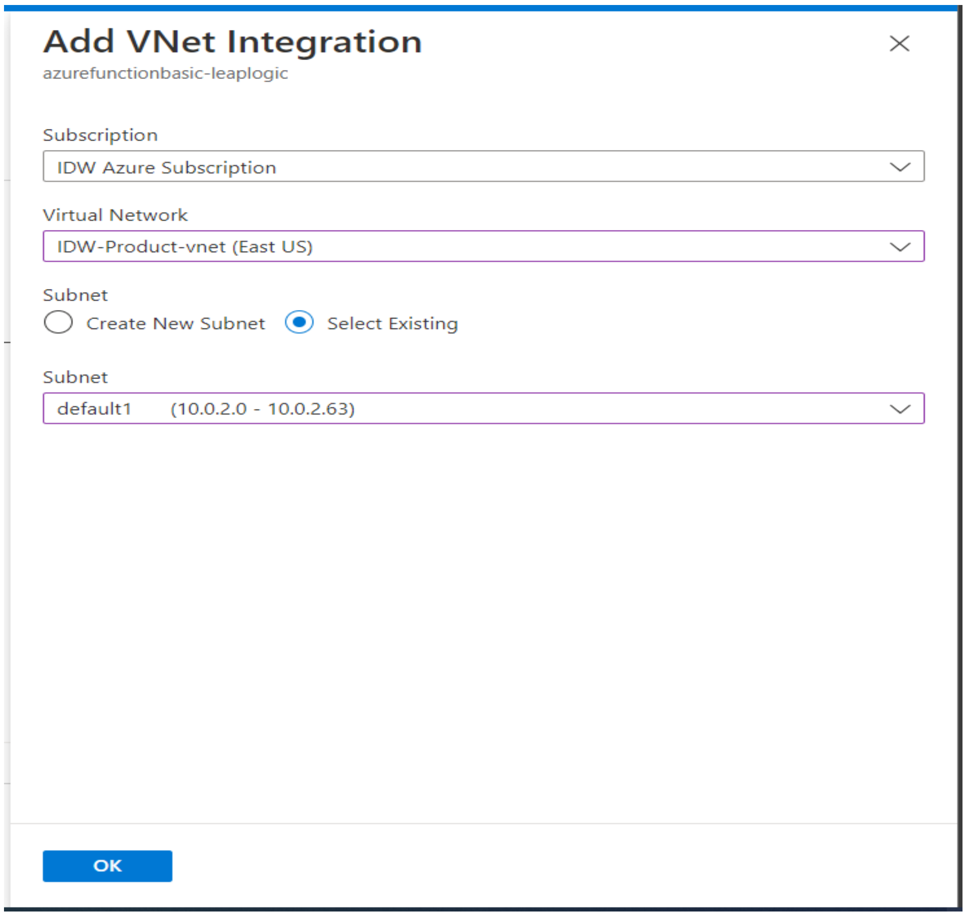
VNet Details

VNet NAME	Not Configured
LOCATION	Not Configured

VNet Address Space

Start Address	End Address
Not Configured	

4. Click **Add VNet.**
5. Select Virtual Network same as Virtual Machine's virtual network where LeapLogic is deployed.
6. Select **Create New Subnet** or **Select Existing** subnet and then click **OK**.


Add VNet Integration

azurefunctionbasic-leaplogic

Subscription: IDW Azure Subscription

Virtual Network: IDW-Product-vnet (East US)

Subnet: Create New Subnet [✓ Select Existing]

Subnet: default1 (10.0.2.0 - 10.0.2.63)

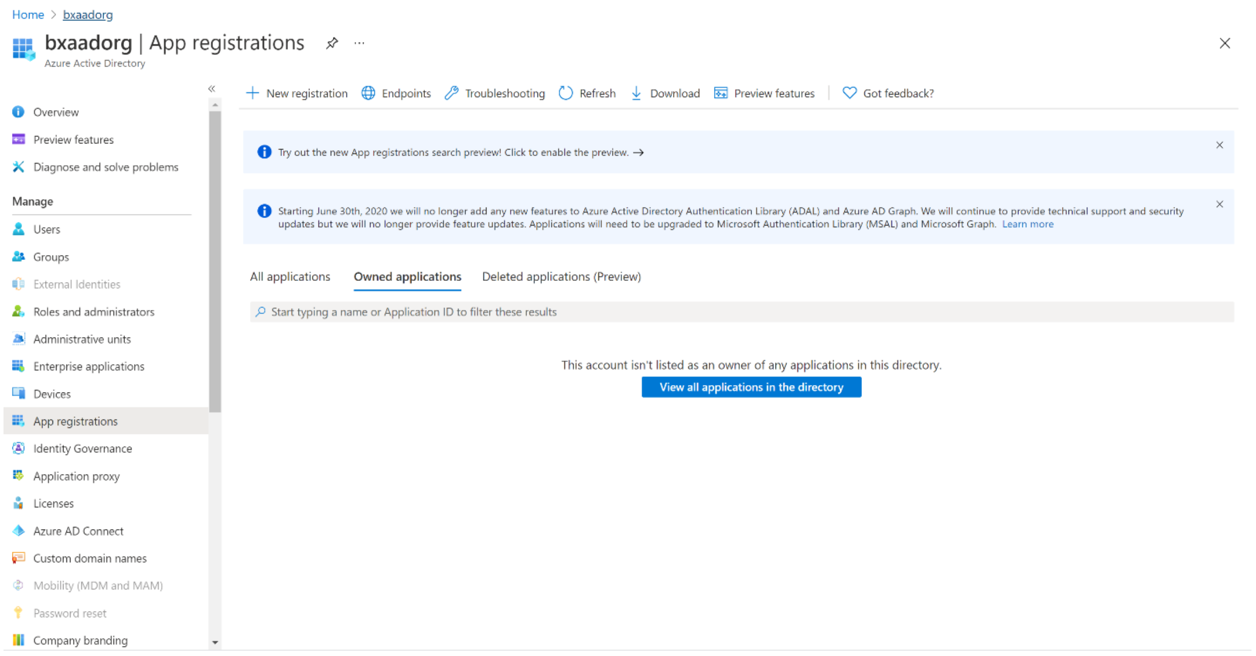
Note

When a VPC connector is used, Azure functions can communicate with LeapLogic through internal IP in a closed network. Ensure flag azure.vnet.private.ip.enabled is set to Y in turin-framework.properties. In case of an open network (https deployment), flag azure.vnet.private.ip.enabled should be set to N.

2.5 Register New Application

To register a new application:

1. In the Azure portal, select Azure Active Directory.
2. Select App registrations.
3. Select New registration.



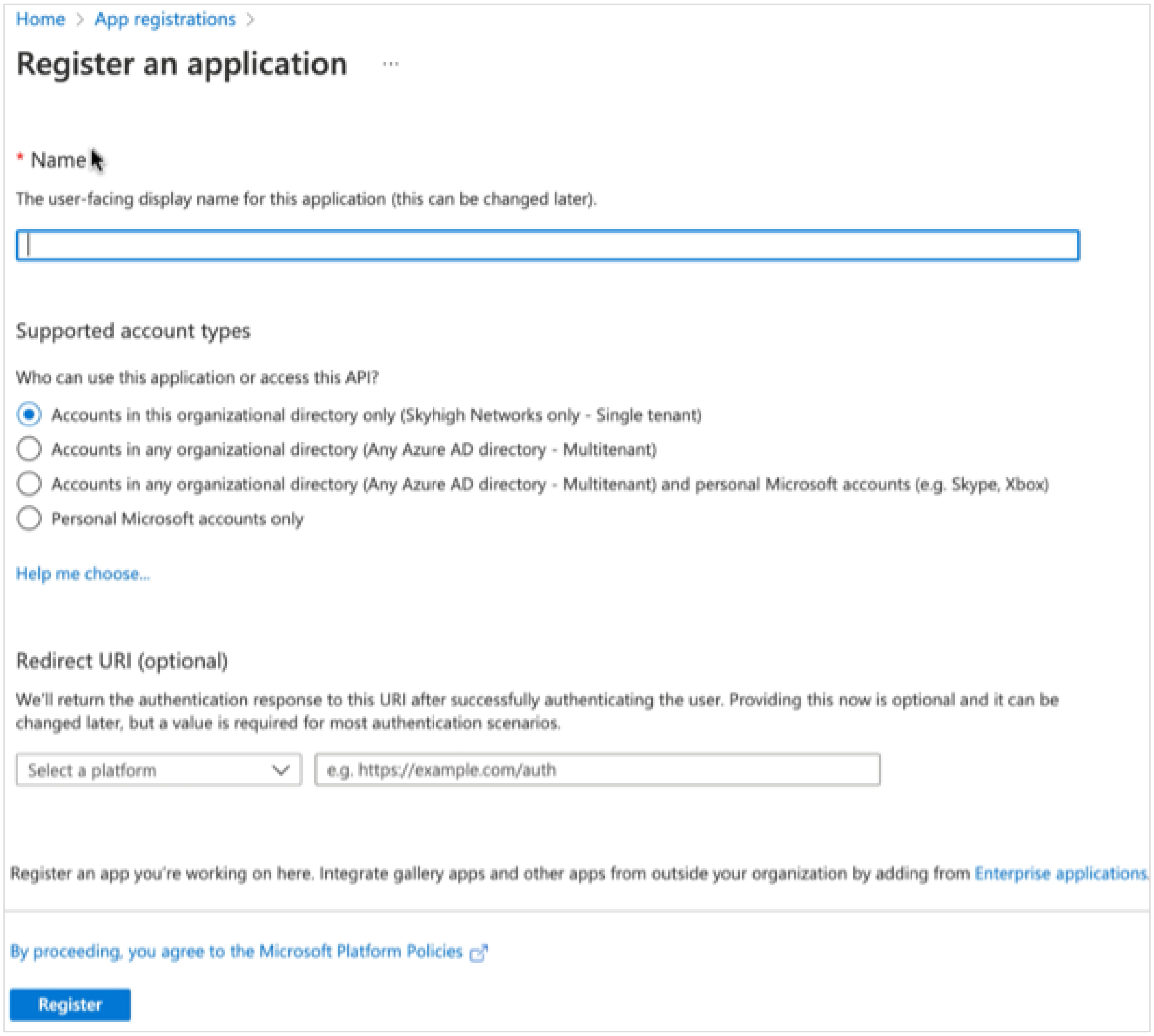
Try out the new App registrations search preview! Click to enable the preview. →

Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure AD Graph. We will continue to provide technical support and security updates but we will no longer provide feature updates. Applications will need to be upgraded to Microsoft Authentication Library (MSAL) and Microsoft Graph. Learn more

Start typing a name or Application ID to filter these results

This account isn't listed as an owner of any applications in this directory.

4. In **Supported account types**, select *Accounts in this organization directory only*.



Home > App registrations >

Register an application ...

* Name

The user-facing display name for this application (this can be changed later).

Supported account types

Who can use this application or access this API?

- [✓ Accounts in this organizational directory only (Skyhigh Networks only - Single tenant)]
- Accounts in any organizational directory (Any Azure AD directory - Multitenant)
- Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)
- Personal Microsoft accounts only

Help me choose...

Redirect URI (optional)

We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

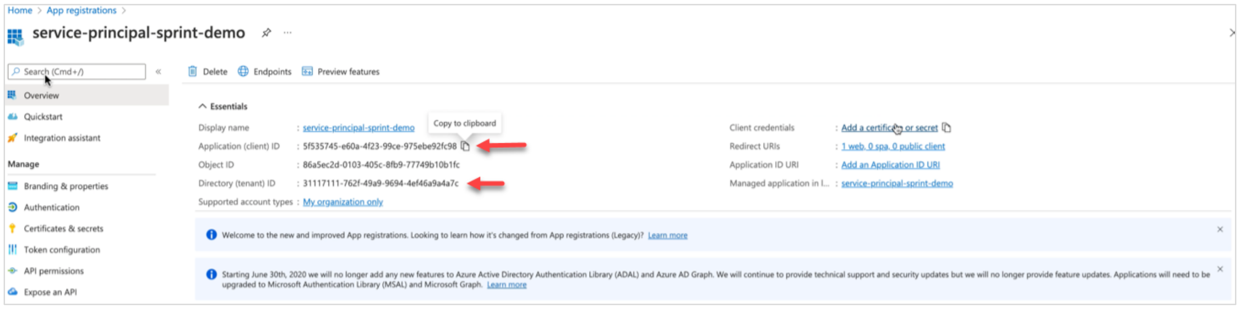
Select a platform | e.g. https://example.com/auth

Register an app you're working on here. Integrate gallery apps and other apps from outside your organization by adding from Enterprise applications.

By proceeding, you agree to the Microsoft Platform Policies

Register

5. After registering a new application, find the application (client) ID and Directory (tenant) ID from the overview menu option.



Home > App registrations >

service-principal-sprint-demo

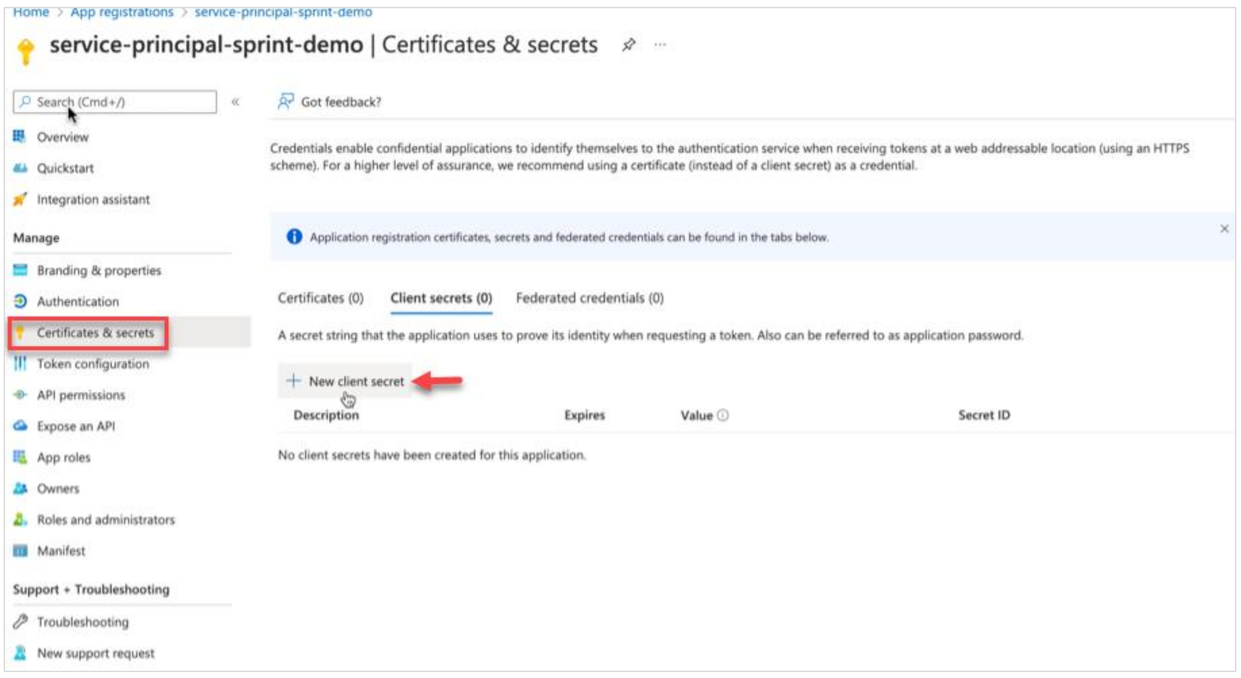
Essentials

Display name	: service-principal-sprint-demo
Application (client) ID	: 5f535745-e60a-4f23-99ce-975ebe92fc98
Object ID	: 86a5ec2d-0103-405c-8fb9-77749b10b1fc
Directory (tenant) ID	: 31117111-762f-49a9-9694-4ef46a9a4a7c
Supported account types	: My organization only
Client credentials	: Add a certificate or secret
Redirect URIs	: 1 web, 0 spa, 0 public client
Application ID URI	: Add an Application ID URI
Managed application in l...	: service-principal-sprint-demo

Welcome to the new and improved App registrations. Looking to learn how it's changed from App registrations (Legacy)? Learn more

Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure AD Graph. We will continue to provide technical support and security updates but we will no longer provide feature updates. Applications will need to be upgraded to Microsoft Authentication Library (MSAL) and Microsoft Graph. Learn more

6. Go to **Certificates & secrets** and in the **Client secrets** tab, click **New client secret**.


Home > App registrations > service-principal-sprint-demo

service-principal-sprint-demo | Certificates & secrets

Credentials enable confidential applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

Application registration certificates, secrets and federated credentials can be found in the tabs below.

Certificates (0) | Client secrets (0) | Federated credentials (0)

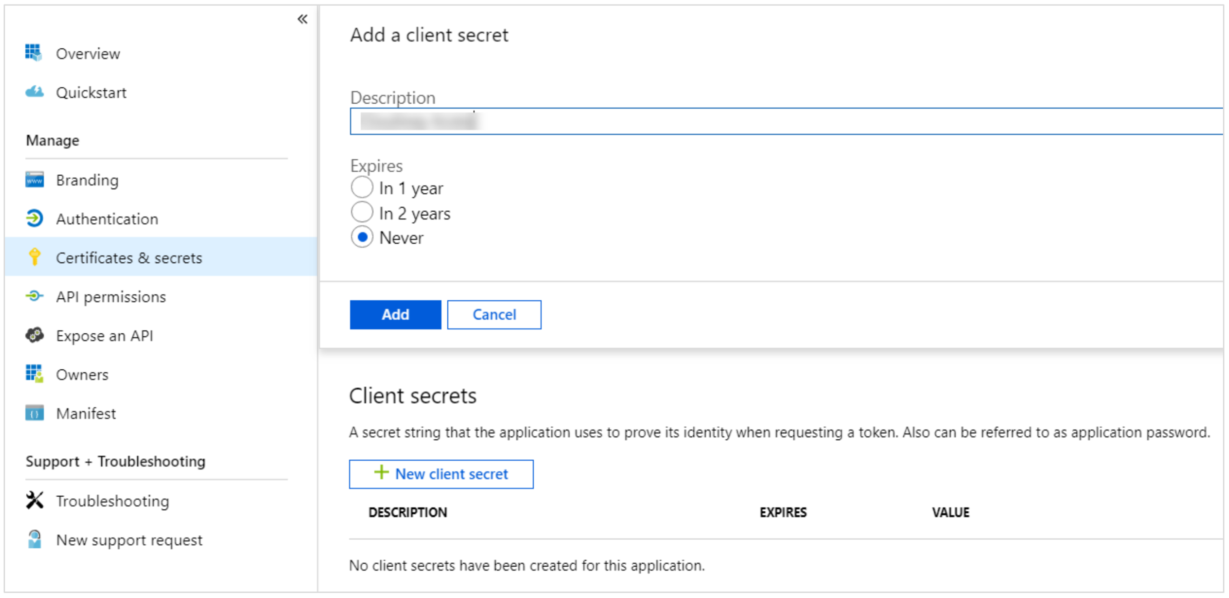
A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

+ New client secret

Description	Expires	Value	Secret ID

No client secrets have been created for this application.

7. Enter a description and expiration date. Click **Add**.


Add a client secret

Description

Expires
- In 1 year
- In 2 years
- [✓ Never]

Add Cancel

Client secrets

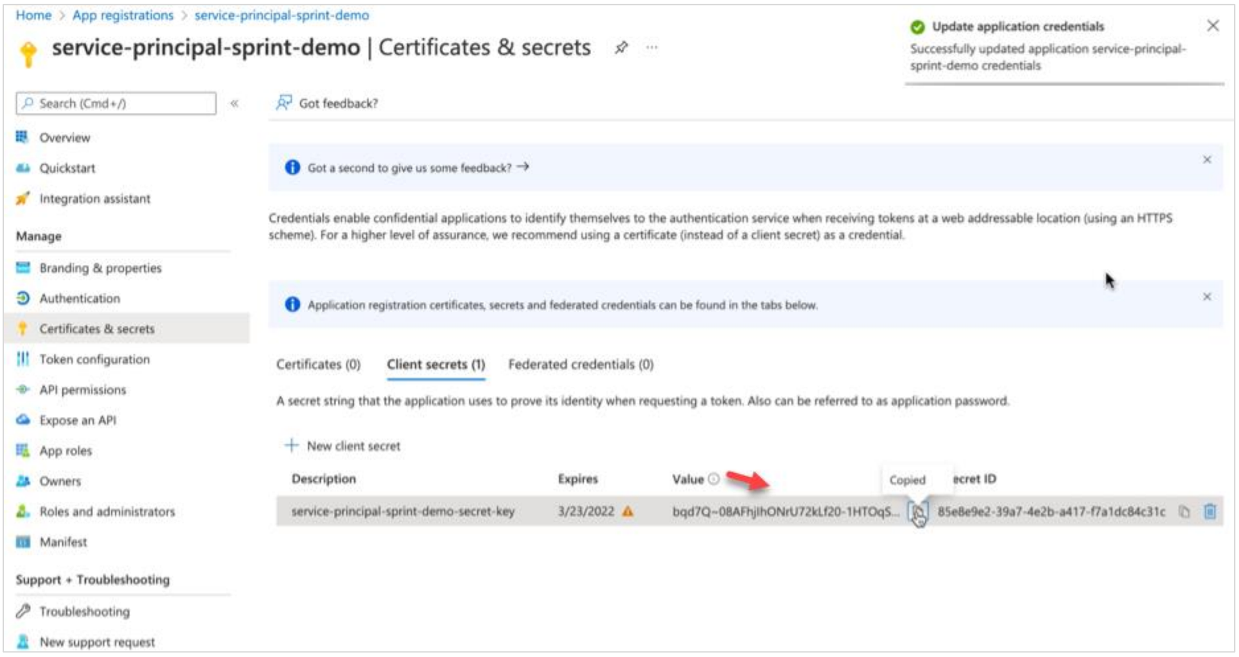
A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

+ New client secret

DESCRIPTION	EXPIRES	VALUE

No client secrets have been created for this application.

Client secrets are available under **Certificates & secrets**.



service-principal-sprint-demo | Certificates & secrets

Update application credentials

Successfully updated application service-principal-sprint-demo credentials

Got a second to give us some feedback? →

Credentials enable confidential applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

Application registration certificates, secrets and federated credentials can be found in the tabs below.

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

Description	Expires	Value	ecret ID
service-principal-sprint-demo-secret-key	3/23/2022	bqd7Q~08AFhjIhONrU72kLf20-1HTOqS...	85e8e9e2-39a7-4e2b-a417-f7a1dc84c31c

3. Getting Help

Contact LeapLogic technical support at info@leaplogic.io