## Guide to Storing and Deploying Flow Code in Azure Blob Storage with Prefect

### Overview

When deploying a Prefect flow, the execution environment requires access to the flow code. Prefect does **not** store flow code in Prefect Server or Prefect Cloud. Instead, you must use external storage.

### Flow code storage options:

* **Git-based storage**: GitHub, GitLab, Bitbucket
* **Docker image-based storage**
* **Cloud-provider storage**: AWS S3, Azure Blob Storage, GCP GCS

This guide focuses on using **Azure Blob Storage** as the flow code storage location.

### Reference Documentation:

* [Prefect: Store Flow Code](https://docs.prefect.io/v3/deploy/infrastructure-concepts/store-flow-code)
* [Cloud-Provider Storage: Azure Blob Storage](https://docs.prefect.io/v3/deploy/infrastructure-concepts/store-flow-code#cloud-provider-storage)

Sample code:  
A screenshot of a computer program

Description automatically generated

### Creating a Deployment for Azure Blob Storage

**Deployment Creation Options**

1. **Python Code**: Use the flow.deploy method.
   * Requires the flow.from\_source method to specify the flow storage location.
   * Key arguments:
     + source: URL to a Git repository or a storage object.
     + entrypoint: Path to the flow file and the function name (e.g., path/to/file.py:flow\_function).
2. **YAML Specification**: Define a deployment in a prefect.yaml file and deploy using the CLI.

### Steps to Set Up Azure Blob Storage

* 1. **Install Required Libraries**
* Install the [prefect-azure](https://prefecthq.github.io/prefect-azure/) library:   
  *pip install -U prefect-azure*
  1. **Register Azure Blocks**
* Register the Azure blocks in Prefect:   
  *prefect block register -m prefect\_azure*
  1. **Generate Azure Access Key**
* Create an access key for a role with **read** and **write** permissions in Azure.
* Navigate to [**Storage Account > Access Keys**](https://learn.microsoft.com/en-us/azure/storage/common/storage-account-keys-manage?tabs=azure-portal#tabpanel_1_azure-portal) in the Azure UI to generate a **connection string** containing all required information.
  1. **Create an Azure Blob Storage Credentials Block**

**Option 1**: Using the Prefect UI:

1. Go to the Prefect workspace: [Prefect Cloud](https://app.prefect.cloud/)
2. Navigate to **Configuration > Blocks**.
3. Click the **+** icon to create a new block.
4. Search for and select **Azure Blob Storage Credentials**, then click **Create**.
5. Enter a block name and paste the connection string into the **Connection String** field.  
   A screenshot of a computer

   Description automatically generated

**Option 2**: Using Code:

* Use the prefect-azure package to create and register the block programmatically.

### Flow Code Storage with Azure Blob Storage

* If the Azure Blob Storage location is **public** or the environment is already authenticated, you can reference the location directly without passing credentials explicitly.
* To explicitly pass credentials, use the **Azure Blob Storage Credentials block** created in the previous step.

### Updating Flow Code in Deployments

* If the flow code is stored in Git-based version control or cloud-based storage:
  + You can update the flow code **without rebuilding the deployment**, provided the flow entrypoint or server-required parameters remain unchanged.
* If changes impact server-required details (e.g., flow entrypoint), rebuild the deployment: **Python**: Rerun the Python script with the deploy method.

### Flow Code Storage for Deployments Using serve

The serve method in Python creates a deployment and starts a **local long-running process** to poll for flow runs simultaneously.

* Similar to deploy, but has fewer requirements for flow code storage:
  + If a **Docker image** is not specified, the flow code location must be defined using the from\_source method.
  + When using serve, from\_source is optional.