# Steps for setting up Prefect with Azure Blob Storage

1. **Create a Poetry Virtual Environment**

First, ensure you have Poetry installed. If not, you can install it using the following command:

*curl -sSL https://install.python-poetry.org | python3 -*

Then, create a new project and a virtual environment:

*poetry new my\_project*

*cd my\_project*

*poetry install*

1. **Install Prefect**

Add Prefect to your project using Poetry:

*poetry add prefect*

1. **Install prefect-azure**

Add the prefect-azure package: ([prefect-azure](https://prefecthq.github.io/prefect-azure/))

*poetry add prefect-azure*

1. **Register a Block**

Register the Azure blocks in Prefect:   
*prefect block register -m prefect\_azure*

1. **Create Azure Blob Storage Blob**

To create a blob, you need a connection string from Azure Blob Storage. Follow these steps:  
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. Go to your Azure portal.
2. Navigate to your Storage Account.
3. Under **Settings**, select **Access keys**.
4. Copy the connection string.
5. **Create an Azure Blob Storage Credentials Block**

You can create this block either in code or using the Prefect UI.

**Option 1:** how to do it in code:

*from prefect\_azure import AzureBlobCredentials*

*# Create and save Azure Blob Storage credentials block*

*AzureBlobCredentials(*

*connection\_string="your\_connection\_string\_here"*

*).save("my-credentials-block")*

**Option 2**: 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**.

Enter a block name and paste the connection string into the **Connection String** field.  
A screenshot of a computer

Description automatically generated

1. **Write Deployment Code and Use Block**

Here’s an example of how to write your deployment code and use the block:

A screenshot of a computer program

Description automatically generated

### **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.

When everything is setup run your script to deploy your flow.

*python example.py*

Running this script will create a deployment associated with the specified work pool and image.

1. **Run Deployment**

### *prefect deployment run my -deployment*

1. **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.

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)