This script is used to manage a local Apache Flink environment using Docker Compose. Here's a detailed breakdown of its functionality:

Script Overview

- The script allows you to **start** or **stop** a Flink environment locally using Docker Compose.
- It takes several arguments: `on` or `off`, AWS SSO profile, chip architecture (amd64 or arm64), Flink language (Python or Java), and optionally an S3 bucket name.
- The script uses **AWS SSO** credentials to configure the AWS environment, which are then passed to Docker containers.

Key Features Explained

1. **Command Argument Handling (`on` or `off`)**:
    - `on`: Starts the local environment.
    - `off`: Stops the local environment.
    - If the argument is incorrect, an error message is displayed with proper usage information.
2. **Argument Parsing**:
    - Parses optional arguments such as:
        - `--profile`: AWS SSO profile name.
        - `--chip`: Specifies the target architecture (amd64 or arm64).
        - `--flink-language`: Specifies the language to be used (Python or Java).
        - `--aws-s3-bucket`: Optionally specifies the AWS S3 bucket name.
3. **Validation Checks**:
    - Checks if required arguments (`--profile`, `--chip`, `--flink-language`) are provided.
    - If any of these required arguments are missing, it displays an appropriate error message and terminates.
4. **AWS SSO Login**:
    - Logs in using the specified AWS SSO profile.
    - Uses `aws2-wrap` to export AWS credentials (`AWS_ACCESS_KEY_ID`, `AWS_SECRET_ACCESS_KEY`, `AWS_SESSION_TOKEN`, `AWS_REGION`) to the environment.
    - Creates an `.env` file that contains the AWS environment variables required for the Docker Compose setup.
5. **Docker Compose Setup**:
    - Depending on the chip architecture, it runs either `linux-docker-compose.yml` or `mac-docker-compose.yml` to start the Flink containers.
    - The `.env` file is used to pass AWS credentials and settings to the Docker containers.
6. **Python Application Handling**:
    - If `flink_language` is set to `python`, it zips Python files in the `python_apps/kickstarter` directory within the Docker container. This likely helps to package the Python Flink jobs.

7. **Stopping the Environment**:
    - If the `off` argument is supplied, the script brings down the Docker Compose setup using the appropriate YAML file (`linux-docker-compose.yml` or `mac-docker-compose.yml`).

Usage Example

The script should be run with the following syntax:

```
./deploy-flink.sh <on | off> --profile=<AWS_SSO_PROFILE_NAME> --chip=
<amd64 | arm64> --flink-language=<python | java> [--aws-s3-bucket=
<AWS_S3_BUCKET_NAME>]
```

- **on**: Start the environment.
- **off**: Stop the environment.
- `--profile=<AWS_SSO_PROFILE_NAME>`: The AWS SSO profile to use.
- `--chip=<amd64 | arm64>`: Specify the chip architecture.
- `--flink-language=<python | java>`: Specify the language to use for Flink applications.
- `[--aws-s3-bucket=<AWS_S3_BUCKET_NAME>]`: Optionally specify an S3 bucket name.

Summary

- The script helps to manage a local Apache Flink environment via Docker, tailored for different architectures and Flink languages.
- It includes AWS integration through SSO and provides flexibility for running different Flink applications (Python or Java).
- The `.env` file generated during execution ensures that all necessary AWS credentials are passed securely to the Docker Compose setup.

This script is valuable for setting up and managing a local Flink environment that interacts with AWS resources and simplifies configuring different aspects like the execution architecture and language.