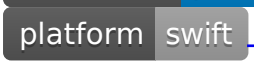
Scaffolded Swift Kitura server application

(https://bluemix.net)


(https://developer.ibm.com/swift/)

Table of Contents

Summary

This scaffolded application provides a starting point for creating Swift applications running on Kitura (http://www.kitura.io/).

Requirements

- Swift 4 (https://swift.org/download/)

Project contents

This application has been generated with the following capabilities and services:

- CloudEnvironment
- Embedded metrics dashboard
- Docker files
- Iterative Development
- IBM Cloud deployment

Embedded metrics dashboard

This application uses the SwiftMetrics package (https://github.com/RuntimeTools/SwiftMetrics) to gather application and system metrics.

These metrics can be viewed in an embedded dashboard on `/swiftmetrics-dash`. The dashboard displays various system and application metrics, including CPU, memory usage, HTTP response metrics and more.

Docker files

The application includes the following files for Docker support:

- `.dockerignore`
- `Dockerfile`
- `Dockerfile-tools`

The `.dockerignore` file contains the files/directories that should not be included in the built docker image. By default this file contains the `Dockerfile` and `Dockerfile-tools`. It can be modified as required.

The `Dockerfile` defines the specification of the default docker image for running the application. This image can be used to run the application.

The `Dockerfile-tools` is a docker specification file similar to the `Dockerfile`, except it includes the tools required for compiling the application. This image can be used to compile the application.

Details on how to build the docker images, compile and run the application within the docker image can be found in the Run section below.

IBM Cloud deployment

Your application has a set of cloud deployment configuration files defined to support deploying your application to IBM Cloud:

- `manifest.yml`
- `.bluemix/toolchain.yml`
- `.bluemix/pipeline.yml`

The `manifest.yml` (https://console.ng.bluemix.net/docs/manageapps/depapps.html#appmanifest) defines options which are passed to the Cloud Foundry `cf push` command during application deployment.

IBM Cloud DevOps (https://console.ng.bluemix.net/docs/services/ContinuousDelivery/index.html) service provides toolchains as a set of tool integrations that support development, deployment, and operations tasks inside IBM Cloud, for both Cloud Foundry and Kubernetes applications. The "Create Toolchain" button creates a DevOps toolchain and acts as a single-click deploy to IBM Cloud including provisioning all required services.

Configuration

Your application configuration information is stored in the `config.json` in the project root directory. This file is in the `.gitignore` to prevent sensitive information from being stored in git.

The connection information for any configured services, such as username, password and hostname, is stored in this file.

The application uses the CloudEnvironment package (https://github.com/IBM-Swift/CloudEnvironment) to read the connection and configuration information from the environment and this file.

If the application is running locally, it can connect to IBM Cloud services using unbound credentials read from this file. If you need to create unbound credentials you can do so from the IBM Cloud web console (example (https://console.ng.bluemix.net/docs/services/Cloudant/tutorials/create_service.html#creating-a-service-instance)), or using the CloudFoundry CLI `cf create-service-key` command (http://cli.cloudfoundry.org/en-US/cf/create-service-key.html).

When you push your application to IBM Cloud, these values are no longer used, instead the application automatically connects to bound services using environment variables.

Run

To build and run the application:

1. `swift build`
2. `.build/debug/generator-swiftserver-projects`

NOTE: On macOS you will need to add options to the `swift build` command: `swift build -Xlinker -lc++`

Docker

To build the two docker images, run the following commands from the root directory of the project:

- `docker build -t myapp-run .`
- `docker build -t myapp-build -f Dockerfile-tools .`
  You may customize the names of these images by specifying a different value after the `-t` option.

To compile the application using the tools docker image, run:

- `docker run -v $PWD:/swift-project -w /swift-project myapp-build /swift-utils/tools-utils.sh build release`

To run the application:

- `docker run -it -p 8080:8080 -v $PWD:/swift-project -w /swift-project myapp-run sh -c .build-ubuntu/release/generator-swiftserver-projects`

Iterative Development

The `iterative-dev.sh` script is included in the root of the generated Swift project and allows for fast & easy iterations for the developer. Instead of stopping the running Kitura server to see new code changes, while the script is running, it will automatically detect changes in the project's **.swift** files and recompile the app accordingly.

To use iterative development:

- For native OS, execute the `./iterative-dev.sh` script from the root of the project.
- With docker, shell into the tools container mentioned above, and run the `./swift-project/iterative-dev.sh` script. File system changes are detected using a low-tech infinitely looping poll mechanism, which works in both local OS/filesystem and across host OS->Docker container volume scenarios.

Deploy to IBM Cloud

You can deploy your application to Bluemix using:

- the CloudFoundry CLI
- an IBM Cloud toolchain

CloudFoundry CLI

You can deploy the application to IBM Cloud using the CloudFoundry command-line:

1. Install the Cloud Foundry command-line (https://docs.cloudfoundry.org/cf-cli/install-go-cli.html)
2. Ensure all configured services have been provisioned
3. Run `cf push` from the project root directory

The Cloud Foundry CLI will not provision the configured services for you, so you will need to do this manually using the IBM Cloud web console (example (https://console.ng.bluemix.net/docs/services/Cloudant/tutorials/create_service.html#creating-a-service-instance)) or the CloudFoundry CLI (`cf create-service` command) [http://cli.cloudfoundry.org/en-US/cf/create-service.html]. The service names and types will need to match your configuration.

IBM Cloud toolchain

You can also set up a default IBM Cloud Toolchain to handle deploying your application to IBM Cloud. This is achieved by publishing your application to a publicly accessible github repository and using the "Create Toolchain" button below. In this case configured services will be automatically provisioned, once, during toolchain creation.

(https://console.ng.bluemix.net/devops/setup/deploy/)

License

All generated content is available for use and modification under the permissive MIT License (see `LICENSE` file), with the exception of SwaggerUI which is licensed under an Apache-2.0 license (see `NOTICES.txt` file).

Generator

This project was generated with generator-swiftserver (https://github.com/IBM-Swift/generator-swiftserver) v5.2.0.