ALIBABA CLOUD

Alibaba Cloud

Function Compute

Migrate traditional frameworks

Document Version: 20201016

Alibaba Cloud

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.

2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company or individual in any form or by any means without the prior written consent of Alibaba Cloud.

3. The content of this document may be changed because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.

4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides this document based on the "status quo", "being defective", and "existing functions" of its products and services. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, special, or punitive damages, including lost profits arising from the use or trust in this document (even if Alibaba Cloud has been notified of the possibility of such a loss).

5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.

6. Please directly contact Alibaba Cloud for any errors of this document.

Document conventions

Style	Description	Example
Danger	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	**Danger:** Resetting will result in the loss of user configuration data.
Warning	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	**Warning:** Restarting will cause business interruption. About 10 minutes are required to restart an instance.
Notice	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	**Notice:** If the weight is set to 0, the server no longer receives new requests.
Note	A note indicates supplemental instructions, best practices, tips, and other content.	**Note:** You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click **Settings**> **Network**> **Set network type**.
Bold	Bold formatting is used for buttons , menus, page names, and other UI elements.	Click **OK**.
`Courier font`	Courier font is used for commands	Run the `cd /d C:/window` command to enter the Windows system folder.
Italic	Italic formatting is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	This format is used for an optional value, where only one item can be selected.	`ipconfig [-all|-t]`
{} or {a|b}	This format is used for a required value, where only one item can be selected.	`switch {active|stand}`

Table of Contents

1.Migrate Beego to Function Compute

This topic describes how to deploy a Beego application to Function Compute. Compared with traditional deployment methods, you can skip the steps such as purchasing a machine and deploy the traditional Beego application to the remote end for production directly with one click. In addition, you have the auto scaling, pay-as-you-go, and maintenance-free features offered.

Context

Beego is an HTTP framework that is used to quickly develop Go applications. It is a RESTful framework used to quickly develop various applications, such as APIs and Web applications and backend services. Beego is mainly inspired by Tornado, Sinatra, and Flask with some features of Go, such as Interface and Struct, combined.

Step 1: Prepare the environment

You do not need to install Docker but only Funcraft. The easiest way to install Funcraft is to download the executable binary files.

1. Install Funcraft on the local machine. For more information, see Install Funcraft.
2. Run the `fun --version` command to check whether the installation is successful.

Step 2: Deploy an application

1. Run the following command to install Beego. For more information, see official example.

```
go get github.com/astaxie/beego
```

2. Run the following command to create a `hello.go` project.

```
package main
import "github.com/astaxie/beego"
func main(){
   beego.Run()
}
```

3. Run the following command to run the local project:

```
go run hello.go
```

4. Run the `fun deploy -y` command to deploy the project to Function Compute.

```
$fun deploy -y
current folder is not a fun project.
Fun detected your application doesn't listen on '0.0.0.0:9000' in hello.go
Fun will replace your addr to '0.0.0.0:9000', and also backup your origin file hello.go to hello.go.bak
? Are your sure? Yes
Could not find any bin files from current folder.
Before using 'fun deploy', you must use 'GOARCH=amd64 GOOS=linux go build -ldflags "-s -w"' to c
ompile your project.
? Let Fun exec this command for you? Yes
Executing command 'GOARCH=amd64 GOOS=linux go build -ldflags "-s -w"'...
Tips:
You must use 'GOARCH=amd64 GOOS=linux go build -ldflags "-s -w"' to recompile your project ever
y time before using fun deploy.
Generating template.yml...
Generate Fun project successfully!
========= Fun will use 'fun deploy' to deploy your application to Function Compute! =========
...  .....  .....

            trigger httpTrigger deploy success
      function express deploy success
service express deploy success
Detect 'DomainName:Auto' of custom domain 'Domain'
Fun will reuse the temporary domain 15014775-XXX.test.functioncompute.com, expired at 2020-04-
03 09:52:55, limited by 1000 per day.
Waiting for custom domain Domain to be deployed...
custom domain Domain deploy success
```

Function Compute must start the service and listen to Port 0.0.0.0:9000. In the deployment log, Funcraft attempts to detect the start port of the application. If the ports do not match, press enter to modify the port, and then Funcraft automatically detects the executable program generated by the build. If no executable program is detected, you are prompted to compile with the specified command. Funcraft will compile the code after you press Enter. *bootstrap* file and *template.yml* file, and finally automatic deployment.

After the deployment is successful, you can see the temporary domain name generated by Function Compute in the log. Through this temporary domain name, you can directly access the just deployed application.

> **Note** The temporary domain name is used only for demonstration and development purposes. It is valid only for a limited time. To use an application in a production environment, you must bind a domain name that has been filed with Alibaba Cloud. For more information, see Bind a custom domain name.

2.Migrate Express to Function Compute

This topic describes how to deploy Express applications to Function Compute. Compared with traditional deployment methods, you can deploy traditional Express applications to remote sites for direct production with one click. In addition, you can also enjoy features such as auto scaling, pay-as-you-go, and maintenance-free.

Context

Express is a simple and flexible Web application development framework based on the Node.js platform. It provides a series of powerful characteristics to help you create a variety of Web and mobile device applications.

Step 1: Prepare the environment

You do not need to install Docker but only Funcraft. The easiest way to install Funcraft is to download the executable binary files.

1. Install Funcraft on the local machine. For more information, see Install Funcraft.
2. Run the `fun --version` command to check whether the installation is successful.

Step 2: Deploy an application

1. Use the following command to create an Express project.

   ```
   npx express-generator
   ```

   For more information, see Express application generator, or you can follow the steps described officially to create a simple Hello world example. If an Express project exists, this step is skipped.

2. Use the following command to enter the sample project you created or an existing project.

   ```
   cd <project-name>
   ```

3. Use the following command to install local dependencies.

   ```
   npm install
   ```

4. Run the command to run the project.
   - MacOS and Linux

     ```
     DEBUG=myapp:* npm start
     ```

   - Windows

     ```
     set DEBUG=myapp:* & npm start
     ```

5. Run the `fun deploy -y` command to deploy the project to Function Compute.

   Funcraft automatically begins to deploy the project.

```
$fun deploy -y
current folder is not a fun project.
Generating /Users/txd123/Desktop/Express/bootstrap...
Generating template.yml...
Generate Fun project successfully!
========= Fun will use 'fun deploy' to deploy your application to Function Compute! =========
using region: cn-qingdao
using accountId: **********3743
using accessKeyId: **********Ptgk
using timeout: 60

Collecting your services information, in order to caculate devlopment changes...

Resources Changes(Beta version! Only FC resources changes will be displayed):
... ... ... ...
            trigger httpTrigger deploy success
      function Express deploy success
service Express deploy success

Detect 'DomainName:Auto' of custom domain 'Domain'
Request a new temporary domain ...
The assigned temporary domain is 15795585-XXX.test.functioncompute.com,  expired at 2020-04-1
2 10:46:25, limited by 1000 per day.
Waiting for custom domain Domain to be deployed...
custom domain Domain deploy success
```

After the project is deployed, check the logs to find the temporary domain name that Function Compute generated. You can use this temporary domain name to access the deployed application.

> **Note** The temporary domain name is used only for demonstration and development purposes. It is valid only for a limited time. To use an application in a production environment, you must bind a domain name that has been filed with Alibaba Cloud. For more information, see Bind a custom domain name.

3.Migrate Gin Applications to Function Compute

This topic describes how to deploy a Gin application to Function Compute. Function Compute provides a method different from traditional deployment methods. You can deploy a traditional Gin application to Function Compute and use the application for production without performing steps such as purchasing a machine. Your Gin application can use the auto-scaling, pay-as-you-go, and O&M-free features of Function Compute.

Context

Gin is a microframework written in Go, which features elegant encapsulation, a user-friendly API, and clear source code annotations. Gin is fast, flexible, and fault-tolerant.

Step 1: Prepare the environment

You do not need to install Docker but only Funcraft. The easiest way to install Funcraft is to download the executable binary files.

1. Install Funcraft on the local machine. For more information, see Install Funcraft.
2. Run the `fun --version` command to check whether the installation is successful.

Step 2: Migrate the application

Example 1

1. Install Gin in an environment that has Go 1.11 or later installed. For more information, see the official sample.

```
go get -u github.com/gin-gonic/gin
```

2. When you create a project named example.go, write the following code into the project:

```
package main
import "github.com/gin-gonic/gin"
func main() {
r := gin.Default()
r.GET("/ping", func(c *gin.Context) {
   c.JSON(200, gin.H{
      "message": "pong",
   })
})
r.Run() // listen and serve on 0.0.0.0:8080 (for windows "localhost:8080")
}
```

3. Run the following command to run the local project:

```
go run example.go
```

4. Run the `fun deploy -y` command to deploy the project to Function Compute.

```
fun deploy -y
```

The following results are returned:

```
current folder is not a fun project.
Fun detected your application doesn't listen on '0.0.0.0:9000' in example.go
Fun will replace your addr to '0.0.0.0:9000', and also backup your origin file example.go to example.
go.bak
? Are your sure? Yes
Could not find any bin files from current folder.
Before using 'fun deploy', you must use 'GOARCH=amd64 GOOS=linux go build -ldflags "-s -w"' to c
omile your project.
? Let Fun exec this command for you? Yes
Executing command 'GOARCH=amd64 GOOS=linux go build -ldflags "-s -w"'...
Tips:
You must use 'GOARCH=amd64 GOOS=linux go build -ldflags "-s -w"' to recompile your project ever
y time before using fun deploy.
Generating template.yml...
Generate Fun project successfully!
========= Fun will use 'fun deploy' to deploy your application to Function Compute! =========
... ... ....
            trigger httpTrigger deploy success
      function express deploy success
service express deploy success
Detect 'DomainName:Auto' of custom domain 'Domain'
Request a new temporary domain ...
The assigned temporary domain is 15014775-1986***.test.functioncompute.com, expired at 2020-04
-03 09:52:55, limited by 1000 per day.
Waiting for custom domain Domain to be deployed...
custom domain Domain deploy success
```

Function Compute must start the service and listen to Port 0.0.0.0:9000. In the deployment log, Funcraft attempts to detect the start port of the application. If the ports do not match, press enter to modify the port, and then Funcraft automatically detects the executable program generated by the build. If no executable program is detected, you are prompted to compile with the specified command. Funcraft will compile the code after you press Enter. *bootstrap* file and *template.yml* file, and finally automatic deployment.

After the deployment is successful, you can see the temporary domain name generated by Function Compute in the log. Through this temporary domain name, you can directly access the just deployed application.

> **Note** The temporary domain name is used only for demonstration and development purposes. It is valid only for a limited time. To use an application in a production environment, you must bind a domain name that has been filed with Alibaba Cloud. For more information, see Bind a custom domain name.

Example 2

1. Run the following command to clone the sample project to your local machine. For more information, see the official sample.

```
git clone https://github.com/tanhe123/mdblog.git
```

2. Modify the configuration file.
   i. In the *config* directory, rename the *config.example.toml* file as *config.toml*.
   ii. Open the *config.toml* file and modify the following items:
   - Change `port=8091` to `port=9000`, so that your application starts on port 9000.
   - Change `debug = true` to `debug = false`, so that the production version is used.
   - Change `dir = "logs"` to `dir = "/tmp"` to write logs to the */tmp* directory. If Apsara File Storage NAS is not mounted, Function Compute has read and write permissions only for this directory.
3. Run the following command to compile the project:

```
go build
```

4. Run the following command to run the local project:

```
./mdblog
```

5. Run the `fun deploy -y` command to deploy the project to Function Compute.

```
fun deploy -y
```

After the project is deployed, check the logs to find the temporary domain name that Function Compute generated. You can use this temporary domain name to access the deployed application.

> **Note** The temporary domain name is used only for demonstration and development purposes. It is valid only for a limited time. To use an application in a production environment, you must bind a domain name that has been filed with Alibaba Cloud. For more information, see Bind a custom domain name.

4.Migrate Hacker News Applications to Function Compute

This topic describes how to deploy a Hacker News application to Function Compute. Function Compute provides a method different from traditional deployment methods. You can deploy a traditional Hacker News application to Function Compute and use the application for production without performing steps such as purchasing a machine. Your Hacker News application can use the auto-scaling, pay-as-you-go, and O&M-free features of Function Compute.

Step 1: Prepare the environment

You do not need to install Docker but only Funcraft. The easiest way to install Funcraft is to download the executable binary files.

1. Install Funcraft on the local machine. For more information, see Install Funcraft.
2. Run the `fun --version` command to check whether the installation is successful.

Step 2: Migrate the application

1. Run the following command to clone the Hacker News sample project to your local machine. If you already have a Hacker News project, skip this step.

   ```
   git clone https://github.com/nuxt/hackernews.git
   ```

2. Run the following command to open the sample project or an existing project:

   ```
   cd hackernews
   ```

3. Run the following command to install dependencies:

   ```
   npm install
   ```

4. Run the following command to run the local project:

   ```
   npm run dev
   ```

5. Run the following command to compile the Hacker News project:

   ```
   npm run build
   ```

6. Run the `fun deploy -y` command to deploy the project to Function Compute.

   ```
   fun deploy -y
   ```

   The following results are returned:

```
current folder is not a fun project.
Generating /Users/XXX/Desktop/hackernews/bootstrap...
Generating template.yml...
Generate Fun project successfully!
========= Fun will use 'fun deploy' to deploy your application to Function Compute! =========
... ... ...
Fun detected that your function hackernews/hackernews sizes exceed 50M. It is recommended th
at using the nas service to manage your function dependencies.
? Do you want to let fun to help you automate the configuration? Yes
? We recommend using the 'NasConfig: Auto' configuration to manage your function dependencies
.
Yes
... ... ...
starting upload /Users/XXX/Desktop/hackernews/node_modules to nas://hackernews/mnt/auto/
node_modules/
start fun nas init...
... ... ...
        trigger httpTrigger deploy success
   function hackernews deploy success
service hackernews deploy success
Detect 'DomainName:Auto' of custom domain 'Domain'
Request a new temporary domain ...
The assigned temporary domain is 14942717-XXX.test.functioncompute.com, expired at 2020-04-02
13:51:57, limited by 1000 per day.
Waiting for custom domain Domain to be deployed...
custom domain Domain deploy success
```

The size of a code package uploaded to Function Compute cannot exceed 50 MB. If Funcraft detects that your code package exceeds this limit, it starts the large dependency wizard. In the wizard, enter *Y* and Funcraft will automatically upload the third-party dependencies in your function to Apsara File Storage NAS.

After the project is deployed, check the logs to find the temporary domain name that Function Compute generated. You can use this temporary domain name to access the deployed application.

> **Note** The temporary domain name is used only for demonstration and development purposes. It is valid only for a limited time. To use an application in a production environment, you must bind a domain name that has been filed with Alibaba Cloud. For more information, see Bind a custom domain name.

5.Migrate Next.js Applications to Function Compute

This topic describes how to deploy a Next.js application to Function Compute. Function Compute provides a method different from traditional deployment methods. You can deploy a traditional Next.js application to Function Compute and use the application for production without performing steps such as purchasing a machine. Your Next.js application can use the auto-scaling, pay-as-you-go, and O&M-free features of Function Compute.

Context

Next.js is a React framework with server-side rendering. The framework is integrated with webpack, Babel, and Express. You can build server-side rendered (SSR) React applications by using only Next.js, React, and ReactDOM, and do not need to be concerned about routing. Next.js is highly integrated, making it easy to implement code splitting, redirection, hot updates, server rendering, and frontend rendering.

Step 1: Prepare the environment

You do not need to install Docker but only Funcraft. The easiest way to install Funcraft is to download the executable binary files.

1. Install Funcraft on the local machine. For more information, see Install Funcraft.
2. Run the `fun --version` command to check whether the installation is successful.

Step 2: Migrate the application

1. Run the following command to create a Next.js project. If you already have a Next.js project, skip this step.

   ```
   npm init next-app
   ```

2. Run the following command to open the project:

   ```
   cd nextjs
   ```

3. Run the following command to run the local project:

   ```
   npm run dev
   ```

   The following results are returned:

   

4. Run the following command to compile the Next.js project:

   ```
   npm run build
   ```

5. Run the `fun deploy -y` command to deploy the project to Function Compute.

   Funcraft automatically begins to deploy the project.

   During deployment, perform the following operations as prompted:

i. When Funcraft detects that the project is not created by it, Funcraft prompts you that it can create a project for you. Press the Enter key or enter *Y*.

ii. After Funcraft creates a project, it prompts you to deploy the project. Press the Enter key or enter *Y*.

Funcraft deploys the application to Function Compute.

After the project is deployed, check the logs to find the temporary domain name that Function Compute generated. You can use this temporary domain name to access the deployed application.

> **Note** The temporary domain name is used only for demonstration and development purposes. It is valid only for a limited time. To use an application in a production environment, you must bind a domain name that has been filed with Alibaba Cloud. For more information, see Bind a custom domain name.

6.Migrate Nuxt.js Applications to Function Compute

This topic describes how to deploy a Nuxt.js application to Function Compute. Function Compute provides a method different from traditional deployment methods. You can deploy a traditional Nuxt.js application to Function Compute and use the application for production without performing steps such as purchasing a machine. Your Nuxt.js application can use the auto-scaling, pay-as-you-go, and O&M-free features of Function Compute.

Context

Nuxt.js is a general-purpose application framework based on Vue.js. It abstracts away the client-server architecture and focuses on user interface rendering for applications. Nuxt.js has preset all the configurations needed to develop server-side rendered (SSR) Vue.js applications. It provides many useful features for the client-server application architecture, such as asynchronous data loading, middleware support, and layout support.

Step 1: Prepare the environment

You do not need to install Docker but only Funcraft. The easiest way to install Funcraft is to download the executable binary files.

1. Install Funcraft on the local machine. For more information, see Install Funcraft.
2. Run the `fun --version` command to check whether the installation is successful.

Step 2: Migrate the application

1. Run the following command to create a Nuxt.js project. If you already have a Nuxt.js project, skip this step.

   ```
   npx create-nuxt-app <project-name>
   ```

2. Run the following command to open the sample project or an existing project:

   ```
   cd <project-name>
   ```

3. Run the following command to install dependencies:

   ```
   yarn install
   ```

4. Run the following command to run the local project:

   ```
   yarn dev
   ```

   The following results are returned:

5. Run the following command to compile the Nuxt.js project:

   ```
   yarn build
   ```

6. Run the `fun deploy -y` command to deploy the project to Function Compute.

   Funcraft automatically begins to deploy the project.

   > **Note** If the .yml file cannot be added, update Funcraft to the latest version.

   After the project is deployed, check the log to find the temporary domain name that Function Compute generated. You can use this temporary domain name to access the deployed application.

   > **Note** The temporary domain name is used only for demonstration and development purposes. It is valid only for a limited time. To use an application in a production environment, you must bind a domain name that has been filed with Alibaba Cloud. For more information, see Bind a custom domain name.

7.Migrate ThinkPHP applications to Function Compute

This topic describes how to deploy a ThinkPHP application to Function Compute. Compared with traditional deployment methods, you can skip the steps such as purchasing a machine and deploy the traditional ThinkPHP application remotely and directly for production with one click. The ThinkPHP application also has the auto scaling, pay-as-you-go and maintenance-free features.

Context

ThinkPHP is a free open source, fast, simple, object-oriented lightweight PHP development framework. It was born for Agile Web application development and simplify enterprise application development. ThinkPHP has always been designed in a simple and practical way, with excellent performance and simple code, and ease of use at the same time. ThinkPHP-based applications that are released under the Apache2 open source license agreement can be used for free. ThinkPHP-based applications can even be released as an open source project or be sold as commercial products.

Step 1: Prepare the environment

You do not need to install Docker but only Funcraft. The easiest way to install Funcraft is to download the executable binary files.

1. Install Funcraft on the local machine. For more information, see Install Funcraft.
2. Run the `fun --version` command to check whether the installation is successful.

Step 2: Deploy an application

1. Use the following command to create a ThinkPHP project. If you have an existing ThinkPHP project, skip this step.

   ```
   composer create-project topthink/think tp
   ```

2. Use the following command to enter the newly created project or an existing project.

   ```
   cd <project-name>
   ```

3. Run the following command to run the local project:

   ```
   php think run
   ```

4. Run the `fun deploy -y` command to deploy the project to Function Compute.

   Funcraft automatically begins to deploy the project.

```
fun deploy -y
current folder is not a fun project.
downloading nginx and php7.2 zip from https://gosspublic.alicdn.com/fun/frameworks/support/fun-support-custom-php-d73a6bd6.zip to /private/var/folders/wl/_2ngtj291wx1cj55xlnn290w0000gn/T/fun-support-custom-php-d73a6bd6.zip...
extract nginx and php7.2 zip to custom runtime...
Generating
... ... ...
      trigger httpTrigger deploy success
   function tp deploy success
service tp deploy success
Detect 'DomainName:Auto' of custom domain 'Domain'
Request a new temporary domain ...
The assigned temporary domain is 15631862-XXX.test.functioncompute.com,  expired at 2020-04-10 13:17:42, limited by 1000 per day.
Waiting for custom domain Domain to be deployed...
custom domain Domain deploy success
```

After the project is deployed, check the logs to find the temporary domain name that Function Compute generated. You can use this temporary domain name to access the deployed application.

> **Note** The temporary domain name is used only for demonstration and development purposes. It is valid only for a limited time. To use an application in a production environment, you must bind a domain name that has been filed with Alibaba Cloud. For more information, see Bind a custom domain name.

8.Migrating Spring Boot to Function Compute

This topic describes how to deploy a Spring Boot application to Function Compute. Compared with traditional deployment methods, you can deploy a traditional Spring Boot application to the remote end directly for production with one click. Features such as auto scaling, pay-as-you-go, and maintenance-free are provided.

Context

Spring Boot was developed by the Pivotal team in 2013 and its first version of the new open-source lightweight framework was released in April 2014. A Spring application is designed based on Spring 4.0. In addition to inheriting excellent features of the Spring Framework, it also simplifies the configuration to further simplify the entire build and development process of Spring applications. In addition, Spring Boot integrates a large number of frameworks to solve problems such as dependency package version conflicts and reference instability.

Step 1: Prepare the environment

You do not need to install Docker but only Funcraft. The easiest way to install Funcraft is to download the executable binary files.

1. Install Funcraft on the local machine. For more information, see Install Funcraft.
2. Run the `fun --version` command to check whether the installation is successful.

Step 2: Deploy an application

1. Create a Spring Boot project. For more information, see Spring Quickstart Guide.
2. Use the following command to enter the sample project you created or an existing project.

   ```
   cd <project-name>
   ```

3. Run the following command to run the local project:
   - Runs a project on the MacOS or Linux platform.

     ```
     ./mvnw spring-boot:run
     ```

   - Run the project on Windows.

     ```
     mvnw spring-boot:run
     ```

4. Under the root directory of the project, run the **mvn package** command for packaging. The compiled output is similar to the following example.

```
$mvn package
[INFO] Scanning for projects...
[INFO]
[INFO] ----------------------< com.example:Spring-Boot >-----------------------
[INFO] Building Spring-Boot 0.0.1-SNAPSHOT
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- maven-resources-plugin:3.1.0:resources (default-resources) @ Spring-Boot ---
... ... ...
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- maven-jar-plugin:3.1.2:jar (default-jar) @ Spring-Boot ---
[INFO] Building jar: /Users/txd123/Desktop/Spring-Boot/target/Spring-Boot-0.0.1-SNAPSHOT.jar
[INFO]
[INFO] --- spring-boot-maven-plugin:2.2.6.RELEASE:repackage (repackage) @ Spring-Boot ---
[INFO] Replacing main artifact with repackaged archive
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  38.850 s
[INFO] Finished at: 2020-03-31T15:09:34+08:00
[INFO] ------------------------------------------------------------------------
```

5. Run the `fun deploy -y` command to deploy the project to Function Compute.

   Funcraft automatically begins to deploy the project.

```
$fun deploy -y
current folder is not a fun project.
Generating template.yml...
Generate Fun project successfully!
========= Fun will use 'fun deploy' to deploy your application to Function Compute! =========
using region: cn-qingdao
using accountId: ***********3743
using accessKeyId: ***********Ptgk
using timeout: 60

Collecting your services information, in order to caculate devlopment changes...

Resources Changes(Beta version! Only FC resources changes will be displayed):

            trigger httpTrigger deploy success
      function Spring-Boot deploy success
service Spring-Boot deploy success

Detect 'DomainName:Auto' of custom domain 'Domain'
Request a new temporary domain ...
The assigned temporary domain is 15639196-XXX.test.functioncompute.com,  expired at 2020-04-1
0 15:19:56, limited by 1000 per day.
Waiting for custom domain Domain to be deployed...
custom domain Domain deploy success
```

After the project is deployed, check the logs to find the temporary domain name that Function Compute generated. You can use this temporary domain name to access the deployed application.

> **Note** The temporary domain name is used only for demonstration and development purposes. It is valid only for a limited time. To use an application in a production environment, you must bind a domain name that has been filed with Alibaba Cloud. For more information, see Bind a custom domain name.

9.Migrate Egg.js to Function Compute

This topic describes how to deploy Egg.js applications in Function Compute. Compared with the traditional deployment method, you can skip the steps such as purchasing the machine and deploy an Egg.js application to the remote end for production. Features such as auto scaling, pay-as-you-go, and maintenance-free are provided.

Context

Egg.js is designed for enterprise-level frameworks and applications. It provides developers with core features of Web development and a plug-in mechanism with high scalability. Developers follow a unified agreement to develop applications, reducing development and maintenance costs. For more information about Egg, see Egg.js.

Step 1: Prepare the environment

You do not need to install Docker but only Funcraft. The easiest way to install Funcraft is to download the executable binary files.

1. Install Funcraft on the local machine. For more information, see Install Funcraft.
2. Run the `fun --version` command to check whether the installation is successful.

Step 2: Deploy an application

Example 1: Deploy a HelloWorld application.

1. Run the following command to create an Egg.js Project. For more information, see quick initialization. If an Egg.js project exists, skip this step.

```
mkdir egg-example
cd egg-example
npm init egg --type=simple
npm i
```

2. Run the following command to run the local project:
3. Run the `fun deploy -y` command to deploy the project to Function Compute.

   Funcraft automatically begins to deploy the project.

```
$fun deploy -y
using template: template.yml
using region: cn-qingdao
using accountId: ***********3743
using accessKeyId: ***********Ptgk
using timeout: 60

Collecting your services information, in order to caculate devlopment changes...

Resources Changes(Beta version! Only FC resources changes will be displayed):
... ... ...
    trigger httpTrigger deploy success
  function egg-example deploy success
service egg-example deploy success

Detect 'DomainName:Auto' of custom domain 'Domain'
Fun will reuse the temporary domain 17090425-19861144305****.test.functioncompute.com, expired
at 2020-04-27 10:27:05, limited by 1000 per day.

Waiting for custom domain Domain to be deployed...
custom domain Domain deploy success
```

After the project is deployed, check the logs to find the temporary domain name that Function Compute generated. You can use this temporary domain name to access the deployed application.

> **Note** The temporary domain name is used only for demonstration and development purposes. It is valid only for a limited time. To use an application in a production environment, you must bind a domain name that has been filed with Alibaba Cloud. For more information, see Bind a custom domain name.

Example 2: The procedure of deploying an open-source Egg.js Web application is described as an example.

For more information about the preview effect of the sample application, see preview.

1. Run the following command to clone the sample to your local disk:

```
git clone https://github.com/OrangeXC/mtime
```

2. Run the following commands to enter *mtime* directory and install dependencies.

```
cd mtime
npm install
```

3. Open the *config/config. default.js* local file, and set the MySQL username and password to

the correct values. When starting an application locally, you need to use the database configuration in the *config/config.default .js*file.

> **Note** By default, Egg.js takes precedence of the *config/config.prod.js* database configuration when starting an application in the production environment. Therefore, you need to configure the database information of the production environment to the file. After the configuration, locally run the `npm run start` and `npm run stop` commands to start or stop the application in production mode to verify whether the configuration is correct.

4. Run the following command to run the local project:

```
npm run dev
```

5. Run the following commands to modify the *config/config.prod.js* file to get the permission to read the Egg.js cache and log directory.

```
config.rundir = '/tmp/run',
config.logger = {
  dir: '/tmp/log',
}
```

6. Run the `fun deploy -y` command to deploy the project to Function Compute.

```
$ fun deploy -y
using template: template.yml
using region: cn-qingdao
using accountId: ***********3743
using accessKeyId: ***********Ptgk
using timeout: 60

Collecting your services information, in order to caculate devlopment changes...

Resources Changes(Beta version! Only FC resources changes will be displayed):
... ... ...
      trigger httpTrigger deploy success
   function egg-example deploy success
service egg-example deploy success

Detect 'DomainName:Auto' of custom domain 'Domain'
Fun will reuse the temporary domain 17090425-19861144305****.test.functioncompute.com, expired at 2020-04-27 10:27:05, limited by 1000 per day.

Waiting for custom domain Domain to be deployed...
custom domain Domain deploy success
```

After the project is deployed, check the logs to find the temporary domain name that Function Compute generated. You can use this temporary domain name to access the deployed application.

> **Note** The temporary domain name is used only for demonstration and development purposes. It is valid only for a limited time. To use an application in a production environment, you must bind a domain name that has been filed with Alibaba Cloud. For more information, see Bind a custom domain name.