SIEMENS

MindSphere

Deployment and update of Cloud Foundry Applications

Compact User Manual

07/2020
V1801.July/2020.1

Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

⚠ DANGER

indicates that death or severe personal injury **will** result if proper precautions are not taken.

⚠ WARNING

indicates that death or severe personal injury **may** result if proper precautions are not taken.

⚠ CAUTION

indicates that minor personal injury can result if proper precautions are not taken.

NOTICE

indicates that property damage can result if proper precautions are not taken.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

⚠ WARNING

Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Siemens AG
Division Digital Factory
Postfach 48 48
90026 NÜRNBERG
GERMANY

V1801.July/2020.1
Ⓟ 07/2020 Subject to change

Table of contents

This short documentation will provide you with some additional information on what are best-practices to make your application available for productive usage on the operator tenant.

Deployment of a Cloud Foundry Application 1

1.1 Develop and test applications on Developer Plan tenant

Checkpoints

- Check, if the application name is in one single manifest file.
- Note that a component has to be registered for each of the application names in the manifest file. The component has to match the respective name in the manifest file.
- Check if the backing services used by your application are defined:
  - In one single manifest file and
  - As components in the "Developer Cockpit".
- Check if your node.js application
  - is built as release incl. all dependencies
  - the version in the package.json are fix and not variable.
    If the application is configured with variable npm versions, it could happen that there is a difference between the developed and the deployed application. This is considered a change in the code base, which will lead to the rejection of the application within CF/ MindSphere.

Prepare for handover to an Operator

- The manifest should not contain HOST, DOMAIN or ROUTE entries. This are deprecated.
- Configure the build pack in the manifest file. Auto detection works not in every case
- Before you pack you Zip-File, check if the PATH entry in the manifest is similar to the Zip-File folder structure.
- The folder in the Zip-file should only contain application necessary files.
- We recommend that following Zip-File structure.
  - /
    /component1/application file.jar
    /component2/application files
    /component2/css/css-files
    /component2/js/java-script files
  - Manifest file should not part of the zip. MindSphere will add it later.
- Be aware that the operator will deploy the application with the same name. Thus, we recommend using random route in the manifest
- DNS-entries or hostnames are global unique in cloud foundry. Two applications cannot have the same host name from scratch in different spaces. Your application should not bind to hard coded application names. They will be different if two or more operators using the application.

- Make a test push with the manifest you will deliver. Mistakes or errors will result that operators cannot deploy the application.
- Every change in the application must go through developer cockpit.
- Default memory settings are 256MB for every application. If your application needs more, define it in the manifest.
- Test your application in developer CF space with MindSphere Gateway registration. Even if it worked on another CF environment.
- Define the exact endpoints required for the components. If you have more than one component, /** might have negative effects. Thus, we recommend using folders under these endpoints, e.g. /html/** or /carcontroll/*
- Backing services can be shared through cf spaces. For more information, refer to Sharing Service Instances. (https://developer.mindsphere.io/paas/paas-cloudfoundry-backingservices.html#sharing-service-instances)

  Note

  MindSphere DevOps plan does not supports sharing of backing services.
- It is not possible to deploy two different applications in one space. Every application uploaded to MindSphere for operator use needs one space.
- An email notification will be sent to the operator as soon as the application is available for productive use.

Handover to an Operator

Assigning an application to an operator is a three way process.

1. Assign the application from the Developer tenant to the operator in the "Developer Cockpit".
2. Operator confirms the handover in the "Operator Cockpit".
3. Developer grants the handover of the application in the "Developer Cockpit".

1.2 Deploy and operate applications on Operator plan tenant

Deployment information

- Two minor versions of the same application cannot be registered at the same time in MindSphere Gateway.
- It is the recommended to use same appname and Space name.
- You cannot update the application, with a version that has not been uploaded from the Developer Cockpit.
- For detection, once the application is in "Deployed" state in Operator Cockpit, it is not possible to change the cf application name.
- Create backing service before pushing the application.

- Be aware, that deleting and redeploying the application might result in a different DNS-name by using random route.
- Download the binaries the whole zip in an empty folder. Do not add new files to it. CF push uploads all files in the given path. That will be a problem with HTML applications, if you add or changes files i.e. pictures.
- Extract also the folder structure of the Zip. This might be part of the CF path variable.
  - "Appbinary.zip" includes the components of your CF application, which you target later using "cf push".
  - Ensure that AppBinary is extracted as a separate folder containing only the service components and does not include the icon or metadata file.
- Operator needs to access the application in Operator Cockpit to check the deployment status.
- Every application needs an own space.
- After adding the roles to a user, this user needs to logoff and logon.
- Prepare your business model as soon as possible before you publish it to the store.

Receive an application from a developer

Receiving an application from a developer is a three step process:

1. The Developer assigns the application to an Operator in the "Developer Cockpit".
2. The Operator confirms the receipt in the application catalog in the "Operator Cockpit".
3. The Developer grants the handover of the application in the "Developer Cockpit".

Grant access of customers using IoT Value plan to your application

1. Navigate to the application details in your "Operator Cockpit" and select "Provisioning" tab.
2. Click "Add to customer".
3. Enter the tenant name and an e-mail address of your customer.
   Note, that provisioning the application can take up to 5 minutes.

1.3 Using application on IoT Value plan tenant

After an application has been provisioned to the IoT Value plan customer, the application will appear on the launchpad after executing the following steps:

1. Tenant admin assigns the new application roles to the users, who will consume the application.
2. Users have to log off to make the application available in their Launchpad.

Update an existing Cloud Foundry Application 2

2.1 Update usecase

In this chapter the following scenario is described:

- You have an existing application deployed in your Operator tenant.
- You want to deploy a new version of your application.
- You want to use the backing services of the older version of your application.

Update of an application is possible in two ways:

- Update application with downtime.
- Update application without downtime.

Update application with downtime

Update an application with downtime does not allows the customers to access the application during upgrade.

To upgrade an application with downtime, proceed with the following steps:

1. Accept the application from a developer in the Operator Cockpit.
2. Inform your customers that they shortly will be disconnected from the application.
3. De-Provision your customers from your application using the "Provisioning" wizard in the Operator Cockpit.
4. De-Register the existing version of the application you want to update.
5. Deploy and register the new version of the application.
6. Provision the new version of your application using "Provisioning" wizard in the Operator Cockpit.

Update application without downtime

To update an application without any downtime, refer to Upgrade a CF application (without downtime) (https://documentation.mindsphere.io/resources/html/operator-cockpit/en-US/119273250827.html).