SIEMENS

MindSphere

部署和更新 Cloud Foundry 应用

精简版用户手册

法律资讯

警告提示系统

为了您的人身安全以及避免财产损失，必须注意本手册中的提示。人身安全的提示用一个警告三角表示，仅与财产损失有关的提示不带警告三角。警告提示根据危险等级由高到低如下表示。

⚠危险
表示如果不采取相应的小心措施，**将会**导致死亡或者严重的人身伤害。

⚠警告
表示如果不采取相应的小心措施，**可能**导致死亡或者严重的人身伤害。

⚠小心
表示如果不采取相应的小心措施，可能导致轻微的人身伤害。

注意
表示如果不采取相应的小心措施，可能导致财产损失。

当出现多个危险等级的情况下，每次总是使用最高等级的警告提示。如果在某个警告提示中带有警告可能导致人身伤害的警告三角，则可能在该警告提示中另外还附带有可能导致财产损失的警告。

合格的专业人员

本文件所属的产品/系统只允许由符合各项工作要求的**合格人员**进行操作。其操作必须遵照各自附带的文件说明，特别是其中的安全及警告提示。 由于具备相关培训及经验，合格人员可以察觉本产品/系统的风险，并避免可能的危险。

按规定使用 Siemens 产品

请注意下列说明：

⚠警告
Siemens 产品只允许用于目录和相关技术文件中规定的使用情况。如果要使用其他公司的产品和组件，必须得到 Siemens 推荐和允许。正确的运输、储存、组装、装配、安装、调试、操作和维护是产品安全、正常运行的前提。必须保证允许的环境条件。必须注意相关文件中的提示。

商标

所有带有标记符号 ® 的都是 Siemens AG 的注册商标。本印刷品中的其他符号可能是一些其他商标。若第三方出于自身目的使用这些商标，将侵害其所有者的权利。

责任免除

我们已对印刷品中所述内容与硬件和软件的一致性作过检查。然而不排除存在偏差的可能性，因此我们不保证印刷品中所述内容与硬件和软件完全一致。印刷品中的数据都按规定经过检测，必要的修正值包含在下一版本中。

Siemens AG
Division Digital Factory
Postfach 48 48
90026 NÜRNBERG
德国

目录

本文档的内容

1

本文档十分简短，将提供一些有关最佳实践的额外信息，以便将您的操作员租户应用用于生产用途。

Cloud Foundry 应用的部署 2

2.1 在 Developer Plan 租户上开发和测试应用

检查点

- 检查应用名是否在单个 manifest 文件中
- 注意，必须为 manifest 文件中的每个应用名注册一个组件。组件必须与 manifest 文件中的相应名称匹配。
- 检查是否已定义应用所使用的后端服务：
  - 在单个 manifest 文件中并且
  - 作为“Developer Cockpit”中的组件。
- 检查 node.js 应用是否
  - 构建为包含所有相关项的版本
  - package.json 中的 version 是固定的而非可变的。
    如果应用配置了可变的 npm versions，则开发的应用和部署的应用之间可能会存在差异。这可视为代码库变更，将导致 CF/MindSphere 拒绝该应用。

准备移交给操作员

- manifest 不应包含 HOST、DOMAIN 或 ROUTE 条目。这些条目已弃用。
- 在 manifest 文件中配置构建包。自动检测并非在任何情况下均有效
- 打包 Zip 文件之前，请检查 manifest 中的 PATH 条目是否与 Zip 文件夹结构类似。
- Zip 文件中的文件夹应仅包含应用所需文件。无 .git 等文件夹。
- 建议采用以下 Zip 文件结构。
  - /
    /component1/application file.jar
    /component2/application files
    /component2/css/css-files
    /component2/js/java-script files
  - Manifest 文件不应是 zip 的一部分。MindSphere 稍后会添加它。

- 请注意，操作员将用相同的名称部署应用。因此，我们建议在 manifest 中使用随机 route
- DNS 条目或主机名在 Cloud Foundry 中是全局唯一的。在不同的空间中，两个应用的 host 名称不能有任何相同之处。应用不应与硬编码的应用名称绑定。如果两个或多个操作员使用该应用，应用名称将会有所差异。
- 对将要交付的 manifest 进行测试推送。失误或错误将导致操作员无法部署应用。此外，Siemens 正在对 manifest 进行测试。如果它无法正常工作，应用将被拒绝。
- 您无法在操作员 cf org 中直接更新应用。应用中的每个更改都必须经过 Developer Cockpit。否则，应用将被拒绝并从系统中自动删除。
- 每个应用的默认内存设置为 256MB。如果您的应用需要更多内存，请在 manifest 中进行定义。
- 注册 Mindspace 网关后，在开发者 CF 空间中测试应用。即使该应用可在另一个 CF 环境中正常工作，也要进行测试。
- 尽可能精确地定义组件端点。如果您有多个组件，/** 可能会产生负面影响。使用 /** 向网关发出信号，说明每次调用（GET、POST 等）都应路由到该端点。无法保证此端点为最后一个被调用的端点。在某些情况下，所有请求将只路由到此端点。因此，建议使用这些端点下的文件夹，例如 /html/** 或 /carcontrol/*
- 后端服务无法通过 CF 空间共享。在 Developer Cockpit 中进行应用设计时应该考虑到这一点。如果两个应用需要相同的服务，那么它们应该属于同一个 MindSphere 应用。
- 无法在一个空间中部署两个不同的应用。每个上传到 Mindsphere 供操作员使用的应用都需要空间。
- 所有需要由操作员更改的环境 variables 都应进行记录。
- 应用的“上传”使用验证最多需要 14 天。

移交给操作员

将应用分配给操作员是一个三向过程。

1. 在“Developer Cockpit”中，将开发者租户的应用分配给操作员。
2. 操作员在“Operator Cockpit”中确认移交。
3. 开发者在“Developer Cockpit”中授权移交应用。

2.2 在操作员计划租户上部署和运行应用

部署信息

- 同一应用的两个次要版本不能同时在 Mindsphere 网关中注册。
- 空间名称必须是 <应用名 - 版本>。否则，系统无法检测该应用。
- 无法使用尚未从 Developer Cockpit 上载的版本更新应用。
- 对于检测，无法更改 CF 应用名称。
- 在推送应用之前创建后端服务。
- 请注意，删除和重新部署应用可能会导致使用随机 route 时产生不同的 DNS 名称。
- 将二进制文件的整个 zip 下载到一个空文件夹中。不要向其中添加新文件。CF push 将上载给定路径中的所有文件。如果添加或更改文件（如图片），HTML 应用将出现问题。
- 提取 Zip 的文件夹结构。这可能是 CF 路径 variable 的一部分。
  - “Appbinary.zip”包括 CF 应用的组件，稍后将使用“cf push”来定位。
  - 确保将 AppBinary 解压缩为仅包含服务组件的单独文件夹，并且不包含图标或元数据文件。
- 系统最多需要 20 分钟来检测部署。
- 每个应用都需要自己的空间。
- 如果应用名称错误，删除应用并使用正确的名称重新部署。系统不会检测到重命名。
- 将角色添加到用户后，此用户需要注销，然后重新登录。
- 在将其发布到 Store 之前，请尽快准备好业务模型。

从开发者处接收应用

从开发者处接收应用是一个三向过程：

1. 开发者在“Developer Cockpit”中将应用分配给操作员。
2. 操作员在“Operator Cockpit”的应用目录中确认接收。
3. 开发者在“Developer Cockpit”中授权移交应用。

授权使用 IoT Value plan 的客户访问您的应用

1. 导航到“Operator Cockpit”中的应用详细信息，然后选择“开通服务”选项卡。
2. 单击“添加至客户”。
3. 输入客户的租户名称和电子邮件地址。

   注意，为应用开通服务可能需要 5 分钟。

2.3 在 IoT Value plan 租户上使用应用

向 IoT Value plan 客户开通应用服务后，应用将在执行以下步骤后显示在 Launchpad 上：

1. 租户管理员将新的应用角色分配给将使用该应用的用户。
2. 用户必须注销才能在其 Launchpad 中看到该应用。

更新现有的 Cloud Foundry 应用

3

3.1 更新用例

本章描述了以下场景：

- 您的操作员租户中部署了一个现有应用。
- 您想要部署应用的新版本。
- 您想要使用旧版应用的后端服务。

更新步骤

如果您已经为 Valuep Plan 开通了应用服务，则应考虑以下步骤：

1. 在 Operator Cockpit 中接受来自开发者的应用。
2. 通知您的客户，他们很快将与应用断开连接。
3. 使用 Operator Cockpit 中的“开通服务”向导，取消向客户开通应用服务。
4. 注销要更新的应用的现有版本。

   无法 Operator Cockpit 中注销，请联系开发支持团队，在您的操作员租户中注销应用。
5. 按照通过 CF 空间共享后端服务凭证 (页 11)部分中的说明进行操作。
6. 部署并注册应用的新版本。
7. 使用 Operator Cockpit 中的“开通服务”向导，为应用的新版本开通服务。

3.2 通过 CF 空间共享后端服务凭证

先决条件

- 应用的早期版本已部署，并且
- 已创建后端服务并将其绑定到此应用。

在以下示例中，应用名为“agentMatrix”。

检查后端服务是否已创建并可用

预先检查以获取后端服务的名称和类型。可以通过 CF 服务完成该操作。

复制到剪贴板

```
CF 服务

name           service         plan      bound apps         last operation
redis-test     redis32         redis-m   agentMatrix        create succeeded
```

创建服务密钥

服务密钥可生成用于手动配置市场服务客户的凭证。

为您的服务完成配置后，本地客户端、其它空间中的应用或部署范围外的实体就可以使用这些密钥访问您的服务。更多信息，请访问：
https://docs.cloudfoundry.org/devguide/services/service-keys.html
要创建密钥，请使用以下命令。

复制到剪贴板

```
>cf create-service-key redis-test mykey
```

提取服务密钥

借助服务密钥，您可以轻松提取凭证。

使用以下命令

复制到剪贴板

```
>cf service-key redis-test mykey
{
 "hostname": "us-cdbr-iron-east-01.cleardb.net",
 "jdbcUrl": "jdbc:mysql://us-cdbr-iron-east-
01.cleardb.net/ad_5bceb00f6a9bae5?user=b74345736ecbf3\u0026password=3374323f",
 "name": "ad_5bceb00f6a9bae5",
 "password":"3374323f",
 "port":"3306",
 "uri": "mysql://b74345736ecbf3:3374323f@us-cdbr-iron-east-
01.cleardb.net:3306/ad_5bceb00f6a9bae5?reconnect=true",
 "username": "b74345736ecbf3"
}
```

提取服务密钥（手动，不推荐）

请注意，一些服务代理不支持服务密钥。在这些情况下，必须手动提取凭证。

在下一步中，我们将从应用的环境 variables 中提取凭证和服务信息。

此操作将通过命令 cf env appname 完成。

我们建议将凭证保存到一个文件中。

复制到剪贴板

```
cf env agentMatrix
Getting env variables for app agentMatrix in org test / space agentMatrix-3.0.0 ...
OK
System-Provided:
{
 "VCAP_SERVICES": {
  "redis32": [
   {
    "binding_name":null,
    "credentials": {
     "dns_servers": [
      "379cd1dc-fdac-89fd.broker.dynamic.redis-service.bosh"
     ],
     "host": "master2.service.dc1.a9ssvc",
     "hosts": [
      "redis-0.node.dc1.a9ssvc",
      "redis-1.node.dc1.a9ssvc",
      "redis-2.node.dc1.a9ssvc"
     ],
     "load_balanced_host": "service.dc1.a9ssvc",
     "password": "password",
     "port":6379
    },
    "instance_name": "redis-test",
    "label": "redis32",
    "name": "redis-test",
    "plan": "redis-m",
    "provider":null,
    "syslog_drain_url":null,
    "tags": [
     "data structure store",
     "database",
     "cache",
     "message broker"
    ],
    "volume_mounts": []
   }
  ]
 }
}
{
 "VCAP_APPLICATION": {
  "application_id": "bf42069d-48f6-453e-bfc5-960bb4df5b32",
```

复制到剪贴板

```
  "application_name": "agentMatrix",
  "application_uris": [
   "agentMatrix.apps.eu1.mindsphere.io"
  ],
  "application_version":"9223a260-d396-436a-8144-5341a605c6c4",
  "cf_api": "https://api.cf.eu1.mindsphere.io",
  "limits": {
   "disk":512,
   "fds":16384,
   "mem":1024
  },
  "name": "agentMatrix",
  "space_id":"82d89932-af13-48ea-b1f3-91eb664332c1",
  "space_name": "agentMatrix-3.0.0",
  "uris": [
   "agentMatrix.apps.eu1.mindsphere.io"
  ],
  "users":null,
  "version":"9223a260-d396-436a-8144-5341a605c6c4"
 }
}
User-Provided:
Deployed:12
No running env variables have been set
No staging env variables have been set
```

导航到新的应用空间

现在，您导航到应使用服务的空间。在示例中，将新应用版本的空间称为 agentMAtrix-3.0.1。

复制到剪贴板

```
cf target -s agentMatrix-3.0.1
```

创建用户提供的服务 (cups)

为在另一个空间中使用该服务，可以创建用户提供的服务 (ups)。

通过用户提供的服务实例，开发者能够使用市场上尚未提供的服务，且其应用可在 Cloud Foundry 上运行，或者以我们的案例为例，可共享来自另一空间的服务。（请参见 https://docs.cloudfoundry.org/devguide/services/user-provided.html）

如果是 redis，ups 需要 host 名、密码、port load_blanced_host 和 hosts 作为参数。

请注意，每个服务的结构都略有不同。但是，命令均为 cf cups instance-name -p {parameter}。

复制到剪贴板

```
cf cups redis-test -p
"{\"host\":\"master.service.dc1.a9ssvc\",\"password\":\"password\",\"port\
":6379,\"load_balanced_host\":\"service.dc1.a9ssvc\", \"hosts\":\"[redis-
0.node.dc1.a9ssvc]\"}"
```

绑定服务

在这里，将 cf cups 命令的服务直接绑定到应用上。无需等待服务创建，因为它只是现有服务的虚拟形式。

复制到剪贴板

```
cf bind-service agentMatrix redis-test
```

重新显示应用

将服务绑定到应用后，重新显示（而不是重启）应用。

复制到剪贴板

```
cf restage agentMatrix
```