



# Red Hat 3scale API Management 2.5

## 安装 3scale

安装和配置 3scale API 管理。



## Red Hat 3scale API Management 2.5 安装 3scale

---

安装和配置 3scale API 管理。

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

## 法律通告

Copyright © 2022 | You need to change the HOLDER entity in the en-US/Installing\_3scale.ent file |.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack<sup>®</sup> Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## 摘要

本指南提供用于安装和配置 3scale API 管理的信息。

# 目录

前言 .....	4
<b>第 1 章 使用 ORACLE 数据库设置 3SCALE 系统镜像 .....</b>	<b>5</b>
1.1. 准备 ORACLE 数据库	5
1.2. 构建系统镜像	6
<b>第 2 章 3SCALE ON OPENSIFT 安装指南 .....</b>	<b>8</b>
2.1. 简介	8
2.1.1. 先决条件	8
2.2. 系统要求	8
2.2.1. 环境要求	8
2.2.2. 硬件要求	8
2.3. 配置节点和权利	9
2.4. 使用模板在 OPENSIFT 上部署 3SCALE	9
2.4.1. 先决条件	9
2.4.2. 导入 3scale 模板	10
2.4.3. 配置 SMTP 变量（可选）	11
2.5. 3SCALE 模板参数	12
2.6. 在 OPENSIFT 中使用带有 3SCALE 的 APICAST	15
2.6.1. 在包含 3scale 的现有 OpenShift 集群上部署 APICAST 模板	15
2.6.2. 从不同的 OpenShift 集群连接 APICAST	16
2.6.3. 更改嵌入式 APICAST 的默认行为	17
2.6.4. 通过内部服务路由在单个 OpenShift 集群上连接多个 APICAST 部署	17
2.6.5. 在其他部署中连接 APICAST	18
2.7. 使用 OPERATOR 部署 3SCALE	18
2.7.1. 先决条件	18
2.7.2. 部署 APIManager 自定义资源	18
2.8. 故障排除	19
2.8.1. 以前的部署导致存在脏的持久性卷声明	19
2.8.2. 从 Docker registry 中拉取错误	20
2.8.3. 在本地挂载持久性卷时 MySQL 的权限问题	20
2.8.4. 无法上传徽标或图像	21
2.8.5. 测试调用无法在 OpenShift 中工作	21
2.8.6. 与 3scale 不同的项目上的 APICAST	21
<b>第 3 章 安装 APICAST .....</b>	<b>23</b>
3.1. 先决条件	23
3.2. 部署选项	23
3.3. 环境	23
3.4. 配置集成设置	23
3.5. 配置服务	24
3.5.1. 声明 API 后端	24
3.5.2. 配置身份验证设置	25
3.5.3. 配置 API 测试调用	25
3.5.4. 保存配置设置	25
<b>第 4 章 在 RED HAT OPENSIFT 上运行 APICAST .....</b>	<b>26</b>
4.1. 先决条件	26
4.2. 设置 RED HAT OPENSIFT	26
4.2.1. 安装 Docker 容器化环境	26
4.2.2. 启动 OpenShift 集群	27
4.2.3. 在远程服务器上设置 OpenShift 集群（可选）	27

---

4.3. 使用 OPENSIFT 模板部署 APICAST	27
4.4. 通过 OPENSIFT 控制台创建路由	28
<b>第 5 章 DOCKER 容器化环境中的 APICAST</b>	<b>32</b>
5.1. 先决条件	32
5.2. 安装 DOCKER 容器化环境	32
5.3. 运行 DOCKER 容器化环境网关	32
5.3.1. Docker 命令选项	33
5.4. 测试 APICAST	33
<b>第 6 章 安装 3SCALE OPERATOR</b>	<b>34</b>
6.1. 简介	34
6.1.1. 先决条件	34
6.2. 创建新的 OPENSIFT 项目	34
6.2.1. 在 OpenShift API 服务器中注册 3scale operator CRD	34
6.2.2. 为 3scale Operator 部署角色和服务账户	35
6.2.3. 部署 3scale Operator	35
<b>第 7 章 3SCALE 高可用性和评估模板</b>	<b>36</b>
7.1. 简介	36
7.2. 先决条件	36
7.3. 高可用性模板	36
7.3.1. 先决条件	36
7.3.2. 使用 HA 模板	36
7.4. 评估模板	37
<b>第 8 章 对 3SCALE 的 REDIS 高可用性(HA)支持</b>	<b>38</b>
8.1. 简介	38
8.2. 为零停机时间设置 REDIS	38
8.3. 为 3SCALE 配置后端组件	39



## 前言

本指南将帮助您安装和配置 3scale



## 第 1 章 使用 ORACLE 数据库设置 3SCALE 系统镜像

本节介绍红帽 3scale API 管理管理员如何使用 Oracle 数据库设置 3scale 系统镜像。默认情况下，3scale 有一个名为 system 的组件，该组件将配置数据存储存储在 MySQL 数据库中。您可以覆盖默认数据库，并将信息存储在外部的 Oracle 数据库中。按照本章中的步骤，使用您自己的 Oracle 数据库客户端二进制文件构建自定义系统容器镜像，并将 3scale 部署到 OpenShift。

### 先决条件

以下 Oracle 软件组建的一个支持的版本：

- Oracle Instant Client Package: Basic 或 Basic Light
- Oracle Instant Client Package: SDK
- Oracle Instant Client Package: ODBC

### 软件包示例

- instantclient-basiclite-linux.x64-12.2.0.1.0.zip 或 instantclient-basic-linux.x64-12.2.0.1.0.zip
- instantclient-sdk-linux.x64-12.2.0.1.0.zip
- instantclient-odbc-linux.x64-12.2.0.1.0-2.zip

要使用 和 Oracle Database 设置 3scale 系统镜像，请执行以下部分中所述的步骤：

- [第 1.1 节 “准备 Oracle 数据库”](#)
- [第 1.2 节 “构建系统镜像”](#)

## 1.1. 准备 ORACLE 数据库

本节介绍准备 Oracle 数据库的步骤。

### 先决条件

- 可以从 OpenShift cluster 访问的一个 Oracle Database 的[支持版本](#)
- 访问 Oracle Database **system** 用户以获取安装过程
- 3scale 2.5 **amp.yml** 模板

### 流程

1. 创建新数据库。

Oracle 数据库需要以下设置才能使用 3scale：

```
ALTER SYSTEM SET max_string_size=extended SCOPE=SPFILE;  
ALTER SYSTEM SET compatible='12.2.0.1' SCOPE=SPFILE;
```

2. 收集数据库详细信息。

获取 3scale 配置需要的信息：

- Oracle 数据库 URL

- Oracle 数据库服务名称
- Oracle 数据库系统密码  
**DATABASE\_URL** 参数必须遵循以下格式：`oracle-enhanced://{user}:{password}@{host}:{port}/{database}`

## 示例

```
DATABASE_URL="oracle-enhanced://user:password@my-oracle-database.com:1521/threescalepdb"
```

## 其他资源

有关在 Oracle 数据库中创建新数据库的详情，请查看 [Oracle 文档](#)。

## 1.2. 构建系统镜像



### 注意

下载系统 Oracle `build.yml` 文件后，您必须手动将 **AMP\_RELEASE** 值从 **2.4.0** 改为 **2.5.0**。

本节提供构建系统镜像的步骤。

## 先决条件

- 必须配置 Oracle 数据库。如需了解更多详细信息，请按照 [第 1.1 节 “准备 Oracle 数据库”](#) 中的步骤操作。

## 流程

1. 克隆 [3scale OpenShift 模板](#) GitHub 存储库。使用以下命令：

```
$ git clone --branch 2.5.0.GA https://github.com/3scale/3scale-amp-openshift-templates.git
```

2. 将您的 Oracle Database Instant 客户端软件包文件放在 **3scale-amp-openshift-templates/amp/system-oracle/oracle-client-files** 目录中。

3. 下载 3scale 2.5 `amp.yml` 模板。

4. 使用 **-f** 选项运行 **oc new-app** 命令并指定 **build.yml** OpenShift 模板：

```
$ oc new-app -f build.yml
```

5. 使用 **-f** 选项运行 **oc new-app** 命令以指示 **amp.yml** OpenShift 模板，并使用 **-p** 选项指定带有 OpenShift 集群域的 **WILDCARD\_DOMAIN** 参数：

```
$ oc new-app -f amp.yml -p WILDCARD_DOMAIN=mydomain.com
```

6. 输入以下 **oc patch** 命令，将 **SYSTEM\_PASSWORD** 替换为您在 [第 1.1 节 “准备 Oracle 数据库”](#) 中设置的 Oracle Database **system** 密码：

```
$ oc patch dc/system-app -p '{"op": "add", "path":
```

```
"/spec/strategy/rollingParams/pre/execNewPod/env/-", "value": {"name":  
"ORACLE_SYSTEM_PASSWORD", "value": "SYSTEM_PASSWORD"}}] --type=json
```

```
$ oc patch dc/system-app -p '{"spec": {"strategy": {"rollingParams": {"post":{"execNewPod":  
{"env": [{"name": "ORACLE_SYSTEM_PASSWORD", "value":  
"SYSTEM_PASSWORD"}]}}}}}}'
```

7. 输入以下命令，替换 **DATABASE\_URL** 以指向 [第 1.1 节“准备 Oracle 数据库”](#) 中指定的 Oracle 数据库：

```
$ oc patch secret/system-database -p '{"stringData": {"URL": "DATABASE_URL"}}'
```

8. 使用以下命令将 pull secret 链接到构建器：

```
$ oc secrets link builder threescale-registry-auth
```

9. 输入 **oc start-build** 命令以构建新系统镜像：

```
$ oc start-build 3scale-amp-system-oracle --from-dir=.
```

## 其他资源

- 有关 3scale 和 Oracle 数据库支持的更多信息，请参阅 [Red Hat 3scale API 管理支持的配置](#)。

## 第 2 章 3SCALE ON OPENSIFT 安装指南

### 2.1. 简介

本节介绍了在 OpenShift 中部署 Red Hat 3scale API Management 2.5 的步骤。

用于内部部署的 3scale 解决方案包括：

- 两个 API 网关：嵌入式 APIcast
- 使用持久性存储的 3scale 管理门户和开发者门户

部署 3scale 解决方案的方法有两种：

- [第 2.4 节 “使用模板在 OpenShift 上部署 3scale”](#)
- [第 2.7 节 “使用 Operator 部署 3scale”](#)



#### 重要

3scale Operator 只是一个技术预览功能。技术预览功能不受红帽产品服务等级协议 (SLA) 支持，且功能可能并不完整。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。有关红帽技术预览功能支持范围的更多信息，请参阅[技术预览功能支持范围](#)。

#### 2.1.1. 先决条件

- 您必须为 UTC（协调世界时间）配置 3scale 服务器。

### 2.2. 系统要求

本节列出了 3scale API 管理 OpenShift 模板的要求。

#### 2.2.1. 环境要求

3scale API 管理需要在 [支持的配置中](#) 指定环境。

持久性卷：

- 3 RWO(ReadWriteOnce)用于 Redis 和 MySQL 持久性的持久性卷
- 1 RWX (ReadWriteMany) 持久性卷用于 CMS 和 System-app Assets

RWX 持久性卷必须配置为可写入组。如需支持所需访问模式的持久卷类型列表，请参阅 [OpenShift 文档](#)。

#### 2.2.2. 硬件要求

硬件要求取决于您的使用需求。红帽建议您测试和配置您的环境，以满足您的特定要求。以下是在 OpenShift 中为 3scale 配置环境时的建议：

- 计算优化的节点用于云环境（AWS c4.2xlarge 或 Azure Standard\_F8）上的部署。

- 如果内存要求超过您当前节点的可用 RAM，则非常大型安装可能需要单独节点（AWS M4 系列或 Azure Av2 系列）。
- 路由和计算任务之间分隔的节点。
- 用于 3scale 特定任务的专用计算节点。
- 将后端侦听器的 **PUMA\_WORKERS** 变量设置为计算节点中的内核数。

## 2.3. 配置节点和权利

在 OpenShift 上部署 3scale 之前，您必须配置节点以及环境所需的授权，以便从红帽获取镜像。

执行以下步骤来配置权利：

1. 在您的每个节点上安装 [Red Hat Enterprise Linux\(RHEL\)](#)。
2. 使用红帽订阅管理器(RHSM)通过 [接口](#) 或 [命令行](#)向红帽注册您的节点。
3. 使用 RHSM [将您的节点附加到 3scale 订阅](#)。
4. 根据以下要求，在节点上 [安装 OpenShift](#)：
  - 使用 [受支持的 OpenShift 版本](#)。
  - 在支持多次写入的文件系统中配置[持久性存储](#)。
5. 安装 [OpenShift 命令行界面](#)。
6. 使用订阅管理器启用对 **rhel-7-server-3scale-amp-2.5-rpms** 存储库的访问权限：

```
sudo subscription-manager repos --enable=rhel-7-server-3scale-amp-2.5-rpms
```

7. 安装 **3scale-amp-template**。模板将保存在 **/opt/amp/templates** 中。

```
sudo yum install 3scale-amp-template
```

## 2.4. 使用模板在 OPENSIFT 上部署 3SCALE

### 2.4.1. 先决条件

- 按照 [第 2.3 节 “配置节点和权利”](#) 中指定的配置了一个 OpenShift 集群。
- 解析到 OpenShift 集群的域（最后是通配符）。
- 访问红帽[容器目录](#)。
- （可选）用于电子邮件功能的已正常工作 SMTP 服务器。

按照以下步骤，使用一个 **.yml** 模板在 OpenShift 上安装 3scale：

- [第 2.4.2 节 “导入 3scale 模板”](#)
- [第 2.4.3 节 “配置 SMTP 变量（可选）”](#)

## 2.4.2. 导入 3scale 模板

执行以下步骤将 3scale 模板导入到 OpenShift 集群中：

1. 在终端会话中以集群管理员身份登录到 OpenShift：

```
oc login
```

2. 选择项目或创建新项目：

```
oc project <project_name>
```

```
oc new-project <project_name>
```

3. 输入 **oc new-app** 命令：

- a. 使用您作为 [配置节点和授权部分](#) 下载的 *amp.yml* 文件的路径指定 **--file** 选项。
- b. 通过将 **WILDCARD\_DOMAIN** 参数设置为 OpenShift 集群的域来指定 **--param** 选项。
- c. （可选）将 **WILDCARD\_POLICY** 参数设置为 **subdomain** 来指定 **--param** 选择来启用通配符域路由：

没有通配符路由：

```
oc new-app --file /opt/amp/templates/amp.yml --param WILDCARD_DOMAIN=
<WILDCARD_DOMAIN>
```

使用通配符路由：

```
oc new-app --file /opt/amp/templates/amp.yml --param WILDCARD_DOMAIN=
<WILDCARD_DOMAIN> --param WILDCARD_POLICY=Subdomain
```

终端会显示新创建的 3scale 管理门户的主和租户 URL 以及凭据。这个输出应包括以下信息：

- master admin 用户名
- master 密码
- 主令牌信息
- 租户用户名
- 租户密码
- 租户令牌信息

4. 以 `admin/xXxYyz123` 身份登录 <https://user-admin.3scale-project.example.com>。

```
* With parameters:
```

```
* ADMIN_PASSWORD=xXxYyz123 # generated
```

```
* ADMIN_USERNAME=admin
```

```
* TENANT_NAME=user
```

```
* MASTER_NAME=master
```

```
* MASTER_USER=master
* MASTER_PASSWORD=xXxXyz123 # generated

--> Success
Access your application via route 'user-admin.3scale-project.example.com'
Access your application via route 'master-admin.3scale-project.example.com'
Access your application via route 'backend-user.3scale-project.example.com'
Access your application via route 'user.3scale-project.example.com'
Access your application via route 'api-user-apicast-staging.3scale-project.example.com'
Access your application via route 'api-user-apicast-production.3scale-project.example.com'
Access your application via route 'apicast-wildcard.3scale-project.example.com'
```

- 记下这些详细信息，以备将来参考。



### 注意

您可能需要等待几分钟，以便 3scale 在 OpenShift 上完全部署，以便您的登录和凭据正常工作。

### 更多信息

如需有关 OpenShift 上通配符域的信息，请访问 [使用通配符路由（用于子域）](#)。

### 2.4.3. 配置 SMTP 变量（可选）

OpenShift 使用电子邮件 [发送通知](#) 并 [邀请新用户](#)。如果要使用这些功能，您必须提供自己的 SMTP 服务器并在 SMTP 配置映射中配置 SMTP 变量。

执行以下步骤在 SMTP 配置映射中配置 SMTP 变量：

- 如果您还没有登录，请登录到 OpenShift：

```
oc login
```

- 为 SMTP 配置映射配置变量。使用 `oc patch` 命令，指定 `configmap` 和 `smtp` 对象，后跟 `-p` 选项，并在 JSON 中为以下变量编写以下新值：

变量	描述
address	允许您将远程邮件服务器指定为中继
username	指定您的邮件服务器用户名
password	指定您的邮件服务器密码
domain	指定 HELO 域
port	指定邮件服务器侦听新连接的端口

变量	描述
身份验证	指定邮件服务器的身份验证类型。允许的值： <b>plain</b> （发送明文中的密码）、 <b>login</b> （发送密码 Base64 编码）或 <b>cram_md5</b> （交换信息和加密消息目标 5 算法以散列重要信息）
openssl.verify.mode	指定在使用 TLS 时，OpenSSL 如何检查证书。允许的值： <b>none</b> 、 <b>peer</b> 、 <b>client_once</b> 或 <b>fail_if_no_peer_cert</b> 。

### 示例

```
oc patch configmap smtp -p '{"data":{"address":"<your_address>"}'}
oc patch configmap smtp -p '{"data":{"username":"<your_username>"}'}
oc patch configmap smtp -p '{"data":{"password":"<your_password>"}'}
```

- 在设置了 configmap 变量后，重新部署 **system-app** 和 **system-sidekiq** pod：

```
oc rollout latest dc/system-app
oc rollout latest dc/system-sidekiq
```

## 2.5. 3SCALE 模板参数

模板参数配置部署期间和之后 3scale(*amp.yml*)模板的环境变量。



### 注意

在 3scale 2.5 中，PostgreSQL 版本已从 9 更新至 10。我们强烈建议您进行 PostgreSQL 配置。请参阅 *Migrating 3scale* 文档中的 [Upgrade Zync Database PostgreSQL 9.5 到 10](#)。

名称	描述	默认值	必需？
APP_LABEL	用于对象应用程序标签	<b>3scale-api-management</b>	是
ZYNC_DATABASE_PASSWORD	PostgreSQL 连接用户的密码。如果未提供，则随机生成。	N/A	是
ZYNC_SECRET_KEY_BASE	适用于 Zync 的机密密钥基础。如果未提供，则随机生成。	N/A	是
ZYNC_AUTHENTICATION_TOKEN	Zync 的身份验证令牌。如果未提供，则随机生成。	N/A	是
AMP_RELEASE	3scale 发行标签。	<b>2.5.0</b>	是



名称	描述	默认值	必需？
ADMIN_PASSWORD	一个随机生成的 3scale 管理员帐户密码。	N/A	是
ADMIN_USERNAME	3scale 管理员帐户用户名。	<b>admin</b>	是
APICAST_ACCESS_TOKEN	阅读 APIcast 将用来下载其配置的唯一访问令牌。	N/A	是
ADMIN_ACCESS_TOKEN	具有所有范围的管理访问令牌，以及用于 API 访问的写入权限。	N/A	否
WILDCARD_DOMAIN	通配符路由的根域。例如，根域 <b>example.com</b> 将生成 <b>3scale-admin.example.com</b> 。	N/A	是
WILDCARD_POLICY	通过将值设置为 "Subdomain" 来启用到嵌入式 APIcast 网关的通配符路由	<b>无</b>	是
TENANT_NAME	管理门户的 root 下具有 -admin 后缀的租户名称。	<b>3scale</b>	是
MYSQL_USER	用于访问数据库的 MySQL 用户的用户名。	<b>mysql</b>	是
MYSQL_PASSWORD	MySQL 用户的密码。	N/A	是
MYSQL_DATABASE	访问的 MySQL 数据库的名称。	<b>system</b>	是
MYSQL_ROOT_PASSWORD	Root 用户的密码。	N/A	是
SYSTEM_BACKEND_USERNAME	用于内部 3scale api 身份验证的内部 3scale API 用户名。	<b>3scale_api_user</b>	是
SYSTEM_BACKEND_PASSWORD	用于内部 3scale api 身份验证的内部 3scale API 密码。	N/A	是

名称	描述	默认值	必需？
REDIS_IMAGE	使用的 redis 镜像	<b>registry.access.redhat.com/rhsc/redis-32-rhel7:3.2</b>	是
MYSQL_IMAGE	要使用的 MySQL 镜像	<b>registry.access.redhat.com/rhsc/mysql-57-rhel7:5.7</b>	是
MEMCACHED_IMAGE	要使用的 Memcached 镜像	<b>registry.access.redhat.com/3scale-amp20/memcached:1.4.15</b>	是
POSTGRES_IMAGE	要使用的 PostgreSQL 镜像	<b>registry.access.redhat.com/rhsc/postgresql-10-rhel7</b>	是
AMP_SYSTEM_IMAGE	要使用的 3scale 系统镜像	<b>registry.access.redhat.com/3scale-amp25/system</b>	是
AMP_BACKEND_IMAGE	要使用的 3scale 后端镜像	<b>registry.access.redhat.com/3scale-amp25/backend</b>	是
AMP_APICAST_IMAGE	使用的 3scale APIcast 镜像	<b>registry.access.redhat.com/3scale-amp25/apicast-gateway</b>	是
AMP_ROUTER_IMAGE	要使用的 3scale 通配符路由器镜像	<b>registry.access.redhat.com/3scale-amp22/wildcard-router</b>	是
AMP_ZYNC_IMAGE	要使用的 3scale Zync 镜像	<b>registry.access.redhat.com/3scale-amp25/zync</b>	是
SYSTEM_BACKEND_SHARED_SECRET	用于将事件从后端导入到系统的共享机密。	N/A	是
SYSTEM_APP_SECRET_KEY_BASE	系统应用程序 secret 密钥基础	N/A	是

名称	描述	默认值	必需？
APICAST_MANAGEMENT_API	APICAST 管理 API 的范围。可以禁用、状态或调试。至少健康检查所需的状况。	<b>status</b>	否
APICAST_OPENSSL_VERIFY	下载配置时打开/关闭 OpenSSL 对等验证。可以设置为 true/false。	<b>false</b>	否
APICAST_RESPONSE_CODES	在 APICAST 中启用日志响应代码。	true	否
APICAST_REGISTRY_URL	解析到 APICAST 策略位置的 URL	<a href="http://apicast-staging:8090/policies">http://apicast-staging:8090/policies</a>	是
MASTER_USER	Master 管理员帐户用户名	<b>master</b>	是
MASTER_NAME	master 管理门户的子域值将附加 <b>-master</b> 后缀	<b>master</b>	是
MASTER_PASSWORD	随机生成的 master 管理员密码	N/A	是
MASTER_ACCESS_TOKEN	具有用于 API 调用 master 级别的权限的令牌	N/A	是
IMAGESTREAM_TAG_IMPORT_INSECURE	如果服务器可以在镜像导入过程中绕过证书验证或直接通过 HTTP 连接，则设置为 true。	<b>false</b>	是

## 2.6. 在 OPENSIFT 中使用带有 3SCALE 的 APICAST

APICAST 可用于 3scale 托管 API Manager，以及 OpenShift Container Platform 中的内部安装。两者的配置过程都有所不同。本节介绍如何在 OpenShift 中使用 API Manager 部署 APICAST。

### 2.6.1. 在包含 3scale 的现有 OpenShift 集群上部署 APICAST 模板

3scale OpenShift 模板默认包含两个嵌入式 APICAST API 网关。如果需要更多 API 网关，或需要单独的 APICAST 部署，您可以在 OpenShift 集群中部署额外的 APICAST 模板。

执行以下步骤在 OpenShift 集群中部署额外的 API 网关：

1. 使用以下配置创建访问令牌：
  - 作用于帐户管理 API

- 具有只读访问权限

2. 登录到您的 APIcast 集群：

```
oc login
```

3. 创建一个允许 APIcast 与 3scale 通信的 secret。指定 **new-basicauth**、**apicast-configuration-url-secret** 和 **--password** 参数，以及 3scale 部署的访问令牌、租户名称和通配符域：

```
oc secret new-basicauth apicast-configuration-url-secret --
password=https://<APICAST_ACCESS_TOKEN>@<TENANT_NAME>-admin.
<WILDCARD_DOMAIN>
```



### 注意

**TENANT\_NAME** 是管理门户可访问的 root 下的名称。**TENANT\_NAME** 的默认值为 **3scale**。如果您在 3scale 部署中使用了自定义值，则必须使用该值。

4. 使用 **oc new-app** 命令导入 APIcast 模板，并使用 **apicast.yml** 文件指定 **--file** 选项：

```
oc new-app --file /opt/amp/templates/apicast.yml
```



### 注意

首先安装 APIcast 模板，如 [第 2.3 节“配置节点和权利”](#) 所述。

## 2.6.2. 从不同的 OpenShift 集群连接 APIcast

如果在 3scale 集群外在不同的 OpenShift 集群上部署 APIcast，则必须通过公共路由连接：

1. 使用以下配置创建 [访问令牌](#)：

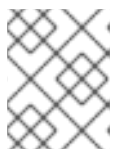
- 作用于帐户管理 API
- 具有只读访问权限

2. 登录到您的 APIcast 集群：

```
oc login
```

3. 创建一个允许 APIcast 与 3scale 通信的 secret。指定 **new-basicauth**、**apicast-configuration-url-secret** 和 **--password** 参数，以及 3scale 部署的访问令牌、租户名称和通配符域：

```
oc secret new-basicauth apicast-configuration-url-secret --
password=https://<APICAST_ACCESS_TOKEN>@<TENANT_NAME>-admin.
<WILDCARD_DOMAIN>
```



### 注意

**TENANT\_NAME** 是管理门户可访问的 root 下的名称。**TENANT\_NAME** 的默认值为 **3scale**。如果您在 3scale 部署中使用了自定义值，则必须使用该值。

4. 使用 **oc new-app** 命令，在 OpenShift 集群外的 OpenShift 集群上部署 APIcast。指定 **--file** 选项以及 **apicast.yml** 文件的路径：

```
oc new-app --file /path/to/file/apicast.yml
```

### 2.6.3. 更改嵌入式 APIcast 的默认行为

在外部 APIcast 部署中，您可以通过 [更改 APIcast OpenShift 模板中的模板参数](#) 来修改默认行为。

在嵌入式 APIcast 部署中，3scale 和 APIcast 从单一模板部署。如果要更改嵌入 APIcast 部署的默认行为，您必须在部署后修改环境变量。

### 2.6.4. 通过内部服务路由在单个 OpenShift 集群上连接多个 APIcast 部署

如果您将多个 APIcast 网关部署到同一个 OpenShift 集群中，您可以将它们配置为通过后端侦听器服务（而非默认的外部路由配置）使用内部路由进行连接。

您必须安装 OpenShift SDN 插件才能通过内部服务路由进行连接。如何连接取决于您安装的 SDN：

#### ovs-subnet

如果您使用 **ovs-subnet** OpenShift 软件定义型网络 (SDN) 插件，请执行以下步骤通过内部路由连接：

1. 如果还没有登录，请登录到您的 OpenShift 集群：

```
oc login
```

2. 输入以下命令显示 **backend-listener** 路由 URL：

```
oc route backend-listener
```

3. 使用到 **apicast.yml** 的路径输入 **oc new-app** 命令：

```
oc new-app -f apicast.yml
```

#### ovs-multitenant

如果使用 **ovs-multitenant** OpenShift SDN 插件，请执行以下步骤通过内部路由连接：

1. 如果还没有登录，请登录到您的 OpenShift 集群：

```
oc login
```

2. 作为管理员，在 **oadm** 命令中使用 **pod-network** 和 **join-projects** 选项来设置两个项目之间的通信：

```
oadm pod-network join-projects --to=<3SCALE_PROJECT> <APICAST_PROJECT>
```

3. 输入以下命令显示 **backend-listener** 路由 URL：

```
oc route backend-listener
```

4. 使用到 **apicast.yml** 的路径输入 **oc new-app** 命令：

-

```
oc new-app -f apicast.yml
```

## 更多信息

如需有关 OpenShift SDN 和项目网络隔离的信息，请参阅 [OpenShift SDN](#)。

### 2.6.5. 在其他部署中连接 APIcast

如果在 Docker 上部署 APIcast，您可以通过将 OpenShift 上部署的 3scale 设为 3scale 来连接 OpenShift 上部署的 APIcast，方法是将 **THREESCALE\_PORTAL\_ENDPOINT** 参数设置为在 OpenShift 上部署的 3scale 管理门户的 URL 和访问令牌。在这种情况下，您不需要设置 **BACKEND\_ENDPOINT\_OVERRIDE** 参数。

如需了解更多详细信息，请参阅 [第 5 章 Docker 容器化环境中的 APIcast](#)。

## 2.7. 使用 OPERATOR 部署 3SCALE

本节介绍了使用 *APIManager* 自定义资源通过 3scale 操作器安装和部署 3scale 解决方案。



### 重要

3scale Operator 只是一个技术预览功能。技术预览功能不受红帽产品服务等级协议 (SLA) 支持，且功能可能并不完整。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。有关红帽技术预览功能支持范围的更多信息，请参阅 [技术预览功能支持范围](#)。

### 2.7.1. 先决条件

- 首先使用 Operator 部署 3scale [需要安装 3scale Operator](#)
- OpenShift Container Platform 3.11
  - 在 OpenShift 集群中具有管理员特权的用户帐户

### 2.7.2. 部署 *APIManager* 自定义资源

部署 *APIManager* 自定义资源将使 Operator 开始处理，并从中部署 3scale 解决方案。

1. 通过创建一个包含以下内容的新 YAML 文件来部署 *APIManager*：



### 注意

*wildcardDomain* 参数可以是您想要将该解析为 IP 地址的任何所需名称，而 IP 地址是一个有效的 DNS 域。*wildcardPolicy* 参数只能是 **None** 或 **Subdomain**。务必删除参数的占位符标记：`< >`。

```
apiVersion: apps.3scale.net/v1alpha1
kind: APIManager
metadata:
  name: example-apimanager
spec:
  productVersion: "2.5"
```

```
wildcardDomain: <wildcardDomain>
wildcardPolicy: <None|Subdomain>
resourceRequirementsEnabled: true
```

2. 如果 **wildcardPolicy** 是 **Subdomain**，则在 OpenShift 路由器级别启用通配符路由。
  - a. 您可以运行以下命令来完成此操作：

```
oc set env dc/router ROUTER_ALLOW_WILDCARD_ROUTES=true -n default
```



### 注意

如需有关 *APIManager* 字段的更多信息，请参阅[参考文档](#)。

```
export NAMESPACE="operator-test"
oc project ${NAMESPACE}
oc create -f <yaml-name>
```

- b. 这应该会在 *operator-test* 项目中触发 3scale 解决方案的部署。

## 2.8. 故障排除

这部分包含常见安装问题列表，并为它们解决提供指导。

- [第 2.8.1 节 “以前的部署导致存在脏的持久性卷声明”](#)
- [第 2.8.2 节 “从 Docker registry 中拉取错误”](#)
- [第 2.8.3 节 “在本地挂载持久性卷时 MySQL 的权限问题”](#)
- [第 2.8.4 节 “无法上传徽标或图像”](#)
- [第 2.8.5 节 “测试调用无法在 OpenShift 中工作”](#)
- [第 2.8.6 节 “与 3scale 不同的项目上的 APIcast”](#)

### 2.8.1. 以前的部署导致存在脏的持久性卷声明

#### 问题

之前的部署尝试导致脏的持久性卷声明(PVC)，从而导致 MySQL 容器无法启动。

#### 原因

在 OpenShift 中删除项目不会清理与其关联的 PVC。

#### 解决方案

1. 使用 **oc get pvc** 命令查找包含错误 MySQL 数据的 PVC：

```
# oc get pvc
NAME                STATUS  VOLUME  CAPACITY  ACCESSMODES  AGE
backend-redis-storage Bound   vol003  100Gi    RWO,RWX      4d
```

```
mysql-storage      Bound  vol006  100Gi  RWO,RWX  4d
system-redis-storage Bound  vol008  100Gi  RWO,RWX  4d
system-storage     Bound  vol004  100Gi  RWO,RWX  4d
```

2. 点 OpenShift UI 中的 **cancel** 部署，以停止 system-mysql 容器集的部署。
3. 删除 MySQL 路径下的所有内容，以清理卷。
4. 启动新的 **system-mysql** 部署。

## 2.8.2. 从 Docker registry 中拉取错误

### 问题

在安装过程中出现以下错误：

```
svc/system-redis - 1EX.AMP.LE.IP:6379
dc/system-redis deploys docker.io/rhscf/redis-32-rhel7:3.2-5.3
deployment #1 failed 13 minutes ago: config change
```

### 原因

OpenShift 通过发出 **docker** 命令来搜索和调取容器镜像。此命令引用 **docker.io** Docker registry，而不是 **registry.access.redhat.com** 红帽容器 registry。

当系统包含 Docker 容器化环境的意外版本时，会出现这种情况。

### 解决方案

使用适当的 Docker 容器化环境版本。

## 2.8.3. 在本地挂载持久性卷时 MySQL 的权限问题

### 问题

system-msql pod 崩溃且不部署，从而导致其他系统依赖它失败部署。pod 日志显示以下错误：

```
[ERROR] Cannot start server : on unix socket: Permission denied
[ERROR] Do you already have another mysqld server running on socket: /var/lib/mysql/mysql.sock ?
[ERROR] Aborting
```

### 原因

MySQL 进程启动时不正确的用户权限。

### 解决方案

1. 用于持久性卷的目录 MUST 具有 root 组的写入权限。对 root 用户具有 rw 权限并不足够，因为 MySQL 服务以 root 组中的不同用户身份运行。以 root 用户身份执行以下命令：

```
chmod -R g+w /path/for/pvs
```

2. 执行以下命令以防止 SELinux 阻止访问：

```
chcon -Rt svirt_sandbox_file_t /path/for/pvs
```



■

## 2.8.4. 无法上传徽标或图像

### 问题

无法上传徽标 - **system-app** 日志显示以下错误：

```
Errno::EACCES (Permission denied @ dir_s_mkdir - /opt/system/public//system/provider-name/2
```

### 原因

OpenShift 无法写入持久卷。

### 解决方案

确保您的持久卷可由 OpenShift 写入。它应归 root 组所有，并且可写入组。

## 2.8.5. 测试调用无法在 OpenShift 中工作

### 问题

测试调用在创建新服务和 OpenShift 上的路由后无法正常工作。通过 curl 直接调用也失败，带有 **service not available** 信息。

### 原因

3scale 默认需要 HTTPS 路由，并且 OpenShift 路由不受保护。

### 解决方案

确保 OpenShift 路由器设置中点击了 **安全路由** 复选框。

## 2.8.6. 与 3scale 不同的项目上的 APIcast

### 问题

APIcast 部署失败（pod 没有变为蓝色）。以下错误会出现在日志中：

```
update acceptor rejected apicast-3: pods for deployment "apicast-3" took longer than 600 seconds to become ready
```

pod 中出现以下错误：

```
Error syncing pod, skipping: failed to "StartContainer" for "apicast" with RunContainerError: "GenerateRunContainerOptions: secrets \"apicast-configuration-url-secret\" not found"
```

### 原因

该机密没有正确设置。

### 解决方案

使用 APIcast v3 创建 secret 时，指定 **apicast-configuration-url-secret**：

```
oc secret new-basicauth apicast-configuration-url-secret --  
password=https://<ACCESS_TOKEN>@<TENANT_NAME>-admin.<WILDCARD_DOMAIN>
```

## 第 3 章 安装 APICAST

APICast 是基于 NGINX 的 API 网关，用于将您的内部和外部 API 服务与 3scale Platform 集成。APICast 利用循环实现负载均衡。

在本指南中，您将了解部署选项、提供的环境以及如何入门。

要获取有关最新发布的支持 APICast 版本的信息，请参阅 [Red Hat 3scale API 管理支持的配置](#) 和 [Red Hat 3scale API Management - 组件详情](#)。

### 3.1. 先决条件

APICast 不是独立的 API 网关。需要连接到 3scale API Manager。

- 您需要一个正常工作的 3scale [On-Premises](#) 实例。

### 3.2. 部署选项

您可以使用托管或自我管理的 APICast。在这两种情况下，APICast 必须连接到 3scale API 管理平台的其余部分：

- **嵌入式 APICast**：默认情况下两个 APICast 网关（staging 和 production）会随 3scale API Management 安装一起提供。它们预先配置好，随时开箱即用。
- **自我管理 APICast**：您可以随时部署 APICast。以下是部署 APICast 的几个推荐选项：
  - **Docker 容器化环境**：下载可直接使用 Docker 格式的容器镜像，其中包含用于在 Docker 格式的容器中运行 APICast 的所有依赖项。
  - **OpenShift**：在受支持的 [OpenShift 版本上运行](#) APICast。您可以将自我管理的 APICasts 连接到 3scale 安装或 3scale 在线帐户。

### 3.3. 环境

默认情况下，当您创建 3scale 帐户时，您将在两个不同的环境中获取嵌入式 APICast：

- **Stage**：仅在配置和测试 API 集成时使用。当您确认您的设置按预期工作时，您可以选择将其部署到生产环境中。OpenShift 模板设置 Staging APICast 的参数，以便在每次 API 调用上重新载入配置(**APICAST\_CONFIGURATION\_LOADER: lazy, APICAST\_CONFIGURATION\_CACHE: 0**)。快速测试 APICast 配置中的更改非常有用。
- **Production**：此环境专用于生产环境。在 OpenShift 模板中为 Production APICast 设置以下参数：**APICAST\_CONFIGURATION\_LOADER: boot, APICAST\_CONFIGURATION\_CACHE: 300**。这意味着，在 APICast 启动时将完全加载配置，并将缓存 300 秒（5 分钟）。5 分钟后将重新加载配置。这意味着，当您将配置提升到生产环境时，可能需要 5 分钟才能应用，除非您触发新的 APICast 部署。

### 3.4. 配置集成设置

进入 `[your_API_name] > Integration > Configuration`

在 Integration 页面的顶部，您将看到集成选项。默认情况下，部署选项是托管的 APICast，身份验证模式是 API 密钥。您可以通过单击右上角的 [编辑集成设置](#) 来更改这些设置。

在 Integration 页面的顶部，您将看到集成选项。默认情况下，您可以找到这些值：

- 部署选项：嵌入式 APIcast。
- 身份验证模式：API 密钥。

您可以通过单击右上角的编辑集成设置来更改这些设置。

## 3.5. 配置服务

### 3.5.1. 声明 API 后端

您需要在 Private Base URL 字段中声明您的 API 后端，它是 API 后端的端点主机。在处理了所有身份验证、授权、速率限值和统计数据后，APIcast 将所有流量重定向到您的 API 后端。

通常，您的 API 的私有基本 URL 将会像 <https://api-backend.yourdomain.com:443> 一样在您管理的域中 ([yourdomain.com](https://yourdomain.com))。例如，如果您与 Twitter API 集成，则私有基本 URL 为 <https://api.twitter.com/>。在本例中，使用由 3scale 托管的 Echo API - 它是一个简单的 API，可以接受任何路径并返回与请求相关的信息（路径、请求参数、标头等）。其私有基础 URL 是 <https://echo-api.3scale.net:443>

Staging: configure & test your integration [documentation](#)

deployed | [deployment history](#)

The screenshot shows a configuration form for an API. On the left, there is a green puzzle piece icon labeled 'API'. The main form area has a label 'Private Base URL\*' followed by a text input field containing the value 'https://echo-api.3scale.net:443'. Below the input field, there is a small text note: 'Private address of your API that will be called by the API gateway.' In the top right corner of the form area, there is a question mark icon.

测试您的私有（非受管）API 是否正常工作。例如，对于 Echo API，您可以使用 **curl** 命令进行以下调用：

```
curl "https://echo-api.3scale.net:443"
```

您将获得以下响应：

```
{
  "method": "GET",
  "path": "/",
  "args": "",
  "body": "",
  "headers": {
    "HTTP_VERSION": "HTTP/1.1",
    "HTTP_HOST": "echo-api.3scale.net",
    "HTTP_ACCEPT": "*/*",
    "HTTP_USER_AGENT": "curl/7.51.0",
    "HTTP_X_FORWARDED_FOR": "2.139.235.79, 10.0.103.58",
    "HTTP_X_FORWARDED_HOST": "echo-api.3scale.net",
    "HTTP_X_FORWARDED_PORT": "443",
    "HTTP_X_FORWARDED_PROTO": "https",
    "HTTP_FORWARDED": "for=10.0.103.58;host=echo-api.3scale.net;proto=https"
  },
  "uuid": "ee626b70-e928-4cb1-a1a4-348b8e361733"
}
```

### 3.5.2. 配置身份验证设置

您可以在 **AUTHENTICATION SETTINGS** 部分中配置 API 身份验证设置。以下字段都是可选：

字段	描述
Host Header	定义自定义主机请求标头。如果您的 API 后端仅接受来自特定主机的流量，则需要此设置。
Secret Token	用于阻止开发人员向 API 后端发出请求。设置此处标头的值，并确保您的后端只允许使用此机密标头调用。
Credentials location	定义凭据是作为 HTTP 标头、查询参数还是 HTTP 基本身份验证传递。
Auth user key	设置与凭证位置关联的用户密钥
错误	为以下错误定义响应代码、内容类型和响应正文：身份验证失败、缺失身份验证，无匹配项。

### 3.5.3. 配置 API 测试调用

您需要为托管暂存环境配置测试调用。在 **API test GET 请求项** 中输入您的 API 的路径（例如，`/v1/word/good.json`）。

### 3.5.4. 保存配置设置

点页面右下角的 **Update & Test Staging Configuration** 按钮保存设置。这会将 APICast 配置部署到 3scale 托管的 staging 环境。如果一切配置正确，则左侧的垂直行应当会为绿色。

如果您使用一个自管理的部署选项，请从 GUI 保存配置，并通过在 staging 或 production 公共基础 URL 字段中添加正确的主机来确保指向部署的 API 网关。在对生产网关进行任何调用前，请不要忘记点 **Promote v.x to Production** 按钮。

找到 staging 部分底部的 **curl** 示例，并从控制台运行它：

```
curl "https://XXX.staging.apicast.io:443/v1/word/good.json?user_key=YOUR_USER_KEY"
```



#### 注意

您应该获得与上述相同的响应，但这次请求将通过 3scale 托管 APICast 实例。注意：您应该确保您拥有具有服务有效凭证的应用程序。如果您使用在注册到 3scale 时创建的默认 API 服务，您应该已有一个应用程序。如果您在测试 curl 中看到 **USER\_KEY** 或 **APP\_ID** 和 **APP\_KEY** 值，您需要首先为该服务创建一个应用程序。

现在，您的 API 与 3scale 集成。

3scale 托管的 APICast 网关进行凭证验证，并应用您为应用程序计划定义的速率限制。如果您试图在没有凭证的情况下调用，或者带有无效凭证，您会看到出错信息。

## 第 4 章 在 RED HAT OPENSIFT 上运行 APICAST

本教程介绍了如何在 Red Hat OpenShift 中部署 APIcast API 网关。

### 4.1. 先决条件

要遵循以下教程步骤，首先需要根据 [安装 APIcast 指南](#) 在 3scale 管理门户中配置 APIcast。确保将 *自助管理的网关* 选为集成设置中的部署选项。您应该已将 Staging 和 Production 环境都配置为继续操作。

### 4.2. 设置 RED HAT OPENSIFT

如果您已有一个正在运行的 OpenShift 集群，您可以跳过这一步。否则，继续阅读。

对于生产环境，您可以按照 [OpenShift 安装的说明](#) 进行操作。

在本教程中，将通过以下命令安装 OpenShift 集群：

- Red Hat Enterprise Linux (RHEL) 7
- Docker 容器化环境 v1.10.3
- OpenShift Origin 命令行界面(CLI)- v1.3.1

#### 4.2.1. 安装 Docker 容器化环境

红帽提供的 Docker 格式容器镜像作为 RHEL 中的 Extras 频道的一部分发布。要启用附加软件仓库，您可以使用 [Subscription Manager](#) 或 yum config Manager。详情请查看 [RHEL 产品文档](#)。

对于部署在 AWS EC2 实例上的 RHEL 7，您将使用以下说明：

1. 列出所有软件仓库：

```
sudo yum repolist all
```

2. 查找并启用 **\*-extras** 仓库。

```
sudo yum-config-manager --enable rhui-REGION-rhel-server-extras
```

3. 安装 Docker 格式的容器镜像：

```
sudo yum install docker docker-registry
```

4. 通过在 `/etc/sysconfig/docker` 文件中添加或取消注释以下行来添加不安全的 registry **172.30.0.0/16**：

```
INSECURE_REGISTRY='--insecure-registry 172.30.0.0/16'
```

5. 启动 Docker 服务：

```
sudo systemctl start docker
```

使用以下命令，您可以验证容器服务是否正在运行：

■

```
sudo systemctl status docker
```

## 4.2.2. 启动 OpenShift 集群

从 [OpenShift 发行页面](#) 下载客户端工具的最新稳定版本(**openshift-origin-client-tools-VERSION-linux-64bit.tar.gz**), 并将从存档中提取的 Linux **oc** 二进制文件放在 **PATH** 中。



### 注意

- 请注意, **oc cluster** 命令只在 1.3+ 或更高版本中提供。
- **docker** 命令以 **root** 用户身份运行, 因此您需要使用 **root** 特权运行任何 **oc** 或 **docker** 命令。

打开终端, 该用户有运行 **docker** 命令的权限并运行 :

```
oc cluster up
```

在输出底部, 您可以找到有关部署的集群的信息 :

```
-- Server Information ...
OpenShift server started.
The server is accessible via web console at:
https://172.30.0.112:8443

You are logged in as:
User: developer
Password: developer

To login as administrator:
oc login -u system:admin
```

注意分配给您的 OpenShift 服务器的 IP 地址。您将在教程中将其指代为 **OPENSIFT-SERVER-IP**。

## 4.2.3. 在远程服务器上设置 OpenShift 集群 (可选)

如果要在远程服务器上部署 OpenShift 集群, 则需要在启动集群时明确指定公共主机名和路由后缀, 以便您可以远程访问 OpenShift Web 控制台。

例如, 如果您要在 AWS EC2 实例上部署, 您应该指定以下选项 :

```
oc cluster up --public-hostname=ec2-54-321-67-89.compute-1.amazonaws.com --routing-
suffix=54.321.67.89.xip.io
```

其中 **ec2-54-321-67-89.compute-1.amazonaws.com** 是公共域, 而 **54.321.67.89** 是实例的 IP。然后, 您将可以访问位于 <https://ec2-54-321-67-89.compute-1.amazonaws.com:8443> 的 OpenShift Web 控制台。

## 4.3. 使用 OPENSIFT 模板部署 APICAST

1. 默认情况下, 您以 *developer* 用户身份登录, 并可继续下一步。

否则，请从您在上一步中下载和安装的 OpenShift 客户端工具使用 **oc login** 命令登录 OpenShift。默认登录凭证为 `username = "developer"` 和 `password = "developer"`：

```
oc login https://OPENSIFT-SERVER-IP:8443
```

您应当在输出中看到 **Login successful.**。

2. 创建您的项目。本例设置显示名为 `gateway`

```
oc new-project "3scalegateway" --display-name="gateway" --description="3scale gateway demo"
```

响应应如下所示：

```
Now using project "3scalegateway" on server "https://172.30.0.112:8443".
```

忽略命令提示符处文本输出中的建议后续步骤，再继续下面的下一步。

3. 通过将 `<access_token>` 和 `<domain>` 替换为您自己的凭证来创建新 secret 来引用项目。有关 `<access_token>` 和 `<domain>` 的更多信息，请参阅下方。

```
oc create secret generic apicast-configuration-url-secret --from-literal=password=https://<access_token>@<admin_portal_domain> --type=kubernetes.io/basic-auth
```

这里的 `<access_token>` 是 3scale 帐户管理 API 的 **访问令牌**（而不是服务令牌），`<domain>-admin.3scale.net` 是 3scale 管理门户的 URL。

响应应如下所示：

```
secret/apicast-configuration-url-secret
```

4. 从模板为您的 APIcast 网关创建一个应用程序，并启动部署：

```
oc new-app -f https://raw.githubusercontent.com/3scale/3scale-amp-openshift-templates/2.5.0.GA/apicast-gateway/apicast.yml
```

您应该在输出的底部看到以下信息：

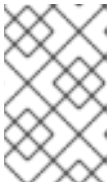
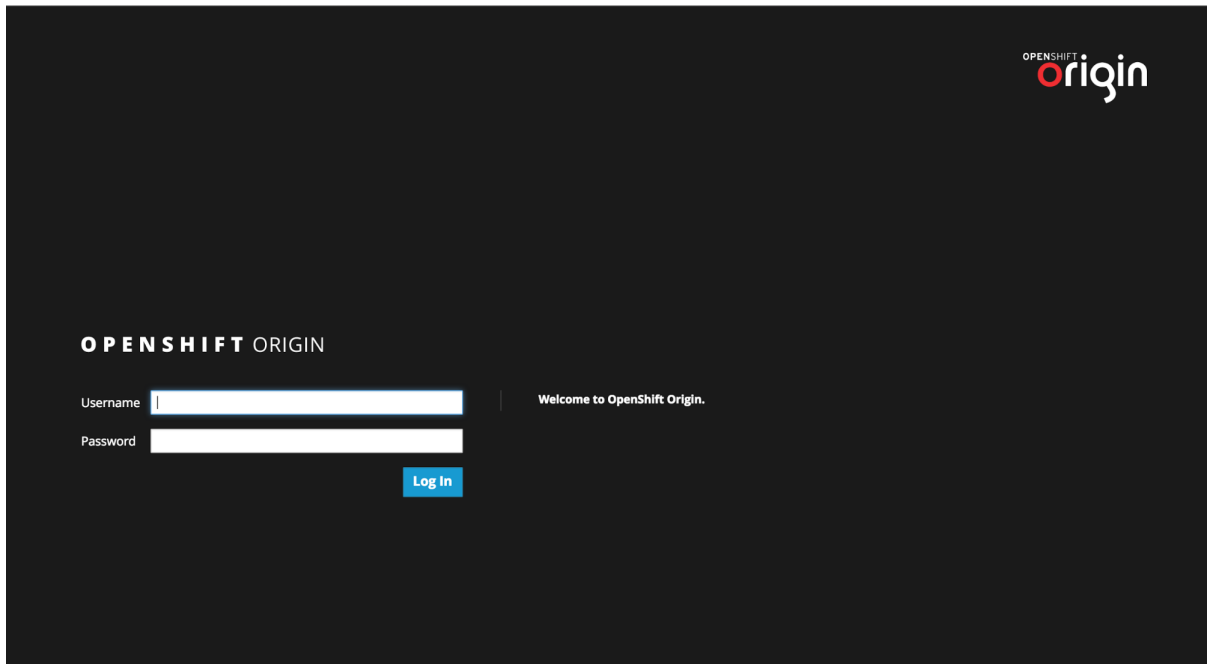
```
--> Creating resources with label app=3scale-gateway ...
deploymentconfig "apicast" created
service "apicast" created
--> Success
Run 'oc status' to view your app.
```

## 4.4. 通过 OPENSIFT 控制台创建路由

1. 在浏览器中打开 OpenShift 集群的 Web 控制台：<https://OPENSIFT-SERVER-IP:8443/console/>  
如果在远程服务器上启动 OpenShift 集群，请使用 `--public-hostname` 中指定的值，而不是 `OPENSIFT-SERVER-IP`。

您应该会看到登录屏幕：

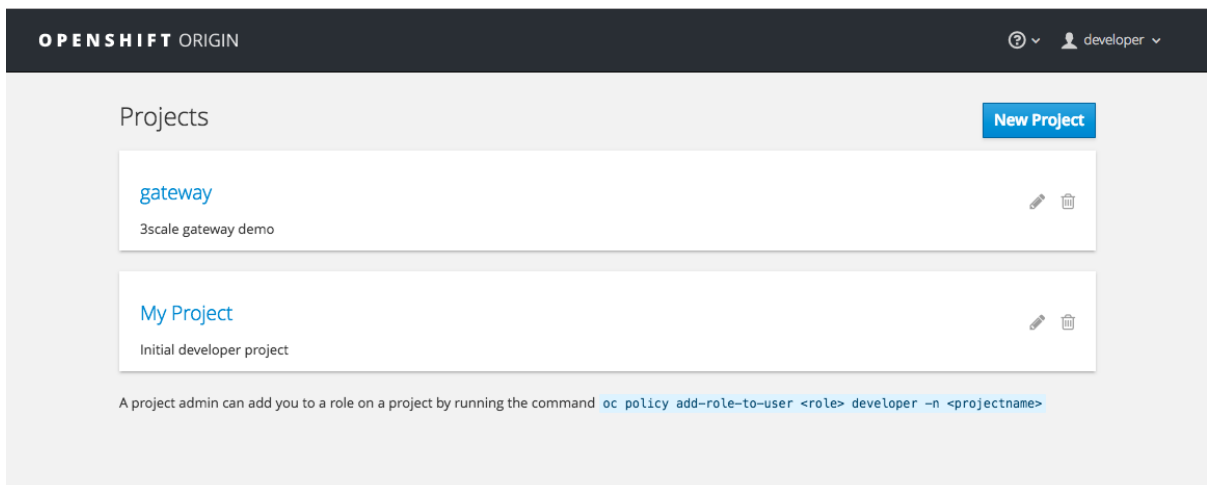




### 注意

您可能会收到有关不受信任的 Web 站点的警告。这是正常的，因为您要尝试通过安全协议访问 Web 控制台，而无需配置有效证书。虽然您应该在生产环境中避免这种情况，但对于此测试设置，您可以继续操作并为此地址创建一个例外。

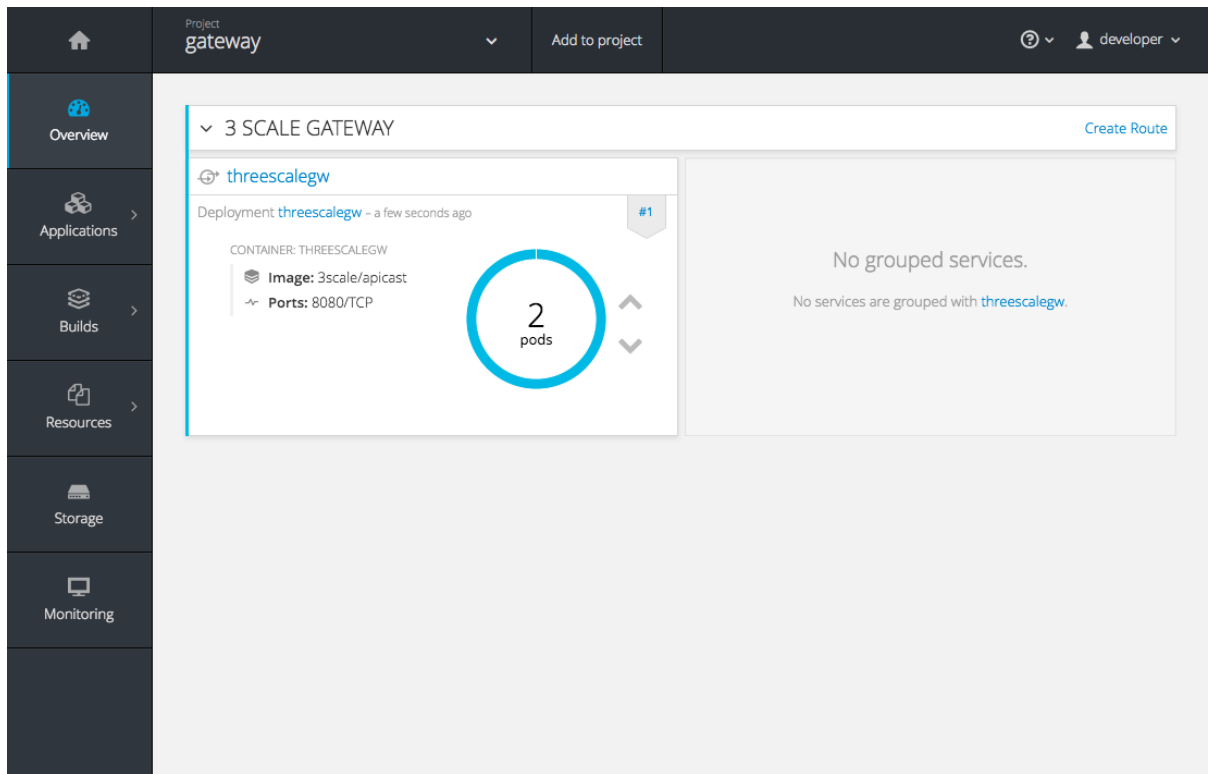
2. 使用在上面的 *Setup OpenShift* 部分中创建或获取的 *开发人员凭据* 登录。您将看到项目列表，包括上面通过命令行创建的 *gateway* 项目。



如果没有看到您的网关项目，您可能用其他用户创建它，并且需要将策略角色分配给此用户。

3. 点 *gateway* 链接，您将看到 *概述* 选项卡。OpenShift 下载了 APICast 的代码，并启动了部署。您可能在部署进行时看到消息 *Deployment #1 running*。

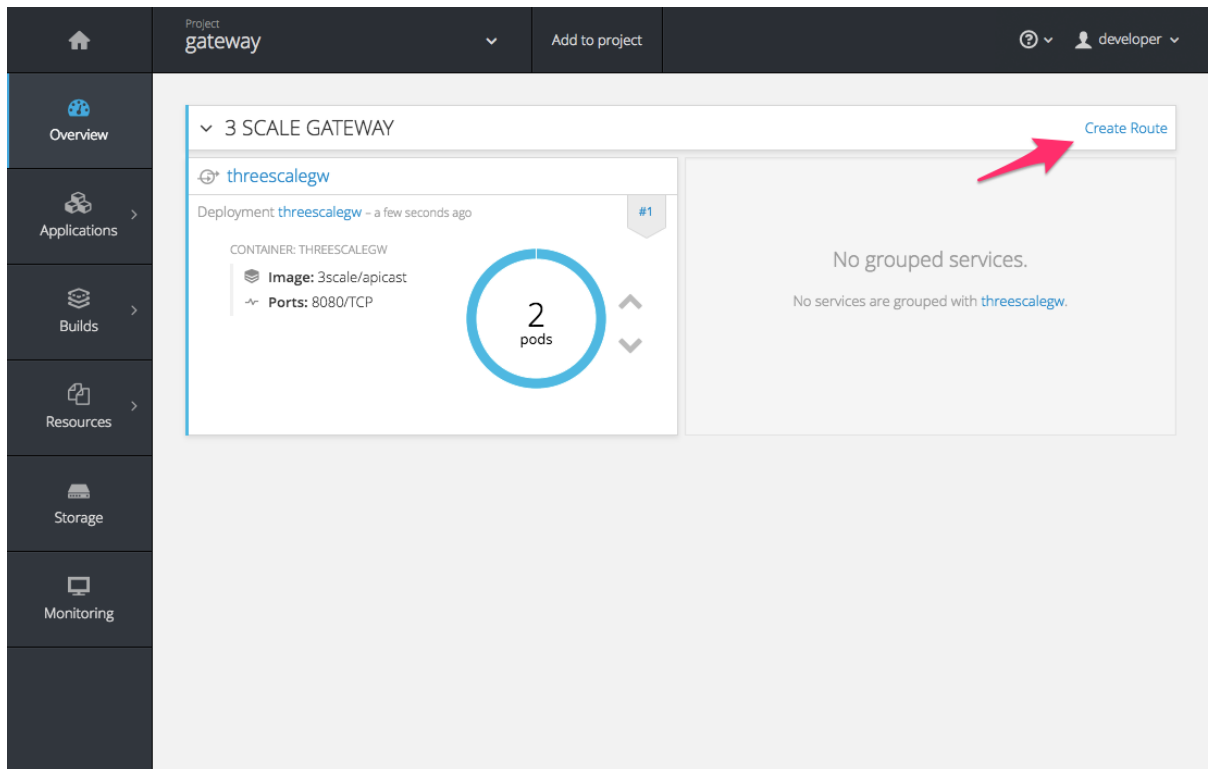
构建完成后，用户界面将刷新和显示 OpenShift 启动的两个 APICast (2 *pods*) 实例，如模板中所定义。



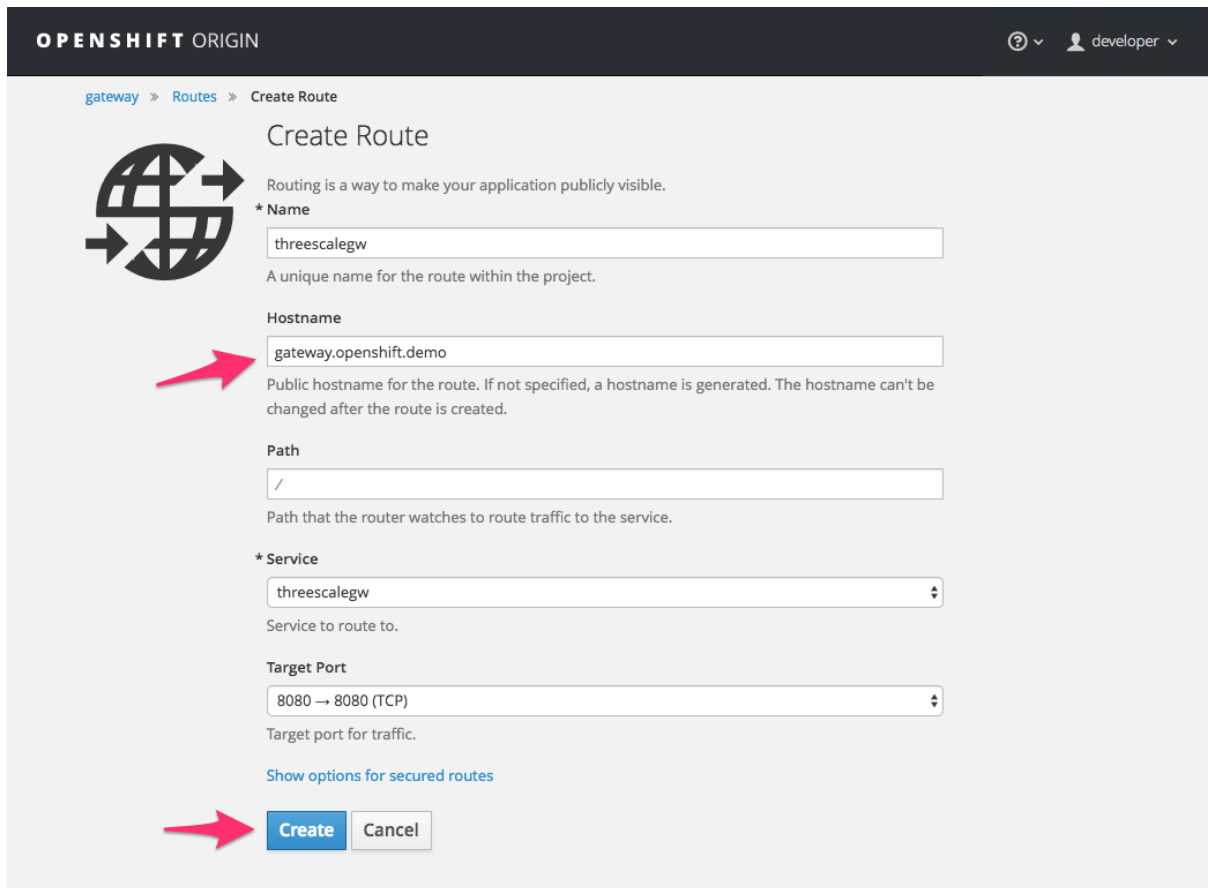
在启动时，每个 APIcast 实例都会使用 3scale 管理门户的 **Integration** 页面中提供的设置从 3scale 下载所需的配置。

OpenShift 将维护两个 APIcast 实例，并监控这两者的健康状态；任何不健康的 APIcast 实例将自动替换为新的实例。

4. 为了允许您的 APIcast 实例接收流量，您需要创建一个路由。首先点 **Create Route**。



在 **Public Base URL** 部分（不带 `http://` 且不带端口）部分输入您在 3scale 中设置的同一主机，如 `gateway.openshift.demo`，然后点 **Create** 按钮。



**OPENSIFT ORIGIN** developer

gateway » Routes » Create Route

## Create Route

Routing is a way to make your application publicly visible.

**\* Name**  
threescalegw  
A unique name for the route within the project.

**Hostname**  
gateway.openshift.demo  
Public hostname for the route. If not specified, a hostname is generated. The hostname can't be changed after the route is created.

**Path**  
/  
Path that the router watches to route traffic to the service.

**\* Service**  
threescalegw  
Service to route to.

**Target Port**  
8080 → 8080 (TCP)  
Target port for traffic.

[Show options for secured routes](#)

**Create** **Cancel**

对于您定义的每个 3scale 服务，您必须创建一个新路由。或者，您可以配置 [嵌入式 APICast 通配符路由](#)，以避免为您定义的每个 3scale 服务创建新路由。

## 第 5 章 DOCKER 容器化环境中的 APICAST

这是一个分步骤指南，可在 Docker 格式的容器中部署 APICast，它可以用作 3scale API 网关。

### 5.1. 先决条件

您必须根据 [安装 APICast 指南](#) 在 3scale 管理门户中配置 APICast。

### 5.2. 安装 DOCKER 容器化环境

本指南涵盖了在 Red Hat Enterprise Linux (RHEL) 7 中设置 Docker 容器化环境的步骤。

红帽提供的 Docker 格式容器作为 RHEL 中的 Extras 频道的一部分发布。要启用附加软件仓库，您可以使用 [Subscription Manager](#) 或 yum 配置管理器。详情请查看 [RHEL 产品文档](#)。

要在 AWS EC2 实例上部署 RHEL 7，请执行以下步骤：

1. 列出所有存储库：**sudo yum repolist all**.
2. 查找 **\*-extras** 存储库。
3. 启用 **extras** 存储库：**sudo yum-config-manager --enable rhui-REGION-rhel-server-extras**.
4. 安装 Docker 容器化环境软件包：**sudo yum install docker**。

对于其他操作系统，请参阅以下 Docker 文档：

- [在 Linux 发行版上安装 Docker 容器化环境](#)
- [在 Mac 上安装 Docker 容器化环境](#)
- [在 Windows 上安装 Docker 容器化环境](#)

### 5.3. 运行 DOCKER 容器化环境网关

1. 启动 Docker 守护进程：  
**sudo systemctl start docker.service**。
2. 检查 Docker 守护进程是否正在运行：  
**sudo systemctl status docker.service**。

您可以下载 Red Hat registry 中的 Docker 格式容器镜像：

```
sudo docker pull registry.access.redhat.com/3scale-amp25/apicast-gateway
```

3. 在 Docker 格式的容器中运行 APICast：  
**sudo docker run --name apicast --rm -p 8080:8080 -e THREESCALE\_PORTAL\_ENDPOINT=https://<access\_token>@<domain>-admin.3scale.net registry.access.redhat.com/3scale-amp25/apicast-gateway**。

在这里，**<access\_token>** 是 3scale 帐户管理 API 的访问令牌。您可以使用 Provider Key 而不是访问令牌。**<domain>-admin.3scale.net** 是 3scale 管理门户的 URL。

此命令在端口 **8080** 上运行名为 *apicast* 的 Docker 格式容器，并从 3scale 门户获取 JSON 配置文件。有关其他配置选项，请参阅 [安装 APICast 指南](#)。

### 5.3.1. Docker 命令选项

您可以在 `docker run` 命令中使用以下选项：

- `--rm`：在容器退出时自动移除容器。
- `-d` 或 `--detach`：在后台运行容器，并打印容器 ID。如果未指定，容器将以前台模式运行，您可以使用 `CTRL + c` 来停止容器。以分离模式启动时，您可以使用 `docker attach` 命令重新连接到容器，例如 `docker attach apicast`。
- `-p` 或 `--publish`：向主机发布容器的端口。该值的格式应为 `<host port="">:<container port="">`，因此 `-p 80:8080` 将容器的端口 `8080` 绑定到主机计算机的端口 `80`。例如，管理 API 使用端口 `8090`，因此您可能希望通过在 `docker run` 命令中添加 `-p 8090:8090` 来发布此端口。
- `-e` 或 `--env`：设置环境变量。
- `-v` 或 `--volume`：挂载卷。该值通常以 `<host path="">:<container path="">[:<options>]` 表示。`<options>` 是一个可选属性；您可以将其设置为 `:ro` 以指定该卷将仅读取（默认情况下，它以读写模式挂载）。示例：`-v /host/path:/container/path:ro`。

有关可用选项的更多信息，请参阅 [Docker 运行参考](#)。

## 5.4. 测试 APICAST

前面的步骤确保 Docker 格式的容器使用您自己的配置文件和 3scale registry 中的 Docker 格式镜像运行。您可以通过 APICast 在端口 `8080` 上测试调用，并提供正确的身份验证凭据，您可以从 3scale 帐户获得这些凭据。

测试调用不仅验证 APICast 是否在正确运行，还验证身份验证和报告是否得到成功处理。



### 注意

确保您用于调用的主机与 [Integration](#) 页面的 `Public Base URL` 字段中配置的主机相同。

## 第 6 章 安装 3SCALE OPERATOR

### 6.1. 简介

在本文档中，您将找到有关部署自定义资源定义 (CRD)、部署基于角色的访问控制 (RBAC) 以及部署 3scale Operator 镜像的详细信息，以便在多个环境中管理同一配置部署。



#### 重要

3scale Operator 只是一个技术预览功能。技术预览功能不受红帽产品服务等级协议 (SLA) 支持，且功能可能并不完整。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。有关红帽技术预览功能支持范围的更多信息，请参阅[技术预览功能支持范围](#)。

#### 6.1.1. 先决条件

- 3scale 2.5 [On-Premises](#) 实例
- OpenShift Container Platform 3.11
  - 在 OpenShift 集群中具有管理员特权的用户帐户



#### 警告

在单独的 *项目* 中部署 3scale operator 和自定义 CRD。如果您将它们部署到包含基础架构的现有 *项目* 中，则可能会更改或删除现有的元素。

### 6.2. 创建新的 OPENSIFT 项目

您必须运行类似以下内容的命令，以创建新的 OpenShift 项目。此命令显示项目名称示例：

```
oc new-project operator-test
```

这会创建新的 *OpenShift 项目*，其中将安装 operator、*APIManager* 自定义资源(CR)和 *Capabilities* 自定义资源。

#### 6.2.1. 在 OpenShift API 服务器中注册 3scale operator CRD

1. 下载并解压缩 tar 文件：[deploy.tar.gz](#)。
2. 以集群管理员身份登录并运行以下命令来部署所有 3scale operator CRD：

```
for i in `ls deploy/crds/*_crd.yaml`; do oc create -f $i ; done
```

- a. 这将注册 *APIManager* CRD，以及与在 *OpenShift API 服务器* 中的 Operator 的 *Capabilities* 功能相关的 CRD。
- b. 如果命令成功运行，您应能够通过 `oc get` 查询此 CRD 定义的资源类型。

- i. 例如，要验证 *APIManager* CRD 是否已正确注册，请执行以下命令：

```
oc get apimanagers
```

3. 您应该看到以下输出：

```
No resources found.
```

## 6.2.2. 为 3scale Operator 部署角色和服务账户

1. 进入 **operator-test** *OpenShift* 项目，并确保不存在其他元素：

```
export NAMESPACE="operator-test"
oc project ${NAMESPACE}
oc get all // This shouldn't return any result
```

2. 部署 3scale Operator 要使用的 *服务账户*：

```
oc create -f deploy/service_account.yaml
```

3. 以集群管理员身份登录到集群，并部署 3scale operator *Role* 以及将该角色附加到所创建的 *ServiceAccount* 的 *RoleBinding*：

```
// As a cluster admin
export NAMESPACE="operator-test"
oc project ${NAMESPACE}
oc create -f deploy/role.yaml
oc create -f deploy/role_binding.yaml
```

## 6.2.3. 部署 3scale Operator

1. 执行以下命令部署 3scale Operator：

```
export NAMESPACE="operator-test"
oc project ${NAMESPACE}
oc create -f deploy/operator.yaml
```

- a. 这将创建一个 *Deployment*，它将包含带有 *Operator* 代码的 *Pod*，并将开始侦听传入的 *APIManager* 和 *Capabilities* 资源。
2. 要验证 3scale Operator 是否已部署并就绪，请运行以下命令：

```
oc get deployment 3scale-operator
```

## 第 7 章 3SCALE 高可用性和评估模板

### 7.1. 简介

本文档描述了 Red Hat 3scale API Management 2.5 On-Premises 安装所使用的高可用性和评估模板。



#### 重要

3scale 高可用性和评估模板只是一个技术预览功能。技术预览功能不受红帽产品服务等级协议 (SLA) 支持，且功能可能并不完整。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。有关红帽技术预览功能支持范围的更多信息，请参阅[技术预览功能支持范围](#)。

### 7.2. 先决条件

- 您需要有一个可用的 OpenShift 集群来部署高可用性和评估模板的元素。

### 7.3. 高可用性模板

High Availability(HA)模板允许您为关键数据库设置 HA。

#### 7.3.1. 先决条件

- 在部署 HA 模板之前，您必须部署和配置外部数据库，并使用负载均衡端点在 HA 配置中配置它们。

#### 7.3.2. 使用 HA 模板

对于 HA，名为 **amp-ha-tech-preview.yml** 的模板允许您将关键数据库部署到 OpenShift 外部。这包括：

- Memcached
- Sphinx
- Zync

标准 **amp.yml** 模板和 **amp-ha-tech-preview.yml** 之间的区别包括：

- 删除以下元素：
  - backend-redis 及其相关组件
  - system-redis 及其相关组件
  - system-mysql 及其相关组件
  - redis 和 MySQL 相关 ConfigMap
  - MYSQL\_IMAGE, REDIS\_IMAGE, MYSQL\_USER, MYSQL\_ROOT\_PASSWORD 参数
- 默认情况下，将非数据库 **DeploymentConfig** 对象类型的副本数从 1 增加到 2。
- 添加以下强制参数，以便您控制外部数据库的位置：



- BACKEND\_REDIS\_STORAGE\_ENDPOINT
- BACKEND\_REDIS\_QUEUES\_ENDPOINT
- SYSTEM\_REDIS\_URL
- APICAST\_STAGING\_REDIS\_URL
- APICAST\_PRODUCTION\_REDIS\_URL
- SYSTEM\_DATABASE\_URL

使用 **amp-ha-tech-preview.yml**，您需要通过新增的强制参数从集群中配置数据库连接（不包括 **system-memcache**、**zync-database** 和 **system-sphinx**）。端点需要数据库负载均衡的连接字符串，包括身份验证信息。另外，对于非数据库部署，pod 副本数默认增加到 2，以在应用级别具有冗余性。

## 7.4. 评估模板

出于评估目的，有一个名为 **amp-eval-tech-preview.yml** 的模板，模板可在无资源请求或限制的情况下部署 3scale 环境。

与 standard **amp.yml** 模板相比，唯一功能区别在于已删除资源限制和请求。这意味着，在此版本中，CPU 和内存级别的 pod 上已删除了最低硬件要求。此模板仅用于评估、测试和开发目的，因为它尝试使用给定的硬件资源以最佳的方式部署组件。

## 第 8 章 对 3SCALE 的 REDIS 高可用性(HA)支持



### 注意

对 3scale 的 Redis 高可用性 (HA) 支持存在一个已知问题。如需更多信息，请参阅 Red Hat 3scale API Management 2.5 发行注记，第 8 章。发行注记中的[已知问题](#)。

### 8.1. 简介

OpenShift 容器平台(OCP)为大多数组件提供高可用性(HA)。如需更多信息，请参阅 [OpenShift Container Platform 3.11 第 30 章。高可用性](#)。

3scale 中的 HA 数据库组件包括：

- **system-redis**：为 3scale 的后台作业提供临时存储，也用作 **system-app** pod 的 *Ruby* 处理的消息总线。
- **backend-redis**：用于统计存储和临时作业存储。



### 注意

**system-redis** 和 **backend-redis** 都可以被 *Redis 集群* (open-source 或 Redis Labs) 替代。

以下 **env vars** 可以被包括在 **system-(app,sidekiq,sphinx)** 部署配置中，虽然只有 **Redis Enterprise** 需要这样做：

- **MESSAGE\_BUS\_REDIS\_URL**（一个 redis URL）
- **REDIS\_NAMESPACE**（命名空间 Sidekiq 的 Redis 密钥的简短字符串）
- **MESSAGE\_BUS\_REDIS\_NAMESPACE**（命名空间系统消息总线 Redis 密钥的简短字符串）

当 Redis pod 结束或被 OCP 终止时，会自动创建一个新的 pod，数据会从持久性存储中恢复，以便 pod 可以继续运行。在描述的场景中，新 pod 启动时会出现少量停机时间。这是因为 Redis 有不支持多主设置的限制。通过将 Redis 镜像加载到部署 Redis 的所有节点上，可以缩短停机时间，从而加快 pod 重启。

### 8.2. 为零停机时间设置 REDIS

如果需要零停机时间，则需要 OCP 外部设置 Redis。使用 3scale pod 的配置选项进行设置的方法有几种：

- 设置您自己的自助管理的 Redis
- 使用 Redis Sentinel：[参考 Redis Sentinel 文档](#)
- redis 作为服务提供：
  - Amazon ElastiCache
  - redis Labs



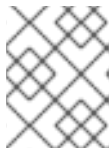
### 注意

红帽不支持上述服务。提及任何此类服务并不意味着红帽认可这些产品或服务。您同意，由于您使用（或依赖）任何外部内容而可能导致的任何损失或费用，红帽不承担任何责任。

## 8.3. 为 3SCALE 配置后端组件

3scale 2.5 中有配置设置来为后端组件配置 Redis HA（故障切换）。您可以在以下部署配置中将其设置为环境变量：**backend-cron**、**backend-listener** 和 **backend-worker**：

- **CONFIG\_REDIS\_SENTINEL\_HOSTS** 和 **CONFIG\_QUEUES\_SENTINEL\_HOSTS**:  
用于主统计数据数据库和 Resque 后台作业数据库的 Sentinel 主机的逗号分隔列表。



### 注意

值的格式应为：`name:value <host>:<port>`。例如：`host1:26379, host2:26379`, 或 `host3:26379`

- **CONFIG\_REDIS\_SENTINEL\_ROLE** 和 **CONFIG\_QUEUES\_SENTINEL\_ROLE**:  
每个 Sentinels 组的角色，可以是 *master* 或 *slave*。目前只支持 *master*（默认）。

这使得 **CONFIG\_REDIS\_PROXY** 和 **CONFIG\_QUEUES\_MASTER\_NAME** 的值采用 Sentinel 组名称而非特定服务器的含义。

- 当没有配置 Sentinel 主机时，环境变量 **CONFIG\_REDIS\_PROXY** 和 **CONFIG\_QUEUES\_MASTER\_NAME** 可以使用 URL 并支持密码保护的数据库，例如：**CONFIG\_REDIS\_PROXY=redis://user:password@server:port/database**
  - 然后，使用密码保护的 Redis 实例建立连接。
- 配置 Sentinel 主机后，必须在 Sentinel 配置中设置密码，并使用 Sentinel 组名称：**CONFIG\_REDIS\_PROXY=master\_group**