



Red Hat 3scale API Management 2.4

インフラストラクチャー

Red Hat 3scale API Management を異なるプラットフォームにデプロイする方法について参照してください。

Red Hat 3scale API Management 2.4 インフラストラクチャー

Red Hat 3scale API Management を異なるプラットフォームにデプロイする方法について参照してください。

法律上の通知

Copyright © 2023 Red Hat, Inc.

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[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

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

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] 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[®] 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.

概要

本書では、Red Hat 3scale API Management 2.4 を使用したデプロイメントおよびインフラストラクチャーの管理について説明します。

目次

第1章 3SCALE API MANAGEMENT 2.3 から 2.4 へのアップグレード	3
1.1. 前提条件	3
1.2. 3SCALE API MANAGEMENT のアップグレード	3
1.3. 管理者 IMPERSONATION の変更 (オプション)	22
第2章 ORACLE DATABASE リレーショナルデータベース管理システムを使用した 3SCALE API MANAGEMENT システムイメージのビルド	24
2.1. 作業を開始する前に	24
2.2. ORACLE DATABASE の準備	24
2.3. システムイメージのビルド	25
第3章 オンプレミス型 3SCALE API MANAGEMENT インストールガイド	27
3.1. 前提条件	27
3.2. 3SCALE OPENSIFT テンプレート	27
3.3. システム要件	27
3.4. ノードおよびエンタイトルメントの設定	28
3.5. テンプレートを使用した OPENSIFT への 3SCALE のデプロイ	28
3.6. 3SCALE テンプレートパラメーター	31
3.7. OPENSIFT 上で 3SCALE と APICAST を使用する	35
3.8. 7.トラブルシューティング	38
第4章 オンプレミス型 3SCALE API MANAGEMENT の操作およびスケーリングガイド	42
4.1. はじめに	42
4.2. APICAST の再デプロイ	42
4.3. APICAST 組み込みワイルドカードルーティング	43
4.4. オンプレミス型 AMP のスケールアップ	43
4.5. 操作のトラブルシューティング	46
第5章 3SCALE API MANAGEMENT HIGH AVAILABILITY AND EVALUATION	47
5.1. はじめに	47
5.2. 前提条件	47
5.3. 高可用性テンプレート	47
5.4. 評価用テンプレート	48
第6章 3SCALE の REDIS 高可用性 (HA) サポート	49
6.1. はじめに	49
6.2. ゼロダウンタイムのための REDIS 設定	49
6.3. 3SCALE 用バックエンドコンポーネントの設定	50
第7章 「HOW TO DEPLOY A FULL-STACK API SOLUTION WITH FUSE, 3SCALE, AND OPENSIFT」 ...	51
7.1. パート 1: FUSE ON OPENSIFT の設定	52
7.2. パート 2: 3SCALE API MANAGEMENT の設定	59
7.3. パート 3: API サービスの統合	62
7.4. パート 4: API および API MANAGEMENT のテスト	62

第1章 3SCALE API MANAGEMENT 2.3 から 2.4 へのアップグレード

3scale API Management 管理者は、インストールをバージョン 2.3 から 2.4 にアップグレードします。オプションで、[管理者データの偽装を変更](#)できます。



警告

このプロセスにより、サービスで中断が生じる可能性があります。メンテナンス期間があることを確認してください。

1.1. 前提条件

- プロジェクトにデプロイされた 3scale API Management 2.3
- ツールの前提条件
 - jq

1.2. 3SCALE API MANAGEMENT のアップグレード

3scale API Management を 2.3 から 2.4 にアップグレードするには、以下の手順にしたがいます。

1. [「ConfigMap の作成」](#)
2. [「システムのマスタールートの作成」](#)
3. [「システムデータベースシークレット の移行」](#)
4. [「-redis シークレットの作成」](#)
5. [「DeploymentConfig のパッチ適用」](#)
6. [「パッチイメージストリーム」](#)
7. [「サービスディスカバリーのエントリーの追加」](#)
8. [「追加の DeploymentConfig の設定」](#)

1.2.1. ConfigMap の作成

新規 OpenShift 要素についてソースされる ConfigMap の詳細が含まれるファイルを作成します。

1. 必要な変数を設定します。

```
export $(oc set env dc/system-app --list|grep THREESCALE_SUPERDOMAIN|sort -u)
export $(oc set env dc/system-app --list|grep APICAST_REGISTRY_URL|sort -u)
APP_LABEL=$(oc get dc backend-listener -o json | jq .spec.template.metadata.labels.app -r)
AMP_RELEASE=2.4.0
```

- 変数が正しく設定されていることを確認します。

```
echo $THREESCALE_SUPERDOMAIN
echo $APICAST_REGISTRY_URL
echo $APP_LABEL
echo $AMP_RELEASE
```

- system-environment** ConfigMap を作成します。

- system-environment.yml** という名前のファイルを作成します。

```
cat<<EOF> system-environment.yml

apiVersion: v1
data:
  AMP_RELEASE: ${AMP_RELEASE}
  APICAST_REGISTRY_URL: ${APICAST_REGISTRY_URL}
  FORCE_SSL: "true"
  PROVIDER_PLAN: enterprise
  RAILS_ENV: production
  RAILS_LOG_LEVEL: info
  RAILS_LOG_TO_STDOUT: "true"
  SSL_CERT_DIR: /etc/pki/tls/certs
  THINKING_SPHINX_PORT: "9306"
  THREESCALE_SANDBOX_PROXY_OPENSSL_VERIFY_MODE: VERIFY_NONE
  THREESCALE_SUPERDOMAIN: ${THREESCALE_SUPERDOMAIN}
kind: ConfigMap
metadata:
  creationTimestamp: null
  labels:
    app: ${APP_LABEL}
    3scale.component: system
  name: system-environment
EOF
```

- 新しい **system-environment** ConfigMap を作成します。

```
oc create -f system-environment.yml
```

- backend-environment** ConfigMap を作成します。

- backend-listener** DeploymentConfig の値を取得します。

```
export $(oc set env dc/backend-listener --list|grep RACK_ENV|sort -u)
```

- backend-environment.yml** という名前の新規ファイルを作成します。

```
cat<<EOF>backend-environment.yml
apiVersion: v1
data:
  RACK_ENV: ${RACK_ENV}
kind: ConfigMap
metadata:
  creationTimestamp: null
  labels:
```

```

app: ${APP_LABEL}
3scale.component: backend
name: backend-environment
EOF

```

- c. 新しい **backend-environment** ConfigMap を作成します。

```
oc create -f backend-environment.yml
```

5. **apicast-environment** ConfigMap を作成します。

- a. **apicast-environment.yml** という名前の新規ファイルを作成します。

```

cat<<EOT>apicast-environment.yml
apiVersion: v1
data:
  APICAST_MANAGEMENT_API: status
  APICAST_RESPONSE_CODES: "true"
  OPENSSE_VERIFY: "false"
kind: ConfigMap
metadata:
  name: apicast-environment
EOT

```

- b. 新しい **apicast-environment** ConfigMap を作成します。

```
oc create -f apicast-environment.yml
```

1.2.2. システムのマスタールートの作成

システムのマスタールートを作成するには、以下を実行します。

1. **MASTER_NAME** を **MASTER_DOMAIN** 環境変数の値で設定します。

```

export $(oc set env dc/system-app --list|grep MASTER_DOMAIN|sort -u)
MASTER_NAME=$MASTER_DOMAIN

```

2. **MASTER_NAME** が正しく設定されていることを確認します。

```
echo $MASTER_NAME
```

3. **system-master** ルートを作成し、**system-master-admin** ルートを削除します。

```

oc create route edge system-master --service=system-master --
hostname=${MASTER_NAME}.${THREESCALE_SUPERDOMAIN} --port=http
oc delete route system-master-admin

```

1.2.3. システムデータベースシークレット の移行

システムデータベースのシークレット を新規 **system -database** OpenShift シークレットに移行するには、以下を実行します。

1. 既存の MySQL 環境変数を取得します。

```
export $(oc set env dc/system-mysql --list|grep MYSQL_ROOT_PASSWORD)
export $(oc set env dc/system-mysql --list | grep MYSQL_DATABASE)
```

2. APP_LABEL 環境変数を取得します。

```
APP_LABEL=$(oc get dc backend-listener -o json | jq .spec.template.metadata.labels.app -r)
```

3. 以下の環境変数を正しく設定していることを確認します。

```
echo ${MYSQL_ROOT_PASSWORD}
echo ${MYSQL_DATABASE}
echo ${APP_LABEL}
```

4. **system-database** シークレットを含むファイルを作成します。

```
cat > system-database.yml <<EOF

apiVersion: v1
kind: Secret
metadata:
  creationTimestamp: null
  labels:
    3scale.component: system
    app: ${APP_LABEL}
  name: system-database
stringData:
  URL: mysql2://root:${MYSQL_ROOT_PASSWORD}@system-
mysql/${MYSQL_DATABASE}
  type: Opaque
EOF
```

5. シークレットを作成します。

```
oc create -f system-database.yml
```

1.2.4. 新規シークレットの作成

アップグレードプロセスを続行するには、以下を作成する必要があります。

- **system-master-apicast** シークレット
- 複数の **-redis** シークレット
- **複数のバックエンドシークレット**
- **複数のシステムシークレット**

1.2.4.1. system-master-apicast シークレットの作成

1. APICAST_ACCESS_TOKEN の値を取得します。

```
export APICAST_ACCESS_TOKEN=$(oc set env --list dc/apicast-production|sort -u|grep
THREESCALE_PORTAL_ENDPOINT|cut -d@ -f1|cut -d/ -f3)
```

2. **system-master-apicast.yml** という名前の新規ファイルを作成します。

```
cat<<EOF> system-master-apicast.yml
apiVersion: v1
kind: Secret
metadata:
  creationTimestamp: null
  labels:
    app: ${APP_LABEL}
    3scale.component: system
  name: system-master-apicast
stringData:
  ACCESS_TOKEN: ${APICAST_ACCESS_TOKEN}
  BASE_URL: http://${APICAST_ACCESS_TOKEN}@system-master:3000
  PROXY_CONFIGS_ENDPOINT: http://${APICAST_ACCESS_TOKEN}@system-
master:3000/master/api/proxy/configs
type: Opaque
EOF
```

3. 新しい **system-master-apicast** シークレットを作成します。

```
oc create -f system-master-apicast.yml
```

[「新規シークレットの作成」](#) に戻る

1.2.4.2. -redis シークレットの作成

1. **system-redis** シークレットを作成します。

- a. 以下の内容で、**system-redis.yml** という名前のファイルを作成します。

```
cat > system-redis.yml <<EOF

apiVersion: v1
kind: Secret
metadata:
  creationTimestamp: null
  labels:
    3scale.component: system
    app: ${APP_LABEL}
  name: system-redis
stringData:
  URL: redis://system-redis:6379/1
  type: Opaque
EOF
```

- b. ファイルに含まれる情報を使用して **system-redis** シークレットを作成します。

```
oc create -f system-redis.yml
```

2. **backend-redis** シークレットを作成します。

- a. 以下の内容で、**backend-redis.yml** という名前のファイルを作成します。

```

cat > backend-redis.yml <<EOF

apiVersion: v1
kind: Secret
type: Opaque
metadata:
  creationTimestamp: null
  labels:
    3scale.component: backend
    app: ${APP_LABEL}
  name: backend-redis
stringData:
  REDIS_QUEUES_SENTINEL_HOSTS: ""
  REDIS_QUEUES_SENTINEL_ROLE: ""
  REDIS_QUEUES_URL: redis://backend-redis:6379/1
  REDIS_STORAGE_SENTINEL_HOSTS: ""
  REDIS_STORAGE_SENTINEL_ROLE: ""
  REDIS_STORAGE_URL: redis://backend-redis:6379/0
EOF

```

- b. ファイルに含まれる情報を使用して、**backend-redis** シークレットを作成します。

```
oc create -f backend-redis.yml
```

3. **apicast-redis** シークレットを作成します。

- a. 以下の内容で **apicast-redis.yml** という名前のファイルを作成します。

```

cat<<EOT>apicast-redis.yml
apiVersion: v1
stringData:
  PRODUCTION_URL: redis://system-redis:6379/1
  STAGING_URL: redis://system-redis:6379/2
kind: Secret
metadata:
  labels:
    3scale.component: apicast
    app: ${APP_LABEL}
  name: apicast-redis
EOT

```

- b. ファイルに含まれる情報を使用して **apicast-redis** シークレットを作成します。

```
oc create -f apicast-redis.yml
```

[「新規シークレットの作成」](#) に戻る

1.2.4.3. **backend-** シークレットの作成

1. **backend-listener** シークレットを作成します。

- a. 以下の内容で、**backend-listener.yml** という名前のファイルを作成します。

```
cat<<EOT>backend-listener.yml
```

```

apiVersion: v1
stringData:
  route_endpoint: https://backend-3scale.${THREESCALE_SUPERDOMAIN}
  service_endpoint: http://backend-listener:3000
kind: Secret
metadata:
  name: backend-listener
EOT

```

- b. ファイルに含まれる情報を使用して、**backend-listener** シークレットを作成します。

```
oc create -f backend-listener.yml
```

2. **backend-internal-api** シークレットを作成します。

- a. 2.3 環境から値を取得します。

```

export $(oc set env dc backend-listener --list|grep CONFIG_INTERNAL_API_USER)
export $(oc set env dc backend-listener --list|grep
CONFIG_INTERNAL_API_PASSWORD)

```

- b. 変数の値を確認します。

```
echo $CONFIG_INTERNAL_API_USER $CONFIG_INTERNAL_API_PASSWORD
```

- c. シークレットを作成します。

```

oc create secret generic backend-internal-api --from-
literal=password=${CONFIG_INTERNAL_API_PASSWORD} --from-
literal=username=${CONFIG_INTERNAL_API_USER}

```

[「新規シークレットの作成」](#) に戻る

1.2.4.4. **system-** シークレットの作成

1. **system-memcache** シークレットを作成します。

- a. 以下の内容で、**system-memcache.yml** という名前のファイルを作成します。

```

cat<<EOT>system-memcache.yml
apiVersion: v1
stringData:
  SERVERS: system-memcache:11211
kind: Secret
metadata:
  name: system-memcache
EOT

```

- b. ファイルに含まれる情報を使用して **system-memcache** シークレットを作成します。

```
oc create -f system-memcache.yml
```

2. **system-recaptcha** シークレットを作成します。

- a. 以下の内容で、**system-recaptcha.yml** という名前のファイルを作成します。

```
cat<<EOT>system-recaptcha.yml
apiVersion: v1
stringData:
  PRIVATE_KEY: ""
  PUBLIC_KEY: ""
kind: Secret
metadata:
  name: system-recaptcha
EOT
```

- b. ファイルに含まれる情報を使用して **system-recaptcha** シークレットを作成します。

```
oc create -f system-recaptcha.yml
```

3. **system-events-hook** シークレットを作成します。

- a. 2.3 環境から値を取得します。

```
export $(oc set env dc backend-worker --list|grep
CONFIG_EVENTS_HOOK_SHARED_SECRET)
```

- b. 変数の値を確認します。

```
echo ${CONFIG_EVENTS_HOOK_SHARED_SECRET}
```

- c. シークレットを作成します。

```
oc create secret generic system-events-hook --from-
literal=PASSWORD=${CONFIG_EVENTS_HOOK_SHARED_SECRET} --from-
literal=URL=http://system-master:3000/master/events/import
```

4. **system-seed** シークレットを作成します。

- a. 2.3 環境から値を取得します。

```
export $(oc set env dc system-app --list|grep ADMIN_ACCESS_TOKEN|uniq)
export $(oc set env dc system-app --list|grep MASTER_ACCESS_TOKEN|uniq)
export $(oc set env dc system-app --list|grep USER_PASSWORD|uniq)
export $(oc set env dc system-app --list|grep USER_LOGIN|uniq)
export $(oc set env dc system-app --list|grep MASTER_DOMAIN|uniq)
export $(oc set env dc system-app --list|grep MASTER_PASSWORD|uniq)
export $(oc set env dc system-app --list|grep MASTER_USER|uniq)
export $(oc set env dc system-app --list|grep TENANT_NAME|uniq)
```

- b. 変数の値を確認します。

```
echo ${ADMIN_ACCESS_TOKEN} ${MASTER_ACCESS_TOKEN}
${USER_PASSWORD} ${USER_LOGIN} ${MASTER_DOMAIN}
${MASTER_PASSWORD} ${MASTER_USER} ${TENANT_NAME}
```

- c. シークレットを作成します。

```
oc create secret generic system-seed --from-
literal=ADMIN_ACCESS_TOKEN=${ADMIN_ACCESS_TOKEN} --from-
literal=MASTER_ACCESS_TOKEN=${MASTER_ACCESS_TOKEN} --from-
literal=ADMIN_PASSWORD=${USER_PASSWORD} --from-
literal=ADMIN_USER=${USER_LOGIN} --from-
literal=MASTER_DOMAIN=${MASTER_DOMAIN} --from-
literal=MASTER_PASSWORD=${MASTER_PASSWORD} --from-
literal=MASTER_USER=${MASTER_USER} --from-
literal=TENANT_NAME=${TENANT_NAME}
```

5. **system-app** シークレットを作成します。

- a. 2.3 環境から値を取得します。

```
export $(oc set env dc system-app --list|grep SECRET_KEY_BASE|uniq)
```

- b. 変数の値を確認します。

```
echo ${SECRET_KEY_BASE}
```

- c. シークレットを作成します。

```
oc create secret generic system-app --from-
literal=SECRET_KEY_BASE=${SECRET_KEY_BASE}
```

[「新規シークレットの作成」](#) に戻る

1.2.5. DeploymentConfig のパッチ適用

1. **backend-cron** DeploymentConfig にパッチを適用します。

```
oc patch dc/backend-cron -p '{"spec":{"template":{"spec":{"containers":[{"name":"backend-
cron","env":[{"name":"CONFIG_REDIS_PROXY","value":null,"valueFrom":{"secretKeyRef":
{"key":"REDIS_STORAGE_URL","name":"backend-redis"}}],
{"name":"CONFIG_REDIS_SENTINEL_HOSTS","value":null,"valueFrom":{"secretKeyRef":
{"key":"REDIS_STORAGE_SENTINEL_HOSTS","name":"backend-redis"}},
{"name":"CONFIG_REDIS_SENTINEL_ROLE","value":null,"valueFrom":{"secretKeyRef":
{"key":"REDIS_STORAGE_SENTINEL_ROLE","name":"backend-redis"}},
{"name":"CONFIG_QUEUES_MASTER_NAME","value":null,"valueFrom":{"secretKeyRef":
{"key":"REDIS_QUEUES_URL","name":"backend-redis"}},
{"name":"CONFIG_QUEUES_SENTINEL_HOSTS","value":null,"valueFrom":{"secretKeyRef":
{"key":"REDIS_QUEUES_SENTINEL_HOSTS","name":"backend-redis"}},
{"name":"CONFIG_QUEUES_SENTINEL_ROLE","value":null,"valueFrom":{"secretKeyRef":
{"key":"REDIS_QUEUES_SENTINEL_ROLE","name":"backend-redis"}},
{"name":"RACK_ENV","value":null,"valueFrom":{"configMapKeyRef":
{"key":"RACK_ENV","name":"backend-environment"}]}]}]},"initContainers":[{"name":"backend-
redis-svc","command":["/opt/app/entrypoint.sh","sh","-c","until rake
connectivity:redis_storage_queue_check; do sleep $SLEEP_SECONDS;\ndone"],"env":
[{"name":"CONFIG_QUEUES_MASTER_NAME","value":null,"valueFrom":{"secretKeyRef":
{"key":"REDIS_QUEUES_URL","name":"backend-redis"}]}]}]}}'}
```

2. **backend-listener** DeploymentConfig にパッチを適用します。

```
oc patch dc/backend-listener -p '{"spec":{"template":{"spec":{"containers":[{"name":"backend-
```

```
listener", "env": [{"name": "CONFIG_REDIS_PROXY", "value": null, "valueFrom": {"secretKeyRef": {"key": "REDIS_STORAGE_URL", "name": "backend-redis"}}, {"name": "CONFIG_REDIS_SENTINEL_HOSTS", "value": null, "valueFrom": {"secretKeyRef": {"key": "REDIS_STORAGE_SENTINEL_HOSTS", "name": "backend-redis"}}, {"name": "CONFIG_REDIS_SENTINEL_ROLE", "value": null, "valueFrom": {"secretKeyRef": {"key": "REDIS_STORAGE_SENTINEL_ROLE", "name": "backend-redis"}}, {"name": "CONFIG_QUEUES_MASTER_NAME", "value": null, "valueFrom": {"secretKeyRef": {"key": "REDIS_QUEUES_URL", "name": "backend-redis"}}, {"name": "CONFIG_QUEUES_SENTINEL_HOSTS", "value": null, "valueFrom": {"secretKeyRef": {"key": "REDIS_QUEUES_SENTINEL_HOSTS", "name": "backend-redis"}}, {"name": "CONFIG_QUEUES_SENTINEL_ROLE", "value": null, "valueFrom": {"secretKeyRef": {"key": "REDIS_QUEUES_SENTINEL_ROLE", "name": "backend-redis"}}, {"name": "RACK_ENV", "value": null, "valueFrom": {"configMapKeyRef": {"key": "RACK_ENV", "name": "backend-environment"}}, {"name": "CONFIG_INTERNAL_API_USER", "value": null, "valueFrom": {"secretKeyRef": {"key": "username", "name": "backend-internal-api"}}, {"name": "CONFIG_INTERNAL_API_PASSWORD", "value": null, "valueFrom": {"secretKeyRef": {"key": "password", "name": "backend-internal-api"}}}]}]}'
```

3. **backend-worker** DeploymentConfig にパッチを適用します。

```
oc patch dc/backend-worker -p '{"spec":{"template":{"spec":{"containers":[{"name":"backend-worker", "env":[{"name":"CONFIG_REDIS_PROXY", "value": null, "valueFrom": {"secretKeyRef": {"key": "REDIS_STORAGE_URL", "name": "backend-redis"}}, {"name": "CONFIG_REDIS_SENTINEL_HOSTS", "value": null, "valueFrom": {"secretKeyRef": {"key": "REDIS_STORAGE_SENTINEL_HOSTS", "name": "backend-redis"}}, {"name": "CONFIG_REDIS_SENTINEL_ROLE", "value": null, "valueFrom": {"secretKeyRef": {"key": "REDIS_STORAGE_SENTINEL_ROLE", "name": "backend-redis"}}, {"name": "CONFIG_QUEUES_MASTER_NAME", "value": null, "valueFrom": {"secretKeyRef": {"key": "REDIS_QUEUES_URL", "name": "backend-redis"}}, {"name": "CONFIG_QUEUES_SENTINEL_HOSTS", "value": null, "valueFrom": {"secretKeyRef": {"key": "REDIS_QUEUES_SENTINEL_HOSTS", "name": "backend-redis"}}, {"name": "CONFIG_QUEUES_SENTINEL_ROLE", "value": null, "valueFrom": {"secretKeyRef": {"key": "REDIS_QUEUES_SENTINEL_ROLE", "name": "backend-redis"}}, {"name": "RACK_ENV", "value": null, "valueFrom": {"configMapKeyRef": {"key": "RACK_ENV", "name": "backend-environment"}}, {"name": "CONFIG_EVENTS_HOOK", "value": null, "valueFrom": {"secretKeyRef": {"key": "URL", "name": "system-events-hook"}}, {"name": "CONFIG_EVENTS_HOOK_SHARED_SECRET", "value": null, "valueFrom": {"secretKeyRef": {"key": "PASSWORD", "name": "system-events-hook"}}]}]}]}]}], "initContainers": [{"name": "backend-redis-svc", "command": ["/opt/app/entrypoint.sh", "sh", "-c", "until rake connectivity:redis_storage_queue_check; do sleep $SLEEP_SECONDS; \ndone"], "env": [{"name": "CONFIG_QUEUES_MASTER_NAME", "value": null, "valueFrom": {"secretKeyRef": {"key": "REDIS_QUEUES_URL", "name": "backend-redis"}}}]}]}'
```

4. **system-app** DeploymentConfig コンテナにパッチを適用します。

```
oc patch dc/system-app -p '{"spec":{"strategy":{"activeDeadlineSeconds":21600,"resources":{},"rollingParams":{"pre":{"execNewPod":{"env":[{"name":"AMP_RELEASE", "value": null, "valueFrom": {"configMapKeyRef": {"key": "AMP_RELEASE", "name": "system-environment"}}, {"name": "APICAST_REGISTRY_URL", "value": null, "valueFrom": {"configMapKeyRef": {"key": "APICAST_REGISTRY_URL", "name": "system-environment"}}, {"name": "FORCE_SSL", "value": null, "valueFrom": {"configMapKeyRef": {"key": "FORCE_SSL", "name": "system-environment"}}]}}}}}'
```

```

{"name":"PROVIDER_PLAN","value":null,"valueFrom":{"configMapKeyRef":
{"key":"PROVIDER_PLAN","name":"system-environment"}}},
{"name":"RAILS_ENV","value":null,"valueFrom":{"configMapKeyRef":
{"key":"RAILS_ENV","name":"system-environment"}}},
{"name":"RAILS_LOG_LEVEL","value":null,"valueFrom":{"configMapKeyRef":
{"key":"RAILS_LOG_LEVEL","name":"system-environment"}}},
{"name":"RAILS_LOG_TO_STDOUT","value":null,"valueFrom":{"configMapKeyRef":
{"key":"RAILS_LOG_TO_STDOUT","name":"system-environment"}}},
{"name":"SSL_CERT_DIR","value":null,"valueFrom":{"configMapKeyRef":
{"key":"SSL_CERT_DIR","name":"system-environment"}}},
{"name":"THINKING_SPHINX_PORT","value":null,"valueFrom":{"configMapKeyRef":
{"key":"THINKING_SPHINX_PORT","name":"system-environment"}}},
{"name":"THREESCALE_SANDBOX_PROXY_OPENSSL_VERIFY_MODE","value":null,"value
From":{"configMapKeyRef":
{"key":"THREESCALE_SANDBOX_PROXY_OPENSSL_VERIFY_MODE","name":"system-
environment"}}},{"name":"THREESCALE_SUPERDOMAIN","value":null,"valueFrom":
{"configMapKeyRef":{"key":"THREESCALE_SUPERDOMAIN","name":"system-
environment"}}},{"name":"DATABASE_URL","value":null,"valueFrom":{"secretKeyRef":
{"key":"URL","name":"system-database"}}},
{"name":"MASTER_DOMAIN","value":null,"valueFrom":{"secretKeyRef":
{"key":"MASTER_DOMAIN","name":"system-seed"}}},
{"name":"MASTER_USER","value":null,"valueFrom":{"secretKeyRef":
{"key":"MASTER_USER","name":"system-seed"}}},
{"name":"MASTER_PASSWORD","value":null,"valueFrom":{"secretKeyRef":
{"key":"MASTER_PASSWORD","name":"system-seed"}}},
{"name":"ADMIN_ACCESS_TOKEN","value":null,"valueFrom":{"secretKeyRef":
{"key":"ADMIN_ACCESS_TOKEN","name":"system-seed"}}},
{"name":"USER_LOGIN","value":null,"valueFrom":{"secretKeyRef":
{"key":"ADMIN_USER","name":"system-seed"}}},
{"name":"USER_PASSWORD","value":null,"valueFrom":{"secretKeyRef":
{"key":"ADMIN_PASSWORD","name":"system-seed"}}},
{"name":"TENANT_NAME","value":null,"valueFrom":{"secretKeyRef":
{"key":"TENANT_NAME","name":"system-seed"}}},
{"name":"THINKING_SPHINX_ADDRESS","value":"system-sphinx"},
{"name":"THINKING_SPHINX_CONFIGURATION_FILE","value":"/tmp/sphinx.conf"},
{"name":"EVENTS_SHARED_SECRET","value":null,"valueFrom":{"secretKeyRef":
{"key":"PASSWORD","name":"system-events-hook"}}},
{"name":"RECAPTCHA_PUBLIC_KEY","value":null,"valueFrom":{"secretKeyRef":
{"key":"PUBLIC_KEY","name":"system-recaptcha"}}},
{"name":"RECAPTCHA_PRIVATE_KEY","value":null,"valueFrom":{"secretKeyRef":
{"key":"PRIVATE_KEY","name":"system-recaptcha"}}},
{"name":"SECRET_KEY_BASE","value":null,"valueFrom":{"secretKeyRef":
{"key":"SECRET_KEY_BASE","name":"system-app"}}},
{"name":"REDIS_URL","value":null,"valueFrom":{"secretKeyRef":
{"key":"URL","name":"system-redis"}}},
{"name":"MEMCACHE_SERVERS","value":null,"valueFrom":{"secretKeyRef":
{"key":"SERVERS","name":"system-memcache"}}},
{"name":"BACKEND_REDIS_URL","value":null,"valueFrom":{"secretKeyRef":
{"key":"REDIS_STORAGE_URL","name":"backend-redis"}}},
{"name":"APICAST_BACKEND_ROOT_ENDPOINT","value":null,"valueFrom":
{"secretKeyRef":{"key":"route_endpoint","name":"backend-listener"}}},
{"name":"BACKEND_ROUTE","value":null,"valueFrom":{"secretKeyRef":
{"key":"route_endpoint","name":"backend-listener"}}},
{"name":"SMTP_ADDRESS","valueFrom":{"configMapKeyRef":
{"key":"address","name":"smtp"}},{"name":"SMTP_USER_NAME","valueFrom":
{"configMapKeyRef":{"key":"username","name":"smtp"}},

```

```

{"name":"SMTP_PASSWORD","valueFrom":{"configMapKeyRef":
{"key":"password","name":"smtp"}},{name":"SMTP_DOMAIN","valueFrom":
{"configMapKeyRef":{"key":"domain","name":"smtp"}},{name":"SMTP_PORT","valueFrom":
{"configMapKeyRef":{"key":"port","name":"smtp"}},
{"name":"SMTP_AUTHENTICATION","valueFrom":{"configMapKeyRef":
{"key":"authentication","name":"smtp"}},
{"name":"SMTP_OPENSSL_VERIFY_MODE","valueFrom":{"configMapKeyRef":
{"key":"openssl.verify.mode","name":"smtp"}},
{"name":"APICAST_ACCESS_TOKEN","value":null,"valueFrom":{"secretKeyRef":
{"key":"ACCESS_TOKEN","name":"system-master-apicast"}},
{"name":"ZYNC_AUTHENTICATION_TOKEN","value":null,"valueFrom":{"secretKeyRef":
{"key":"ZYNC_AUTHENTICATION_TOKEN","name":"zync"}},
{"name":"CONFIG_INTERNAL_API_USER","value":null,"valueFrom":{"secretKeyRef":
{"key":"username","name":"backend-internal-api"}},
{"name":"CONFIG_INTERNAL_API_PASSWORD","value":null,"valueFrom":{"secretKeyRef":
{"key":"password","name":"backend-internal-api"}},},,"template":{"spec":{"containers":
[{"name":"system-master","env":{"name":"AMP_RELEASE","value":null,"valueFrom":
{"configMapKeyRef":{"key":"AMP_RELEASE","name":"system-environment"}},
{"name":"APICAST_REGISTRY_URL","value":null,"valueFrom":{"configMapKeyRef":
{"key":"APICAST_REGISTRY_URL","name":"system-environment"}},
{"name":"FORCE_SSL","value":null,"valueFrom":{"configMapKeyRef":
{"key":"FORCE_SSL","name":"system-environment"}},
{"name":"PROVIDER_PLAN","value":null,"valueFrom":{"configMapKeyRef":
{"key":"PROVIDER_PLAN","name":"system-environment"}},
{"name":"RAILS_ENV","value":null,"valueFrom":{"configMapKeyRef":
{"key":"RAILS_ENV","name":"system-environment"}},
{"name":"RAILS_LOG_LEVEL","value":null,"valueFrom":{"configMapKeyRef":
{"key":"RAILS_LOG_LEVEL","name":"system-environment"}},
{"name":"RAILS_LOG_TO_STDOUT","value":null,"valueFrom":{"configMapKeyRef":
{"key":"RAILS_LOG_TO_STDOUT","name":"system-environment"}},
{"name":"SSL_CERT_DIR","value":null,"valueFrom":{"configMapKeyRef":
{"key":"SSL_CERT_DIR","name":"system-environment"}},
{"name":"THINKING_SPHINX_PORT","value":null,"valueFrom":{"configMapKeyRef":
{"key":"THINKING_SPHINX_PORT","name":"system-environment"}},
{"name":"THREESCALE_SANDBOX_PROXY_OPENSSL_VERIFY_MODE","value":null,"value
From":{"configMapKeyRef":
{"key":"THREESCALE_SANDBOX_PROXY_OPENSSL_VERIFY_MODE","name":"system-
environment"}},{"name":"THREESCALE_SUPERDOMAIN","value":null,"valueFrom":
{"configMapKeyRef":{"key":"THREESCALE_SUPERDOMAIN","name":"system-
environment"}},{"name":"DATABASE_URL","value":null,"valueFrom":{"secretKeyRef":
{"key":"URL","name":"system-database"}},
{"name":"MASTER_DOMAIN","value":null,"valueFrom":{"secretKeyRef":
{"key":"MASTER_DOMAIN","name":"system-seed"}},
{"name":"MASTER_USER","value":null,"valueFrom":{"secretKeyRef":
{"key":"MASTER_USER","name":"system-seed"}},
{"name":"MASTER_PASSWORD","value":null,"valueFrom":{"secretKeyRef":
{"key":"MASTER_PASSWORD","name":"system-seed"}},
{"name":"ADMIN_ACCESS_TOKEN","value":null,"valueFrom":{"secretKeyRef":
{"key":"ADMIN_ACCESS_TOKEN","name":"system-seed"}},
{"name":"USER_LOGIN","value":null,"valueFrom":{"secretKeyRef":
{"key":"ADMIN_USER","name":"system-seed"}},
{"name":"USER_PASSWORD","value":null,"valueFrom":{"secretKeyRef":
{"key":"ADMIN_PASSWORD","name":"system-seed"}},
{"name":"TENANT_NAME","value":null,"valueFrom":{"secretKeyRef":
{"key":"TENANT_NAME","name":"system-seed"}},
{"name":"EVENTS_SHARED_SECRET","value":null,"valueFrom":{"secretKeyRef":

```

```

{"key":"PASSWORD","name":"system-events-hook"}},
{"name":"RECAPTCHA_PUBLIC_KEY","value":null,"valueFrom":{"secretKeyRef":
{"key":"PUBLIC_KEY","name":"system-recaptcha"}},
{"name":"RECAPTCHA_PRIVATE_KEY","value":null,"valueFrom":{"secretKeyRef":
{"key":"PRIVATE_KEY","name":"system-recaptcha"}},
{"name":"SECRET_KEY_BASE","value":null,"valueFrom":{"secretKeyRef":
{"key":"SECRET_KEY_BASE","name":"system-app"}},
{"name":"REDIS_URL","value":null,"valueFrom":{"secretKeyRef":
{"key":"URL","name":"system-redis"}},
{"name":"MEMCACHE_SERVERS","value":null,"valueFrom":{"secretKeyRef":
{"key":"SERVERS","name":"system-memcache"}},
{"name":"BACKEND_REDIS_URL","value":null,"valueFrom":{"secretKeyRef":
{"key":"REDIS_STORAGE_URL","name":"backend-redis"}},
{"name":"APICAST_BACKEND_ROOT_ENDPOINT","value":null,"valueFrom":
{"secretKeyRef":{"key":"route_endpoint","name":"backend-listener"}},
{"name":"BACKEND_ROUTE","value":null,"valueFrom":{"secretKeyRef":
{"key":"route_endpoint","name":"backend-listener"}},
{"name":"APICAST_ACCESS_TOKEN","value":null,"valueFrom":{"secretKeyRef":
{"key":"ACCESS_TOKEN","name":"system-master-apicast"}},
{"name":"ZYNC_AUTHENTICATION_TOKEN","value":null,"valueFrom":{"secretKeyRef":
{"key":"ZYNC_AUTHENTICATION_TOKEN","name":"zync"}},
{"name":"CONFIG_INTERNAL_API_USER","value":null,"valueFrom":{"secretKeyRef":
{"key":"username","name":"backend-internal-api"}},
{"name":"CONFIG_INTERNAL_API_PASSWORD","value":null,"valueFrom":{"secretKeyRef":
{"key":"password","name":"backend-internal-api"}},{"name":"system-provider","env":
[{"name":"AMP_RELEASE","value":null,"valueFrom":{"configMapKeyRef":
{"key":"AMP_RELEASE","name":"system-environment"}},
{"name":"APICAST_REGISTRY_URL","value":null,"valueFrom":{"configMapKeyRef":
{"key":"APICAST_REGISTRY_URL","name":"system-environment"}},
{"name":"FORCE_SSL","value":null,"valueFrom":{"configMapKeyRef":
{"key":"FORCE_SSL","name":"system-environment"}},
{"name":"PROVIDER_PLAN","value":null,"valueFrom":{"configMapKeyRef":
{"key":"PROVIDER_PLAN","name":"system-environment"}},
{"name":"RAILS_ENV","value":null,"valueFrom":{"configMapKeyRef":
{"key":"RAILS_ENV","name":"system-environment"}},
{"name":"RAILS_LOG_LEVEL","value":null,"valueFrom":{"configMapKeyRef":
{"key":"RAILS_LOG_LEVEL","name":"system-environment"}},
{"name":"RAILS_LOG_TO_STDOUT","value":null,"valueFrom":{"configMapKeyRef":
{"key":"RAILS_LOG_TO_STDOUT","name":"system-environment"}},
{"name":"SSL_CERT_DIR","value":null,"valueFrom":{"configMapKeyRef":
{"key":"SSL_CERT_DIR","name":"system-environment"}},
{"name":"THINKING_SPHINX_PORT","value":null,"valueFrom":{"configMapKeyRef":
{"key":"THINKING_SPHINX_PORT","name":"system-environment"}},
{"name":"THREESCALE_SANDBOX_PROXY_OPENSSL_VERIFY_MODE","value":null,"value
From":{"configMapKeyRef":
{"key":"THREESCALE_SANDBOX_PROXY_OPENSSL_VERIFY_MODE","name":"system-
environment"}},{"name":"THREESCALE_SUPERDOMAIN","value":null,"valueFrom":
{"configMapKeyRef":{"key":"THREESCALE_SUPERDOMAIN","name":"system-
environment"}},{"name":"DATABASE_URL","value":null,"valueFrom":{"secretKeyRef":
{"key":"URL","name":"system-database"}},
{"name":"MASTER_DOMAIN","value":null,"valueFrom":{"secretKeyRef":
{"key":"MASTER_DOMAIN","name":"system-seed"}},
{"name":"MASTER_USER","value":null,"valueFrom":{"secretKeyRef":
{"key":"MASTER_USER","name":"system-seed"}},
{"name":"MASTER_PASSWORD","value":null,"valueFrom":{"secretKeyRef":
{"key":"MASTER_PASSWORD","name":"system-seed"}},

```

```

{"name":"ADMIN_ACCESS_TOKEN","value":null,"valueFrom":{"secretKeyRef":
{"key":"ADMIN_ACCESS_TOKEN","name":"system-seed"}}},
{"name":"USER_LOGIN","value":null,"valueFrom":{"secretKeyRef":
{"key":"ADMIN_USER","name":"system-seed"}}},
{"name":"USER_PASSWORD","value":null,"valueFrom":{"secretKeyRef":
{"key":"ADMIN_PASSWORD","name":"system-seed"}}},
{"name":"TENANT_NAME","value":null,"valueFrom":{"secretKeyRef":
{"key":"TENANT_NAME","name":"system-seed"}}},
{"name":"EVENTS_SHARED_SECRET","value":null,"valueFrom":{"secretKeyRef":
{"key":"PASSWORD","name":"system-events-hook"}}},
{"name":"RECAPTCHA_PUBLIC_KEY","value":null,"valueFrom":{"secretKeyRef":
{"key":"PUBLIC_KEY","name":"system-recaptcha"}}},
{"name":"RECAPTCHA_PRIVATE_KEY","value":null,"valueFrom":{"secretKeyRef":
{"key":"PRIVATE_KEY","name":"system-recaptcha"}}},
{"name":"SECRET_KEY_BASE","value":null,"valueFrom":{"secretKeyRef":
{"key":"SECRET_KEY_BASE","name":"system-app"}}},
{"name":"REDIS_URL","value":null,"valueFrom":{"secretKeyRef":
{"key":"URL","name":"system-redis"}}},
{"name":"MEMCACHE_SERVERS","value":null,"valueFrom":{"secretKeyRef":
{"key":"SERVERS","name":"system-memcache"}}},
{"name":"BACKEND_REDIS_URL","value":null,"valueFrom":{"secretKeyRef":
{"key":"REDIS_STORAGE_URL","name":"backend-redis"}}},
{"name":"APICAST_BACKEND_ROOT_ENDPOINT","value":null,"valueFrom":
{"secretKeyRef":{"key":"route_endpoint","name":"backend-listener"}}},
{"name":"BACKEND_ROUTE","value":null,"valueFrom":{"secretKeyRef":
{"key":"route_endpoint","name":"backend-listener"}}},
{"name":"APICAST_ACCESS_TOKEN","value":null,"valueFrom":{"secretKeyRef":
{"key":"ACCESS_TOKEN","name":"system-master-apicast"}}},
{"name":"ZYNC_AUTHENTICATION_TOKEN","value":null,"valueFrom":{"secretKeyRef":
{"key":"ZYNC_AUTHENTICATION_TOKEN","name":"zync"}}},
{"name":"CONFIG_INTERNAL_API_USER","value":null,"valueFrom":{"secretKeyRef":
{"key":"username","name":"backend-internal-api"}}},
{"name":"CONFIG_INTERNAL_API_PASSWORD","value":null,"valueFrom":{"secretKeyRef":
{"key":"password","name":"backend-internal-api"}}},{"name":"system-developer","env":
[{"name":"AMP_RELEASE","value":null,"valueFrom":{"configMapKeyRef":
{"key":"AMP_RELEASE","name":"system-environment"}}},
{"name":"APICAST_REGISTRY_URL","value":null,"valueFrom":{"configMapKeyRef":
{"key":"APICAST_REGISTRY_URL","name":"system-environment"}}},
{"name":"FORCE_SSL","value":null,"valueFrom":{"configMapKeyRef":
{"key":"FORCE_SSL","name":"system-environment"}}},
{"name":"PROVIDER_PLAN","value":null,"valueFrom":{"configMapKeyRef":
{"key":"PROVIDER_PLAN","name":"system-environment"}}},
{"name":"RAILS_ENV","value":null,"valueFrom":{"configMapKeyRef":
{"key":"RAILS_ENV","name":"system-environment"}}},
{"name":"RAILS_LOG_LEVEL","value":null,"valueFrom":{"configMapKeyRef":
{"key":"RAILS_LOG_LEVEL","name":"system-environment"}}},
{"name":"RAILS_LOG_TO_STDOUT","value":null,"valueFrom":{"configMapKeyRef":
{"key":"RAILS_LOG_TO_STDOUT","name":"system-environment"}}},
{"name":"SSL_CERT_DIR","value":null,"valueFrom":{"configMapKeyRef":
{"key":"SSL_CERT_DIR","name":"system-environment"}}},
{"name":"THINKING_SPHINX_PORT","value":null,"valueFrom":{"configMapKeyRef":
{"key":"THINKING_SPHINX_PORT","name":"system-environment"}}},
{"name":"THREESCALE_SANDBOX_PROXY_OPENSSL_VERIFY_MODE","value":null,"value
From":{"configMapKeyRef":
{"key":"THREESCALE_SANDBOX_PROXY_OPENSSL_VERIFY_MODE","name":"system-
environment"}}},{"name":"THREESCALE_SUPERDOMAIN","value":null,"valueFrom":

```

```

{"configMapKeyRef":{"key":"THREESCALE_SUPERDOMAIN","name":"system-environment"}}, {"name":"DATABASE_URL","value":null,"valueFrom":{"secretKeyRef":{"key":"URL","name":"system-database"}}}, {"name":"MASTER_DOMAIN","value":null,"valueFrom":{"secretKeyRef":{"key":"MASTER_DOMAIN","name":"system-seed"}}}, {"name":"MASTER_USER","value":null,"valueFrom":{"secretKeyRef":{"key":"MASTER_USER","name":"system-seed"}}}, {"name":"MASTER_PASSWORD","value":null,"valueFrom":{"secretKeyRef":{"key":"MASTER_PASSWORD","name":"system-seed"}}}, {"name":"ADMIN_ACCESS_TOKEN","value":null,"valueFrom":{"secretKeyRef":{"key":"ADMIN_ACCESS_TOKEN","name":"system-seed"}}}, {"name":"USER_LOGIN","value":null,"valueFrom":{"secretKeyRef":{"key":"ADMIN_USER","name":"system-seed"}}}, {"name":"USER_PASSWORD","value":null,"valueFrom":{"secretKeyRef":{"key":"ADMIN_PASSWORD","name":"system-seed"}}}, {"name":"TENANT_NAME","value":null,"valueFrom":{"secretKeyRef":{"key":"TENANT_NAME","name":"system-seed"}}}, {"name":"EVENTS_SHARED_SECRET","value":null,"valueFrom":{"secretKeyRef":{"key":"PASSWORD","name":"system-events-hook"}}}, {"name":"RECAPTCHA_PUBLIC_KEY","value":null,"valueFrom":{"secretKeyRef":{"key":"PUBLIC_KEY","name":"system-recaptcha"}}}, {"name":"RECAPTCHA_PRIVATE_KEY","value":null,"valueFrom":{"secretKeyRef":{"key":"PRIVATE_KEY","name":"system-recaptcha"}}}, {"name":"SECRET_KEY_BASE","value":null,"valueFrom":{"secretKeyRef":{"key":"SECRET_KEY_BASE","name":"system-app"}}}, {"name":"REDIS_URL","value":null,"valueFrom":{"secretKeyRef":{"key":"URL","name":"system-redis"}}}, {"name":"MEMCACHE_SERVERS","value":null,"valueFrom":{"secretKeyRef":{"key":"SERVERS","name":"system-memcache"}}}, {"name":"BACKEND_REDIS_URL","value":null,"valueFrom":{"secretKeyRef":{"key":"REDIS_STORAGE_URL","name":"backend-redis"}}}, {"name":"APICAST_BACKEND_ROOT_ENDPOINT","value":null,"valueFrom":{"secretKeyRef":{"key":"route_endpoint","name":"backend-listener"}}}, {"name":"BACKEND_ROUTE","value":null,"valueFrom":{"secretKeyRef":{"key":"route_endpoint","name":"backend-listener"}}}, {"name":"APICAST_ACCESS_TOKEN","value":null,"valueFrom":{"secretKeyRef":{"key":"ACCESS_TOKEN","name":"system-master-apicast"}}}, {"name":"ZYNC_AUTHENTICATION_TOKEN","value":null,"valueFrom":{"secretKeyRef":{"key":"ZYNC_AUTHENTICATION_TOKEN","name":"zync"}}}, {"name":"CONFIG_INTERNAL_API_USER","value":null,"valueFrom":{"secretKeyRef":{"key":"username","name":"backend-internal-api"}}}, {"name":"CONFIG_INTERNAL_API_PASSWORD","value":null,"valueFrom":{"secretKeyRef":{"key":"password","name":"backend-internal-api"}}}], "volumes":{"configMap":{"defaultMode":420,"items":[{"key":"zync.yml","path":"zync.yml"}, {"key":"rolling_updates.yml","path":"rolling_updates.yml"}, {"key":"service_discovery.yml","path":"service_discovery.yml"}],"name":"system","name":"system-config"}}}}}'

```

5. **system-sidekiq** DeploymentConfig を更新して、**新規シークレット** からシステムのデータベース情報を収集します。

```

oc patch dc/system-sidekiq -p '{"spec":{"template":{"spec":{"containers":[{"name":"system-sidekiq","env":[{"name":"AMP_RELEASE","value":null,"valueFrom":{"configMapKeyRef":{"key":"AMP_RELEASE","name":"system-environment"}}}, {"name":"APICAST_REGISTRY_URL","value":null,"valueFrom":{"configMapKeyRef":{"key":"APICAST_REGISTRY_URL","name":"system-environment"}}},

```

```

{"name":"FORCE_SSL","value":null,"valueFrom":{"configMapKeyRef":
{"key":"FORCE_SSL","name":"system-environment"}}},
{"name":"PROVIDER_PLAN","value":null,"valueFrom":{"configMapKeyRef":
{"key":"PROVIDER_PLAN","name":"system-environment"}}},
{"name":"RAILS_ENV","value":null,"valueFrom":{"configMapKeyRef":
{"key":"RAILS_ENV","name":"system-environment"}}},
{"name":"RAILS_LOG_LEVEL","value":null,"valueFrom":{"configMapKeyRef":
{"key":"RAILS_LOG_LEVEL","name":"system-environment"}}},
{"name":"RAILS_LOG_TO_STDOUT","value":null,"valueFrom":{"configMapKeyRef":
{"key":"RAILS_LOG_TO_STDOUT","name":"system-environment"}}},
{"name":"SSL_CERT_DIR","value":null,"valueFrom":{"configMapKeyRef":
{"key":"SSL_CERT_DIR","name":"system-environment"}}},
{"name":"THINKING_SPHINX_PORT","value":null,"valueFrom":{"configMapKeyRef":
{"key":"THINKING_SPHINX_PORT","name":"system-environment"}}},
{"name":"THREESCALE_SANDBOX_PROXY_OPENSSL_VERIFY_MODE","value":null,"value
From":{"configMapKeyRef":
{"key":"THREESCALE_SANDBOX_PROXY_OPENSSL_VERIFY_MODE","name":"system-
environment"}}},{"name":"THREESCALE_SUPERDOMAIN","value":null,"valueFrom":
{"configMapKeyRef":{"key":"THREESCALE_SUPERDOMAIN","name":"system-
environment"}}},{"name":"DATABASE_URL","value":null,"valueFrom":{"secretKeyRef":
{"key":"URL","name":"system-database"}}},
{"name":"MASTER_DOMAIN","value":null,"valueFrom":{"secretKeyRef":
{"key":"MASTER_DOMAIN","name":"system-seed"}}},
{"name":"MASTER_USER","value":null,"valueFrom":{"secretKeyRef":
{"key":"MASTER_USER","name":"system-seed"}}},
{"name":"MASTER_PASSWORD","value":null,"valueFrom":{"secretKeyRef":
{"key":"MASTER_PASSWORD","name":"system-seed"}}},
{"name":"ADMIN_ACCESS_TOKEN","value":null,"valueFrom":{"secretKeyRef":
{"key":"ADMIN_ACCESS_TOKEN","name":"system-seed"}}},
{"name":"USER_LOGIN","value":null,"valueFrom":{"secretKeyRef":
{"key":"ADMIN_USER","name":"system-seed"}}},
{"name":"USER_PASSWORD","value":null,"valueFrom":{"secretKeyRef":
{"key":"ADMIN_PASSWORD","name":"system-seed"}}},
{"name":"TENANT_NAME","value":null,"valueFrom":{"secretKeyRef":
{"key":"TENANT_NAME","name":"system-seed"}}},
{"name":"EVENTS_SHARED_SECRET","value":null,"valueFrom":{"secretKeyRef":
{"key":"PASSWORD","name":"system-events-hook"}}},
{"name":"RECAPTCHA_PUBLIC_KEY","value":null,"valueFrom":{"secretKeyRef":
{"key":"PUBLIC_KEY","name":"system-recaptcha"}}},
{"name":"RECAPTCHA_PRIVATE_KEY","value":null,"valueFrom":{"secretKeyRef":
{"key":"PRIVATE_KEY","name":"system-recaptcha"}}},
{"name":"SECRET_KEY_BASE","value":null,"valueFrom":{"secretKeyRef":
{"key":"SECRET_KEY_BASE","name":"system-app"}}},
{"name":"REDIS_URL","value":null,"valueFrom":{"secretKeyRef":
{"key":"URL","name":"system-redis"}}},
{"name":"MEMCACHE_SERVERS","value":null,"valueFrom":{"secretKeyRef":
{"key":"SERVERS","name":"system-memcache"}}},
{"name":"BACKEND_REDIS_URL","value":null,"valueFrom":{"secretKeyRef":
{"key":"REDIS_STORAGE_URL","name":"backend-redis"}}},
{"name":"APICAST_BACKEND_ROOT_ENDPOINT","value":null,"valueFrom":
{"secretKeyRef":{"key":"route_endpoint","name":"backend-listener"}}},
{"name":"BACKEND_ROUTE","value":null,"valueFrom":{"secretKeyRef":
{"key":"route_endpoint","name":"backend-listener"}}},
{"name":"APICAST_ACCESS_TOKEN","value":null,"valueFrom":{"secretKeyRef":
{"key":"ACCESS_TOKEN","name":"system-master-apicast"}}},
{"name":"ZYNC_AUTHENTICATION_TOKEN","value":null,"valueFrom":{"secretKeyRef":

```

```
{
  "key": "ZYNC_AUTHENTICATION_TOKEN", "name": "zync"
},
{
  "name": "CONFIG_INTERNAL_API_USER", "value": null, "valueFrom": { "secretKeyRef": { "key": "username", "name": "backend-internal-api" } }
},
{
  "name": "CONFIG_INTERNAL_API_PASSWORD", "value": null, "valueFrom": { "secretKeyRef": { "key": "password", "name": "backend-internal-api" } }
},
{
  "name": "check-svc", "command": [ "bash", "-c", "bundle exec sh -c \"until rake boot:redis && curl --output /dev/null --silent --fail --head http://system-master:3000/status; do sleep $SLEEP_SECONDS; done\" ], "env": { "name": "REDIS_URL", "value": null, "valueFrom": { "secretKeyRef": { "key": "URL", "name": "system-redis" } } }
},
{
  "name": "system-config", "volumes": { "configMap": { "defaultMode": 420, "items": [ { "key": "zync.yml", "path": "zync.yml" }, { "key": "rolling_updates.yml", "path": "rolling_updates.yml" }, { "key": "service_discovery.yml", "path": "service_discovery.yml" } ] } }
}
```

6. **system-sphinx** DeploymentConfig を更新し、**新規シークレット** からシステムのデータベース情報を収集します。

```
oc patch dc/system-sphinx -p '{"spec":{"template":{"spec":{"containers":[{"name":"system-sphinx","env":[{"name":"DATABASE_URL","value":"","valueFrom":{"secretKeyRef":{"key":"URL","name":"system-database"}}}]}}]}}'}
```

7. **apicast-staging** DeploymentConfig コンテナにパッチを適用します。

```
oc patch dc/apicast-staging -p '{"spec":{"template":{"spec":{"containers":[{"name":"apicast-staging","env":[{"name":"THREESCALE_PORTAL_ENDPOINT","value":null,"valueFrom":{"secretKeyRef":{"key":"PROXY_CONFIGS_ENDPOINT","name":"system-master-apicast"}}}],{"name":"BACKEND_ENDPOINT_OVERRIDE","value":null,"valueFrom":{"secretKeyRef":{"key":"service_endpoint","name":"backend-listener"}}}],{"name":"APICAST_MANAGEMENT_API","value":null,"valueFrom":{"configMapKeyRef":{"key":"APICAST_MANAGEMENT_API","name":"apicast-environment"}}}],{"name":"OPENSSL_VERIFY","value":null,"valueFrom":{"configMapKeyRef":{"key":"OPENSSL_VERIFY","name":"apicast-environment"}}}],{"name":"APICAST_RESPONSE_CODES","value":null,"valueFrom":{"configMapKeyRef":{"key":"APICAST_RESPONSE_CODES","name":"apicast-environment"}}}],{"name":"APICAST_CONFIGURATION_LOADER","value":"lazy"}, {"name":"APICAST_CONFIGURATION_CACHE","value":"0"}, {"name":"THREESCALE_DEPLOYMENT_ENV","value":"staging"}, {"name":"REDIS_URL","value":null,"valueFrom":{"secretKeyRef":{"key":"STAGING_URL","name":"apicast-redis"}}}]}}]}}'}
```

8. **apicast-production** DeploymentConfig コンテナにパッチを適用します。

```
oc patch dc/apicast-production -p '{"spec":{"template":{"spec":{"containers":[{"name":"apicast-production","env":[{"name":"THREESCALE_PORTAL_ENDPOINT","value":null,"valueFrom":{"secretKeyRef":{"key":"PROXY_CONFIGS_ENDPOINT","name":"system-master-apicast"}}}],{"name":"BACKEND_ENDPOINT_OVERRIDE","value":null,"valueFrom":{"secretKeyRef":{"key":"service_endpoint","name":"backend-listener"}}}],{"name":"APICAST_MANAGEMENT_API","value":null,"valueFrom":{"configMapKeyRef":{"key":"APICAST_MANAGEMENT_API","name":"apicast-environment"}}}],{"name":"OPENSSL_VERIFY","value":null,"valueFrom":{"configMapKeyRef":{"key":"OPENSSL_VERIFY","name":"apicast-environment"}}}],{"name":"APICAST_RESPONSE_CODES","value":null,"valueFrom":{"configMapKeyRef":{"key":"APICAST_RESPONSE_CODES","name":"apicast-environment"}}}],{"name":"APICAST_CONFIGURATION_LOADER","value":"boot"}, {"name":"APICAST_CONFIGURATION_CACHE","value":"300"}, {"name":
```


- 3. **amp-backend** イメージストリームにパッチを適用します。

```
oc patch imagestream/amp-backend --type=json -p [{"op": "add", "path": "/spec/tags/-",
"value": {"annotations": {"openshift.io/display-name": "AMP Backend 2.4.0"}, "from": { "kind":
"DockerImage", "name": "registry.access.redhat.com/3scale-amp24/backend"}, "name":
"2.4.0", "referencePolicy": {"type": "Source"}}}]'
oc patch imagestream/amp-backend --type=json -p [{"op": "add", "path": "/spec/tags/-",
"value": {"annotations": {"openshift.io/display-name": "AMP Backend (latest)"}, "from": {
"kind": "ImageStreamTag", "name": "2.4.0"}, "name": "latest", "referencePolicy": {"type":
"Source"}}}]'
```

- 4. **amp-zync** イメージストリームにパッチを適用します。

```
oc patch imagestream/amp-zync --type=json -p [{"op": "add", "path": "/spec/tags/-", "value":
{"annotations": {"openshift.io/display-name": "AMP Zync 2.4.0"}, "from": { "kind":
"DockerImage", "name": "registry.access.redhat.com/3scale-amp24/zync"}, "name": "2.4.0",
"referencePolicy": {"type": "Source"}}}]'
oc patch imagestream/amp-zync --type=json -p [{"op": "add", "path": "/spec/tags/-", "value":
{"annotations": {"openshift.io/display-name": "AMP Zync (latest)"}, "from": { "kind":
"ImageStreamTag", "name": "2.4.0"}, "name": "latest", "referencePolicy": {"type": "Source"}}}]'
```

1.2.7. サービスディスカバリーのエントリーの追加

アップグレードプロセスの一環として、サービスディスカバリーの ConfigMap エントリーを追加します。サービスディスカバリーは、OpenShift クラスタで検出可能な実行中のサービスを認識することで、API の管理を可能にする機能です。

1. システム ConfigMap を編集して、エントリーを追加します。

```
oc edit configmap system
```

2. エントリーを追加します。

```
service_discovery.yml: |
  production:
    enabled: <%= cluster_token_file_exists = File.exists?(cluster_token_file_path =
'/var/run/secrets/kubernetes.io/serviceaccount/token') %>
    server_scheme: 'https'
    server_host: 'kubernetes.default.svc.cluster.local'
    server_port: 443
    bearer_token: "<%= File.read(cluster_token_file_path) if cluster_token_file_exists %>"
    authentication_method: service_account # can be service_account|oauth
    oauth_server_type: builtin # can be builtin|rh_sso
    client_id:
    client_secret:
    timeout: 1
    open_timeout: 1
    max_retry: 5
    verify_ssl: <%= OpenSSL::SSL::VERIFY_NONE %> # 0
```

『サービスディスカバリー』ガイドを参照して、サービスをインポートするその他の設定を継続することができます。

1.2.8. 追加の DeploymentConfig の設定

1. **system-memcache** DeploymentConfig にパッチを適用します。

```
oc patch dc/system-memcache --patch='{"spec":{"template":{"spec":{"containers":[{"name": "memcache", "image": "registry.access.redhat.com/3scale-amp20/memcached"}]}}}}'
```

2. **system-resque** DeploymentConfig を削除します。

```
oc delete dc/system-resque
```

3. **system-sidekiq** 並行処理を推奨されるレベルに増やすには、環境変数を設定します。

```
oc set env dc/system-sidekiq RAILS_MAX_THREADS=25
```

4. 環境変数を設定して、表示されるリリースバージョンを更新します。

```
oc set env dc/system-app AMP_RELEASE=2.4.0
```

1.3. 管理者 IMPERSONATION の変更（オプション）

3scale API Management はオープンソースであるため、偽装データは一般に公開されています。このため、一部のデータを変更する必要がある場合があります。

- 管理者の偽装に使用する一意のユーザー名。
- 管理者ユーザーの偽装メールのドメイン。

たとえば、username :<your-username> および domain:< example.com> を想定します。管理者の偽装を変更するには、以下の手順を実行する必要があります。

1. 以下の内容で、**system-impersonation-secret.yml** という名前のファイルをローカルに作成します。

```
cat > system-impersonation-secret.yml <<EOF

apiVersion: v1
kind: Secret
metadata:
  creationTimestamp: null
  labels:
    3scale.component: system
    app: ${APP_LABEL}
  name: system-impersonation
stringData:
  username: "<your-username>"
  domain: "<example.com>"
type: Opaque
EOF
```

2. <your-username> および <example.com> を、選択したユーザー名とドメインに変更します。
3. シークレットを作成します。

```
oc create -f system-impersonation-secret.yml
```

4. 以下を使用して、このシークレットから環境変数を設定します。

```
oc set env --from=secret/system-impersonation --prefix=IMPERSONATION_ADMIN  
dc/system-app
```

5. **system-app** を再デプロイします。

```
oc rollout latest system-app
```

6. **system-app** デプロイメントの **system-master** コンテナに接続します。

```
oc rsh -c system-master dc/system-app
```

7. このコンテナの実行で、**<your-username>** および **<example.com>** を適宜変更します。

```
bundle exec rake "impersonation_admin_user:update[<your-username>,<example.com>]"
```

これでユーザーインターフェースからテナントの権限を借用できるはずです。

第2章 ORACLE DATABASE リレーショナルデータベース管理システムを使用した 3SCALE API MANAGEMENT システムイメージのビルド

デフォルトでは、3scale には設定データを MySQL データベースに保管する **system** というコンポーネントが含まれています。任意で、デフォルトのデータベースをオーバーライドし、情報を外部の Oracle Database に保管することができます。本章の手順に従って、独自の Oracle Database クライアントバイナリーでカスタムのシステムコンテナイメージをビルドし、3scale を OpenShift にデプロイします。

2.1. 作業を開始する前に

2.1.1. Oracle ソフトウェアコンポーネントの取得

カスタムの 3scale システムコンテナイメージをビルドする前に、[以下の](#) Oracle ソフトウェアコンポーネントのサポート対象バージョンを取得する必要があります。

- Oracle Instant Client Package Basic または Basic Light
- Oracle Instant Client パッケージ SDK
- Oracle Instant Client パッケージ ODBC

2.1.2. 会議の前提条件

以下の前提条件も満たす必要があります。

- OpenShift クラスターからアクセスできる Oracle Database の [サポート対象バージョン](#)
- インストール手順に必要な Oracle Database の **system** ユーザーへのアクセス
- Red Hat 3scale 2.4 amp.yml テンプレート

2.2. ORACLE DATABASE の準備

1. 新規データベースの作成

Oracle Database が 3scale と動作するようにするには、以下の設定が必要です。

```
ALTER SYSTEM SET max_string_size=extended SCOPE=SPFILE;
```

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

2. データベースの詳細を収集します。

3scale の設定に必要な以下の情報を取得します。

- Oracle Database の URL
- Oracle Database の [サービス名](#)
- Oracle Database の **system** ユーザー名およびパスワード
- Oracle Database の サービス名

Oracle Database で新規データベースを作成する方法については、Oracle の [ドキュメント](#) を参照してください。

2.3. システムイメージのビルド

1. [3scale-amp-openshift-templates github](#) リポジトリのクローンを作成します。
2. Oracle Database の Instant Client パッケージファイルを **3scale-amp-openshift-templates/amp/system-oracle/oracle-client-files** ディレクトリーに置きます。
3. `-f` オプションで **build.yml** OpenShift テンプレートを指定して、**oc new-app** コマンドを実行します。

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

4. `-f` オプションで、**amp.yml** OpenShift テンプレートおよび `-p` オプションを指定し、**WILDCARD_DOMAIN** パラメーターに OpenShift クラスターのドメインを指定して、**oc new-app** コマンドを実行します。

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

5. 「[Oracle Database の準備](#)」セクションで収集した以下の情報を指定して、`loop` で以下のシェールを入力します。

- **{USER}**: Oracle Database で 3scale を表すユーザー名
- **{PASSWORD}**: **USER** のパスワード
- **{ORACLE_DB_URL}**: Oracle Database の URL
- **{DATABASE}**: Oracle Database で作成したデータベースのサービス名
- **{PORT}**: Oracle Database のポート番号

```
for dc in system-app system-resque system-sidekiq system-sphinx; do oc env dc/$dc --
  overwrite DATABASE_URL="oracle-enhanced://{USER}:
  {PASSWORD}@{ORACLE_DB_URL}:{PORT}/{DATABASE}"; done
```

6. 以下の **oc patch** コマンドを入力し、上記の手順で指定した **USER**、**PASSWORD**、**ORACLE_DB_URL**、**PORT**、および **DATABASE** の値を指定します。

```
$ oc patch dc/system-app -p '{"op": "replace", "path":
  "/spec/strategy/rollingParams/pre/execNewPod/env/1/value", "value": "oracle-
  enhanced://{USER}:{PASSWORD}@{ORACLE_DB_URL}:{PORT}/{DATABASE}"}' --
  type=json
```

7. 以下の **oc patch** コマンドを入力し、**SYSTEM_PASSWORD** フィールドに独自の Oracle Database の **system** ユーザーのパスワードを指定します。

```
$ 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
```

8. **oc start-build** コマンドを入力し、新しいシステムイメージをビルドします。

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

第3章 オンプレミス型 3SCALE API MANAGEMENT インストールガイド

本書では、OpenShift テンプレートを使用して Red Hat 3scale API Management 2.4（オンプレミス）を OpenShift にインストールする手順を説明します。

3.1. 前提条件

- UTC 用に 3scale API Management サーバーを設定する必要があります（協定世界時）。
- OpenShift で（自己署名ではなく）有効な HTTPS 証明書を設定する必要があります。詳細は、「[Create Secure Routes on OpenShift](#)」を参照してください。

3.2. 3SCALE OPENSIFT テンプレート

3scale API Management 2.4 は OpenShift テンプレートを提供します。このテンプレートを使用して、3scale API Management を OpenShift Container Platform にデプロイすることができます。

3scale API Management テンプレートは、以下で構成されています。

- 2つの組み込み APIcast API ゲートウェイ
- 1つの 3scale API Management 管理ポータルおよび永続ストレージを持つデベロッパーポータル

3.3. システム要件

本セクションでは、3scale API Management OpenShift テンプレートの要件について説明します。

3.3.1. 環境要件

3scale API Management には、「[Red Hat 3scale API Management Supported Configurations](#)」で規定される環境が必要です。

永続ボリューム：

- Redis および MySQL の永続用の 3つの RWO (ReadWriteOnce) 永続ボリューム
- CMS および System-app Assets 用の 1つの RWX (ReadWriteMany) 永続ボリューム

RWX 永続ボリュームはグループ書き込みができるように設定する必要があります。必要なアクセスモードをサポートする永続ボリュームタイプのリストは、[OpenShift のドキュメント](#)を参照してください。

3.3.2. ハードウェア要件

ハードウェア要件は、使用の必要性に応じて異なります。Red Hat は、テストを行い個々の要件を満たすように環境を設定することを推奨します。OpenShift 上の 3scale の環境を設定する場合、以下が推奨されます。

- クラウド環境へのデプロイメントには、コンピュータタスクに最適化したノードを使用します (AWS c4.2xlarge または Azure Standard_F8)。

- メモリーの要件が現在のノードで使用できる RAM よりも大きい場合、非常に大きなインストールでは、Redis に別のノードが必要になることがあります (AWS M4 シリーズまたは Azure Av2 シリーズ)。
- ルーティングタスクとコンピュートタスクには別のノードを使用します。
- 3scale 固有のタスクには専用のコンピュートノードを使用します。
- バックエンドリスナーの **PUMA_WORKERS** 変数をコンピュートノードのコア数に設定します。

3.4. ノードおよびエンタイトルメントの設定

3scale を OpenShift にデプロイする前に、ノードおよびお使いの環境が Red Hat からイメージを取得するのに必要なエンタイトルメントを設定する必要があります。

以下の手順を実行し、エンタイトルメントを設定します。

1. 各ノードに [Red Hat Enterprise Linux \(RHEL\)](#) をインストールします。
2. [インターフェース](#) または [コマンドライン](#) で Red Hat Subscription Manager (RHSM) を使用し、Red Hat にノードを登録します。
3. RHSM を使用して [ノードを 3scale サブスクリプションに割り当てます](#)。
4. 以下の要件に準拠して、ノードに [OpenShift](#) をインストールします。
 - [サポート対象バージョンの OpenShift](#) を使用する。
 - 複数書き込みをサポートするファイルシステムで [永続ストレージ](#) を設定します。
5. [OpenShift コマンドラインインターフェース](#) をインストールします。
6. サブスクリプションマネージャーを使用して、**rhel-7-server-3scale-amp-2.4-rpms** リポジトリへのアクセスを有効にします。

```
sudo subscription-manager repos --enable=rhel-7-server-3scale-amp-2.4-rpms
```

7. **3scale-amp-template** をインストールします。テンプレートは **/opt/amp/templates** に保存されます。

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

3.5. テンプレートを使用した OPENSIFT への 3SCALE のデプロイ

3.5.1. 前提条件

- [第 3 章およびエンタイトルメントの設定 セクション](#)で指定された OpenShift クラスター。
- OpenShift クラスターに対して解決する [ドメイン](#) (できればワイルドカード)
- Red Hat [コンテナーカタログ](#) へのアクセス
- (オプション) 稼働中の電子メール機能用 SMTP サーバー

以下の手順に従って、a .yml テンプレートを使用して 3scale API Management を OpenShift にインストールします。

- [3scale API Management テンプレートのインポート](#)
- [SMTP 変数の設定 \(任意\)](#)

3.5.2. 3scale テンプレートのインポート

3scale API Management テンプレートを OpenShift クラスターにインポートするには、以下の手順を実施します。

1. ターミナルセッションから、OpenShift にクラスター管理者としてログインします。

```
oc login
```

2. プロジェクトを選択するか新しいプロジェクトを作成します。

```
oc project <project_name>
```

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

3. **oc new-app** コマンドを入力します。

- a. **--file** オプションを使用して、ノードおよびエンタイトルメント [の設定セクションの一部としてダウンロードした amp.yml ファイルへのパスを指定します](#)。
- b. **--param** オプションを使用して、**WILDCARD_DOMAIN** パラメーターに OpenShift クラスターのドメインを設定します。
- c. オプションで、**--param** オプションを使用して、**WILDCARD_POLICY** パラメーターを **subdomain** に設定してワイルドカードドメインルーティングを有効にします。
ワイルドカードルーティングを有効にしない場合:

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

ワイルドカードルーティングを有効にする場合:

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

ターミナルには、マスターおよびテナント URL と、新たに作成された 3scale API Management 管理ポータル [の](#) クレデンシャルが表示されます。この出力には以下の情報が含まれます。

- マスター管理者のユーザー名
- マスターのパスワード
- マスターのトークン情報
- テナントのユーザー名

- テナントのパスワード
- テナントのトークン情報

4. <https://user-admin.3scale-project.example.com> に admin/xXxYyz123 としてログインします。

* With parameters:

```
* ADMIN_PASSWORD=xXxYyz123 # generated
* ADMIN_USERNAME=admin
* TENANT_NAME=user

* MASTER_NAME=master
* MASTER_USER=master
* MASTER_PASSWORD=xXxYyz123 # 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'

5. 後で確認できるようにするため、詳細を書き留めておきます。



注記

3scale API Management が OpenShift に完全にデプロイして、ログインおよびクレデンシャルが有効になるまで、数分かかることがあります。

詳細情報

OpenShift でのワイルドカードドメインに関する詳細は、(サブドメインの) [ワイルドカードルートの使用について参照してください](#)。

3.5.3. SMTP 変数の設定 (任意)

OpenShift は、メールを使用して [通知を送信し](#)、[新規ユーザーを招待](#) します。これらの機能を使用する場合は、独自の SMTP サーバーを提供し、SMTP 設定マップで SMTP 変数を設定する必要があります。

SMTP ConfigMap で SMTP 変数を設定するには、以下の手順を実施します。

1. OpenShift にログインしていない場合はログインします。

oc login

- a. SMTP ConfigMap の変数を設定します。oc patch コマンドを使用して configmap および smtp オブジェクトを指定し、続いて -p オプションを指定し、以下の変数に対して JSON 形式で新しい値を指定します。

変数	説明

address	リモートメールサーバーをリレーとして指定できます。
username	メールサーバーのユーザー名を指定します。
password	メールサーバーのパスワードを指定します。
domain	HELO ドメインを指定します。
port	メールサーバーが新しい接続をリッスンするポートを指定します。
authentication	メールサーバーの認証タイプを指定します。指定できる値は plain (パスワードをクリアテキストで送信)、 login (パスワードを Base64 エンコードで送信)、または cram_md5 (ハッシュ関数に Message Digest 5 アルゴリズムを使用し認証情報を交換) です。
openssl.verify.mode	TLS の使用時に OpenSSL が証明書をチェックする方法を指定します。指定できる値は none 、 peer 、 client_once 、または fail_if_no_peer_cert です。

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>"}}'
```

2. configmap 変数を設定した後、**system-app** および **system-sidekiq** Pod を再デプロイします。

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

3.6. 3SCALE テンプレートパラメーター

テンプレートパラメーターは、デプロイ中およびデプロイメント後の 3scale API Management yml テンプレートの環境変数を設定します。

名前	説明	デフォルト値	必須/任意
APP_LABEL	オブジェクトアプリのラベルに使用されます。	"3scale-api-management"	はい

ZYNC_DATABASE_PASSWORD	PostgreSQL 接続ユーザーのパスワード。指定のない場合は無作為に生成されます。	該当せず	必須
ZYNC_SECRET_KEY_BASE	Zync の秘密鍵ベース。指定のない場合は無作為に生成されます。	該当せず	必須
ZYNC_AUTHENTICATION_TOKEN	Zync の認証トークン。指定のない場合は無作為に生成されます。	該当せず	必須
AMP_RELEASE	3scale API Management リリースタグ	2.4.0	はい
ADMIN_PASSWORD	無作為に生成される 3scale API Management 管理者アカウントのパスワード。	該当なし	必須
ADMIN_USERNAME	3scale API Management 管理者アカウントのユーザー名	admin	はい
APICAST_ACCESS_TOKEN	APIcast が設定のダウンロードに使用する読み取り専用アクセストークン	該当せず	必須
ADMIN_ACCESS_TOKEN	すべての API をスコープとし、書き込みアクセス権限が設定された管理者アクセストークン	該当せず	任意
WILDCARD_DOMAIN	ワイルドカードルートのルートドメイン。たとえば、ルートドメイン example.com は 3scale-admin.example.com を生成します。	該当せず	必須
WILDCARD_POLICY	値を「Subdomain」として設定して、組み込まれた APIcast ゲートウェイへのワイルドカードルートを有効にします。	なし	はい

TENANT_NAME	ルート下のテナント名。 -admin 接尾辞を付けて 管理 UI が利用できま す。	3scale	はい
MYSQL_USER	データベースのアクセス に使用される MySQL ユーザーのユーザー名	mysql	はい
MYSQL_PASSWORD	MySQL ユーザーのパス ワード	該当せず	必須
MYSQL_DATABASE	アクセスされた MySQL データベースの名前	system	はい
MYSQL_ROOT_PASSW ORD	Root ユーザーのパス ワード	該当せず	必須
SYSTEM_BACKEND_US ERNAME	内部 3scale api auth の 内部 3scale API ユー ザー名	3scale_api_user	はい
SYSTEM_BACKEND_PA SSWORD	内部 3scale api auth の 内部 3scale API パス ワード	該当せず	必須
REDIS_IMAGE	使用する Redis イメージ	registry.access.redhat.c om/rhscsl/redis-32- rhel7:3.2	はい
MYSQL_IMAGE	使用する Mysql イメージ	registry.access.redhat.c om/rhscsl/mysql-57- rhel7:5.7	はい
MEMCACHED_IMAGE	使用する Memcached イ メージ	registry.access.redhat.c om/3scale- amp20/memcached:1.4.1 5	はい
POSTGRESQL_IMAGE	使用する Postgresql イ メージ	registry.access.redhat.c om/rhscsl/postgresql- 95-rhel7:9.5	はい
AMP_SYSTEM_IMAGE	使用する 3scale システ ムイメージ	registry.access.redhat.c om/3scale- amp24/system	はい
AMP_BACKEND_IMAGE	使用する 3scale バック エンドイメージ	registry.access.redhat.c om/3scale- amp24/backend	はい

AMP_APICAST_IMAGE	使用する 3scale APIcast イメージ	registry.access.redhat.com/3scale-amp24/apicast-gateway	はい
AMP_ROUTER_IMAGE	使用する 3scale ワイルドカードルーターイメージ	registry.access.redhat.com/3scale-amp22/wildcard-router	はい
AMP_ZYNC_IMAGE	使用する 3scale Zync イメージ	registry.access.redhat.com/3scale-amp24/zync	はい
SYSTEM_BACKEND_SHARED_SECRET	バックエンドからシステムにイベントをインポートするための共有シークレット	該当せず	必須
SYSTEM_APP_SECRET_KEY_BASE	システムアプリケーションの秘密鍵ベース	該当せず	必須
APICAST_MANAGEMENT_API	APIcast Management API のスコープ。 disable、status、または debug を設定できます。ヘルスチェックには最低でも status が必要です。	status	いいえ
APICAST_OPENSSL_VERIFY	設定のダウンロード時に OpenSSL ピア検証を有効または無効にします。true または false を設定できます。	false	いいえ
APICAST_RESPONSE_CODES	APIcast のログインレスポンスコードを有効にします。	true	任意
APICAST_REGISTRY_URL	APIcast ポリシーの場所に解決する URL	http://apicast-staging:8090/policies	はい
MASTER_USER	マスター管理者アカウントのユーザー名	master	はい
MASTER_NAME	マスター管理ポータルの子ドメイン値。 - master 接尾辞が付けられます。	master	はい
MASTER_PASSWORD	無作為に生成されるマスター管理者のパスワード	該当せず	必須

MASTER_ACCESS_TOKEN	API呼び出しのマスターレベル権限が設定されたトークン	該当せず	必須
IMAGESTREAM_TAG_IMPORT_INSECURE	イメージのインポート中にサーバーが証明書の検証を回避できる、またはHTTP経由で直接接続できる場合は、trueを設定します。	false	はい

3.7. OPENSIFT 上で 3SCALE と APICAST を使用する

APIcast は 3scale SaaS 用の API Manager と OpenShift Container Platform におけるオンプレミスインストール用 API Manager で利用できます。両構成で、設定手順は異なります。本セクションでは、OpenShift 上で API Manager と共に APIcast をデプロイする方法を説明します。

3.7.1. 3scale が含まれる既存の OpenShift クラスターでの APIcast テンプレートのデプロイ

3scale API Management OpenShift テンプレートには、デフォルトで 2 つの組み込み APIcast API ゲートウェイが含まれています。より多くの API ゲートウェイが必要な場合や、別の APIcast デプロイメントが必要な場合は、追加の APIcast テンプレートを OpenShift クラスターにデプロイすることができます。

追加の API ゲートウェイを OpenShift クラスターにデプロイするには、以下の手順を実施します。

1. 以下の設定で [アクセストークン](#) を作成します。

- スコープ: Account Management API
- アクセス権限: 読み取り専用

2. APIcast クラスターにログインします。

```
oc login
```

3. APIcast が 3scale API Management と通信できるようにするシークレットを作成します。**new-basicauth**、**apicast-configuration-url-secret** を指定し、**--password** パラメーターで 3scale API Management デプロイメントのアクセストークン、テナント名、およびワイルドカードメインを設定します。

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



注記

TENANT_NAME は、管理 UI が利用可能なルート下の名前です。**TENANT_NAME** 3scale のデフォルト値。3scale API Management デプロイメントでカスタム値を使用した場合は、ここでその値を使用する必要があります。

4. APIcast テンプレート **apicast.yml** をローカルマシンにインストールします。

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

このコマンドにより、APIcast テンプレートをディレクトリ **/opt/amp/templates** にインストールします。

5. **--file** オプションで **apicast.yml** ファイルを指定して、**oc new-app** コマンドで APIcast テンプレートをインポートします。

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

3.7.2. 3scale が含まれる OpenShift クラスタ外部の OpenShift クラスタから APIcast を接続

3scale API Management クラスタ外部の別の OpenShift クラスタに APIcast をデプロイする場合は、パブリックルート経由で接続する必要があります。

1. 以下の設定で **アクセストークン** を作成します。

- スコープ: Account Management API
- アクセス権限: 読み取り専用

2. APIcast クラスタにログインします。

```
oc login
```

3. APIcast が 3scale API Management と通信できるようにするシークレットを作成します。**new-basicauth**、**apicast-configuration-url-secret** を指定し、**--password** パラメーターで 3scale API Management デプロイメントのアクセストークン、テナント名、およびワイルドカードドメインを設定します。

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



注記

TENANT_NAME は、管理 UI が利用可能なルート下の名前です。**'TENANT_NAME'** のデフォルト値は **3scale** です。3scale API Management デプロイメントでカスタム値を使用した場合は、ここでその値を使用する必要があります。

4. `oc new-app` コマンドを使用して、OpenShift クラスター外部の OpenShift クラスターに APIcast をデプロイします。 `--file` オプションで `apicast.yml` ファイルのファイルパスを指定します。

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

5. `apicast` `BACKEND_ENDPOINT_OVERRIDE` 環境変数を URL バックエンドに設定し、続いて 3scale API Management デプロイメントが含まれる OpenShift クラスターのワイルドカードドメインを更新します。

```
oc env dc/apicast --overwrite BACKEND_ENDPOINT_OVERRIDE=https://backend-
<TENANT_NAME>.<WILDCARD_DOMAIN>
```

3.7.3. 他のデプロイメントからの APIcast の接続

他のプラットフォームに APIcast をデプロイした後、3scale API Management OpenShift クラスターの `BACKEND_ENDPOINT_OVERRIDE` 環境変数を設定して、OpenShift 上の 3scale API Management に接続できます。

1. 3scale API Management OpenShift クラスターにログインします。

```
oc login
```

2. `system-app` オブジェクト `BACKEND_ENDPOINT_OVERRIDE` 環境変数を設定します。

- ネイティブインストールを使用している場合：
`BACKEND_ENDPOINT_OVERRIDE=https://backend.<your_openshift_subdomain>
bin/apicast`
- Docker コンテナ環境を使用している場合：`docker run -e
BACKEND_ENDPOINT_OVERRIDE=https://backend.<your_openshift_subdomain>`

3.7.4. Built-in APIcast のデフォルト動作の変更

外部の APIcast デプロイメントでは、APIcast OpenShift [テンプレートのテンプレートパラメーターを変更することで](#)、デフォルトの動作を変更できます。

Embedded APIcast デプロイメントでは、3scale API Management および APIcast は単一のテンプレートからデプロイされます。Embedded APIcast デプロイメントのデフォルト動作を変更する場合は、デプロイ後に環境変数を変更する必要があります。

3.7.5. 内部サービスルートを紹介した、単一 OpenShift クラスター上の複数 APIcast デプロイメントの接続

同じ OpenShift クラスターに複数の APIcast ゲートウェイをデプロイする場合、デフォルトの外部ルート設定ではなく、バックエンドリスナーサービスを介して内部ルートを使用して接続するよう設定できます。

内部サービスルート経由で接続するには、OpenShift SDN プラグインがインストールされている必要があります。接続方法は、インストールされている SDN によって異なります。

`ovs-subnet`

ovs-subnet OpenShift SDN プラグインを使用している場合は、以下の手順を実施して内部ルート経由で接続します。

1. OpenShift クラスターにログインしていない場合はログインします。

```
oc login
```

2. **apicast.yml** ファイルへのパスを指定して **oc new-app** コマンドを入力します。
 - a. **--param** オプションを使用して、**BACKEND_ENDPOINT_OVERRIDE** パラメーターを OpenShift クラスターの 3scale API Management プロジェクトのドメインに設定します。

```
oc new-app -f apicast.yml --param BACKEND_ENDPOINT_OVERRIDE=http://backend-listener.<THREESCALE_PROJECT>.svc.cluster.local:3000
```

ovs-multitenant

'ovs-multitenant' Openshift SDN プラグインを使用している場合は、以下の手順を実施して内部ルート経由で接続します。

1. OpenShift クラスターにログインしていない場合はログインします。

```
oc login
```

2. admin として、pod **-network** および **join-projects** オプションで、両方のプロジェクト間の通信を設定するために、**oadm** コマンドを指定します。

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

3. **apicast.yml** ファイルへのパスを指定して **oc new-app** オプションを入力します。
 - a. **--param** オプションを使用して、**BACKEND_ENDPOINT_OVERRIDE** パラメーターを OpenShift クラスターの 3scale API Management プロジェクトのドメインに設定します。

```
oc new-app -f apicast.yml --param BACKEND_ENDPOINT_OVERRIDE=http://backend-listener.<THREESCALE_PROJECT>.svc.cluster.local:3000
```

詳細情報

Openshift SDN およびプロジェクトネットワークの分離についての情報は、「[OpenShift SDN](#)」を参照してください。

3.8.7.トラブルシューティング

本セクションでは、典型的なインストールの問題と、その問題を解決するためのアドバイスについて説明します。

- [以前のデプロイメントがダーティな永続ボリューム要求を離れる](#)
- [誤って Docker レジストリーからプルされる](#)
- [永続ボリュームがローカルでマウントされている場合の MySQL の権限の問題](#)
- [ロゴまたはイメージをアップロードできない](#)

- [OpenShift でのセキュアなルートの作成](#)
- [3scale とは異なるプロジェクト上の APIcast](#)

3.8.1. 以前のデプロイメントがダーティな永続ボリューム要求を離れる

問題

以前のデプロイメントがダーティな永続ボリューム要求 (PVC) を残そうとするため、MySQL コンテナの起動に失敗する。

原因

OpenShift のプロジェクトを削除しても、それに関連する PVC は消去されない。

解決策

1. `oc get pvc` コマンドを使用してエラーのある MySQL データが含まれる PVC を探します。

```
# 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. OpenShift UI の **cancel deployment** をクリックして、system-mysql Pod のデプロイメントを停止します。
3. MySQL パス以下にあるものすべてを削除し、ボリュームをクリーンアップします。
4. 新たに **system-mysql** のデプロイメントを開始します。

3.8.2. 誤って Docker レジストリーからプルされる

問題

インストール中に以下のエラーが発生する。

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

原因

OpenShift は `docker` コマンドを実行し、コンテナイメージを検索およびプルします。このコマンドは、registry. **access.redhat.com Red Hat** コンテナレジストリーではなく、**docker.io** Docker レジストリーを参照します。

これは、システムに予期せぬバージョンの Docker コンテナ環境が含まれる場合に発生します。

解決方法

[適切なバージョン](#) の Docker コンテナ環境を使用します。

3.8.3. 永続ボリュームがローカルでマウントされている場合の MySQL の権限の問題

問題

system-mysql Pod がクラッシュし、デプロイされないため、それに依存する他のシステムのデプロイメントに失敗する。Pod ログに以下のエラーが記録される。

```
[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 プロセスが不適切なユーザー権限で起動されている。

解決方法

1. 永続ボリュームに使用されるディレクトリーには、root グループの書き込み権限が必要です。MySQL サービスは root グループの別のユーザーとして実行されるため、root ユーザーの読み取り/書き込み権限では不十分です。root ユーザーとして以下のコマンドを実行します。

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

2. 以下のコマンドを実行して、SELinux がアクセスをブロックしないようにします。

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

3.8.4. ログまたはイメージをアップロードできない

問題

ログをアップロードできず、**system-app** ログに以下のエラーが表示される。

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

原因

OpenShift が永続ボリュームに書き込みを行うことができない。

解決方法

OpenShift が永続ボリュームに書き込みを行えるようにします。永続ボリュームのグループ所有者を root グループにし、またグループによる書き込みを可能にする必要があります。

3.8.5. OpenShift でのセキュアなルートの作成

問題

OpenShift で新しいサービスとルートを作成した後に、テストコールが動作しない。curl 経由のダイレクトコールも失敗し、**service not available** が出力される。

原因

3scale はデフォルトで HTTPS ルートが必要で、OpenShift ルートはセキュアではない。

解決方法

OpenShift のルーター設定で **secure route** チェックボックスが選択されていることを確認します。

3.8.6. 3scale 以外のプロジェクト上の APIcast

問題

APIcast のデプロイに失敗する (Pod が青にならない)。以下のエラーがログに記録されます。

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

以下のエラーが Pod に表示されます。

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

原因

シークレットが適切に設定されていない。

解決策

APIcast v3 でシークレットを作成する時に **apicast-configuration-url-secret** を指定します。

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

第4章 オンプレミス型 3SCALE API MANAGEMENT の操作およびスケーリングガイド

4.1. はじめに

本書では、オンプレミス型 Red Hat 3scale AMP 2.4 インストール環境での操作およびスケーリングタスクについて説明します。

4.1.1. 前提条件

[サポート対象バージョンの OpenShift](#) にインストールされた初期設定済みのオンプレミス型 AMP インスタンス

本書は、ノートパソコンやこれに類するエンドユーザー機器上のローカルインストールを対象としていません。

4.1.2. 詳細はこちら

- [『開発者ガイド』の「アプリケーションの正常性」](#)
- [OpenShift のドキュメント](#)

4.2. APICAST の再デプロイ

AMP On-Premises をデプロイし、選択した APICast デプロイメント方法を選択したら、AMP ダッシュボードでシステムの変更をテストし、プロモートすることができます。デフォルトでは、OpenShift 上の APICast デプロイメントでは (組み込み型のデプロイおよび他の OpenShift クラスター上のデプロイの両方で)、AMP UI を介して変更をステージング環境用と実稼働環境用のゲートウェイにパブリッシュできるように設定されています。

APICast を OpenShift に再デプロイします。

1. システムに変更を加えます。
2. UI でステージング環境にデプロイしてテストします。
3. UI で実稼働環境にプロモートします。

デフォルトでは、APICast はプロモートされた更新を 5 分ごとに取得し、パブリッシュします。

Docker コンテナ環境またはネイティブインストールで APICast を使用している場合は、ステージング環境用と実稼働環境用のゲートウェイを設定し、パブリッシュした変更をゲートウェイが取得する頻度を設定する必要があります。APICast ゲートウェイを設定したら、AMP UI で APICast を再デプロイできます。

Docker コンテナ環境またはネイティブインストールに APICast を再デプロイするには、以下を実行します。

1. APICast ゲートウェイを設定し、オンプレミス型 AMP に接続します。
2. システムに変更を加えます。
3. UI でステージング環境にデプロイしてテストします。

4. UI で実稼働環境にプロモートします。

APIcast は、設定された頻度でプロモートされた更新を取得しパブリッシュします。

4.3. APICAST 組み込みワイルドカードルーティング

AMP デプロイメントでサブドメインレベルでワイルドカードドメインルーティングをサポートする組み込み APIcast ゲートウェイ。この機能により、実稼働およびステージングゲートウェイの公開ベース URL のサブドメインに名前を付けることができます。この機能を使用するには、[オンプレミスのインストール時に](#) 有効にする必要があります。



注記

ワイルドカードルーティングをサポートする OpenShift Container Platform バージョンを使用していることを確認します。サポート対象バージョンの詳細は「[Red Hat Fuse でサポートされる構成](#)」を参照してください。

AMP は DNS 機能を提供しないため、指定された公開ベース URL は、デプロイされた OpenShift クラスターの **WILDCARD_DOMAIN** パラメーターで指定された DNS 設定と一致する必要があります。

4.3.1. ワイルドカードの変更

ワイルドカードを変更するには、以下の手順を実行します。

1. AMP にログインします。
2. API ゲートウェイ設定ページに移動します：**APIs → your API → Integration → edit APIcast configuration**
3. 任意の文字列接頭辞を使用してステージング環境および実稼働環境用の公開ベース URL を変更し、以下の要件に従います。
 - API エンドポイントは数字の文字で開始してはいけない

以下は、ドメイン **example.com** 上のステージングゲートウェイの有効なワイルドカードの例です。

```
apiname-staging.example.com
```

詳細情報

ルーティングの詳細は、[OpenShift のドキュメント](#) を参照してください。

4.4. オンプレミス型 AMP のスケールアップ

4.4.1. ストレージのスケールアップ

APIcast デプロイメントの規模が大きくなると、利用可能なストレージの量を増やす必要が生じる可能性があります。ストレージをスケールアップする方法は、永続ストレージに使用しているファイルシステムのタイプによって異なります。

ネットワークファイルシステム (NFS) を使用している場合は、**oc edit pv** コマンドを使用して永続ボリュームをスケールアップできます。

```
oc edit pv <pv_name>
```

他のストレージ手段を使用している場合は、以降のセクションに挙げる方法のいずれかを使用して、永続ボリュームを手動でスケールアップする必要があります。

4.4.1.1. 方法 1: 永続ボリュームのバックアップおよびスワップ

1. 既存の永続ボリューム上のデータをバックアップします。
2. 新しいサイズ要件に合わせて、ターゲット永続ボリュームを作成し、アタッチします。
3. 事前バインド型の永続ボリューム要求を作成し、新しい PVC のサイズと永続ボリュームの名前を指定します。永続ボリューム名には **volumeName** フィールドを使用します。
4. 新しく作成した PV に、バックアップからデータを復元します。
5. 新しい PV の名前でデプロイメント設定を変更します。

```
oc edit dc/system-app
```

6. 新しい PV が設定され正常に機能していることを確認します。
7. 以前の PVC を削除して、それが要求していたリソースを解放します。

4.4.1.2. 方法 2: AMP のバックアップおよび復元

1. 既存の永続ボリューム上のデータをバックアップします。
2. 3scale Pod をシャットダウンします。
3. 新しいサイズ要件に合わせて、ターゲット永続ボリュームを作成し、アタッチします。
4. 新しく作成した PV に、バックアップからデータを復元します。
5. 事前バインド型の永続ボリューム要求を作成します。以下の項目を指定します。
 - a. 新しい PVC のサイズ
 - b. 永続ボリューム名 (**volumeName** フィールドを使用)
6. AMP.yml をデプロイします。
7. 新しい PV が設定され正常に機能していることを確認します。
8. 以前の PVC を削除して、それが要求していたリソースを解放します。

4.4.2. パフォーマンスのスケールアップ

4.4.2.1. オンプレミス型 3scale デプロイメントの設定

デフォルトでは、3scale デプロイメントは Pod ごとに1つのプロセスを実行します。Pod ごとに実行するプロセスを増やすことで、パフォーマンスを向上させることができます。Red Hat は、各ノードのコアごとに1つまたは2つのプロセスを実行することを推奨します。

Pod にプロセスを追加するには、以下の手順を実行します。

1. OpenShift クラスターにログインします。

```
oc login
```

2. 3scale プロジェクトに切り替えます。

```
oc project <project_name>
```

3. 適切な環境変数に、希望する Pod ごとのプロセス数を設定します。

- a. APIcast Pod: **APICAST_WORKERS** (Red Hat は、APIcast が APIcast Pod で利用できる CPU の数によってワーカーの数を判断できるように、この環境変数を未設定のままにすることを推奨します)

- b. バックエンド Pod: **PUMA_WORKERS**

- c. システム Pod: **UNICORN_WORKERS**

```
oc env dc/apicast --overwrite APICAST_WORKERS=<number_of_processes>
```

```
oc env dc/backend --overwrite PUMA_WORKERS=<number_of_processes>
```

```
oc env dc/system-app --overwrite UNICORN_WORKERS=<number_of_processes>
```

4.4.2.2. 垂直および水平のハードウェアスケーリング

リソースを追加することで、OpenShift 上の AMP デプロイメントのパフォーマンスを高めることができます。コンピュートノードを Pod として OpenShift クラスターに追加すること (水平スケーリング) や、既存のコンピュートノードに割り当てるリソースを増やすこと (垂直スケーリング) ことができます。

水平スケーリング

コンピュートノードを Pod として OpenShift に追加することができます。追加のコンピュートノードがクラスター内の既存ノードと一致する場合には、環境変数を再設定する必要はありません。

垂直スケーリング

既存のコンピュートノードに割り当てるリソースを増やすことができます。割り当てるリソースを増やす場合は、追加のプロセスを Pod に追加してパフォーマンスを高める必要があります。



注記

Red Hat は、3scale デプロイメントにおいて、仕様や設定の異なるコンピュートノードを混在させることは推奨していません。

4.4.2.3. ルーターのスケールアップ

トラフィックの増加に応じて、OCP ルーターがリクエストを適切に処理できるようにする必要があります。ルーターがリクエストのスループットを制限している場合には、ルーターノードをスケールアップする必要があります。

4.4.2.4. 詳細はこちら

- タスクのスケーリング、ハードウェアコンピュートノードの OpenShift への追加
- コンピュートノードの追加
- ルーター

4.5. 操作のトラブルシューティング

4.5.1. ログへのアクセス

各コンポーネントのデプロイメント設定には、アクセスと例外のログが含まれます。デプロイメントで問題が発生した場合には、これらのログで詳細を確認してください。

3scale のログにアクセスするには、以下の手順に従います。

1. ログを必要とする Pod の ID を確認します。

```
oc get pods
```

2. **oc logs** と選択した Pod の ID を入力します。

```
oc logs <pod>
```

システム Pod にはコンテナが 2 つあり、それぞれに別個のログがあります。コンテナのログにアクセスするには、**--container** パラメーターで **system-provider** および **system-developer** を指定します。

```
oc logs <pod> --container=system-provider  
oc logs <pod> --container=system-developer
```

4.5.2. ジョブキュー

ジョブキューには、**system-sidekiq** Pod から送られる情報のログが含まれます。これらのログを使用して、クラスターがデータを処理しているかどうかを確認します。OpenShift CLI を使用してログを照会することができます。

```
oc get jobs
```

```
oc logs <job>
```

第5章 3SCALE API MANAGEMENT HIGH AVAILABILITY AND EVALUATION

5.1. はじめに

ここでは、オンプレミス型 Red Hat 3scale API Management 2.4 [インストール環境で使用される高可用性テンプレート](#)および[評価用テンプレート](#)について説明します。

5.2. 前提条件

- 高可用性テンプレートおよび評価用テンプレートの要素をデプロイできる OpenShift クラスターが必要です。

5.3. 高可用性テンプレート

高可用性 (HA) テンプレートを使用すると、重要なデータベースの HA を設定できます。

5.3.1. 前提条件

- HA テンプレートをデプロイする前に、外部データベースをデプロイおよび設定し、負荷分散されたエンドポイントで HA を設定する必要があります。

5.3.2. HA テンプレートの使用

高可用性のために **amp-ha-tech-preview.yml** という名前のテンプレートを使用すると、OpenShift 外部に重要なデータベースをデプロイできます。ただし、以下は除外されます。

- Memcached
- Sphinx
- Zync

標準の **amp.yml** テンプレートと **amp-ha-tech-preview.yml** には、以下の違いがあります。

- 以下の要素が削除されています。
 - backend-redis およびその関連コンポーネント
 - system-redis およびその関連コンポーネント
 - system-mysql およびその関連コンポーネント
 - Redis および MySQL 関連の ConfigMaps
 - MYSQL_IMAGE、REDIS_IMAGE、MYSQL_USER、MYSQL_ROOT_PASSWORD パラメーター
- デフォルトで、データベースではない **DeploymentConfig** オブジェクトタイプのレプリカの数が増加されます。
- 以下の必須パラメーターが追加されているため、外部データベースの場所を制御できます。
 - BACKEND_REDIS_STORAGE_ENDPOINT

- BACKEND_REDIS_QUEUES_ENDPOINT
- SYSTEM_REDIS_URL
- APICAST_STAGING_REDIS_URL
- APICAST_PRODUCTION_REDIS_URL
- SYSTEM_DATABASE_URL

amp-ha-tech-preview.yml を使用する場合、新たに追加された必須パラメーターによりクラスター外のデータベース接続を設定する必要があります (ただし、永続的なデータが含まれない **system-memcache**、**zync-database**、および **system-sphinx** は除外)。エンドポイントには、クレデンシャルを含む、データベースの負荷分散用接続文字列が必要です。また、データベースではないデプロイメントについては、アプリケーションレベルでの冗長性を確保するためにデフォルトで Pod レプリカの数が増えています。

5.4. 評価用テンプレート

評価の目的で、リソースのリクエストや制限のない 3scale 環境をデプロイする **amp-eval-tech-preview.yml** という名前のテンプレートが提供されています。

標準の **amp.yml** テンプレートとの唯一の機能的な違いは、リソースの制限とリクエストが削除されたことです。そのため、このバージョンでは CPU およびメモリーレベルでハードウェアの最低要件が Pod で削除されました。このテンプレートは、指定のハードウェアリソースを使用して、可能な限りコンポーネントをデプロイしようとするため、評価、テスト、および開発のみの使用を目的としています。

第6章 3SCALE の REDIS 高可用性 (HA) サポート



注記

3scale における Redis 高可用性 (HA) のサポートには、既知の問題があります。詳細は、Red Hat 3scale API Management 2.4 リリースノート (セクション 1.5) を参照してください。 [リリースノートの既知の問題](#)。

6.1. はじめに

高可用性 (HA) は、OpenShift Container Platform (OCP) によりほとんどのコンポーネントで提供されます。詳細は、[OpenShift Container Platform 3.11 第 30 章](#) を参照してください。「High Availability」を参照してください。

3scale では HA のデータベースコンポーネントに以下が含まれます。

- **system-redis**: 3scale API Management のバックグラウンドジョブの一時ストレージを提供し、**system-app** Pod の Ruby プロセスのメッセージバスとしても使用されます。
- **backend-redis**: 統計ストレージおよび一時ジョブストレージに使用されます。



注記

system-redis および **backend-redis** は、どちらも **Redis Cluster** (open-source または Redis Labs) に置き換えることができます。

以下の **env 変数** は **system-** (**app**、**sidekiq**、**sphinx**) デプロイメント設定に設定できませんが、Redis Enterprise の要件のみになります。

- **MESSAGE_BUS_REDIS_URL** (redis URL)
- **REDIS_NAMESPACE** (namespace Sidekiq の Redis キーへの短い文字列)
- **MESSAGE_BUS_REDIS_NAMESPACE** (namespace System メッセージバスの Redis キーへの短い文字列)

Redis Pod が使えなくなったり、OCP によって強制的に終了されたりすると、新しい Pod が自動作成されます。データは永続ストレージから復元されるため、Pod は動作し続けます。このような場合、新しい Pod が起動するまで短いダウンタイムが発生します。これは、Redis がマルチマスター設定をサポートしないという制限によるものです。Redis をデプロイしたすべてのノードに Redis イメージを事前に読み込むと、ダウンタイムを削減することができ、Pod の再起動を迅速に行うことができます。

6.2. ゼロダウンタイムのための REDIS 設定

ダウンタイムをゼロにすることが必須の要件であれば、Redis を OCP 外部に設定する必要があります。3scale Pod の設定オプションを使用してこの設定を行うには、いくつかの方法があります。

- 独自の自己管理型 Redis を設定する
- Redis Sentinel を使用する (『[Redis Sentinel Documentation](#)』を参照)
- サービスとして提供される Redis:
例:

- Amazon ElastiCache
- Redis Labs



注記

Red Hat は上記のサービスにサポートを提供しません。このようなサービスの言及は、Red Hat による製品やサービスの推奨を意味するものではありません。Red Hat は、Red Hat 外部のコンテンツを使用 (または依存) して発生した損失や費用の責任を負いません。

6.3. 3SCALE 用バックエンドコンポーネントの設定

3scale API Management 2.4 では、バックエンドコンポーネントの Redis HA (フェイルオーバー) を設定できます。以下のデプロイメント設定では、**backend-cron**、**backend-listener**、および **backend-worker** の環境変数として設定することができます。

- **CONFIG_REDIS_SENTINEL_HOSTS** および **CONFIG_QUEUES_SENTINEL_HOSTS**:
メイン統計データベースに対する Sentinel ホストのコンマ区切りリストおよび Resque バックグラウンドジョブのデータベース



注記

値は name:value <host>:<port> の形式にする必要があります (例: **host1:26379**、**host2:26379**、または **host3:26379**)。

- **CONFIG_REDIS_SENTINEL_ROLE** および **CONFIG_QUEUES_SENTINEL_ROLE**:
各 Sentinels グループのロールで、**master** または **slave** のいずれか。現時点でサポートされるのは **master** (デフォルト) だけです。

これにより、**CONFIG_REDIS_PROXY** と **CONFIG_QUEUES_MASTER_NAME** の値は、特定のサーバーではなく Sentinel グループ名の意味を取ります。

- Sentinel ホストが設定されていない場合には、環境変数 **CONFIG_REDIS_PROXY** および **CONFIG_QUEUES_MASTER_NAME** は URL を使用し、パスワードで保護されたデータベースをサポートします (例: **CONFIG_REDIS_PROXY=redis://user:password@server:port/database**)。
 - その後、パスワードで保護された Redis インスタンスとの間で接続が確立されます。
- Sentinel ホストを設定する場合は、パスワードを Sentinel 設定で設定し、代わりに Sentinel グループ名 (**CONFIG_REDIS_PROXY=master_group**) を使用する必要があります。

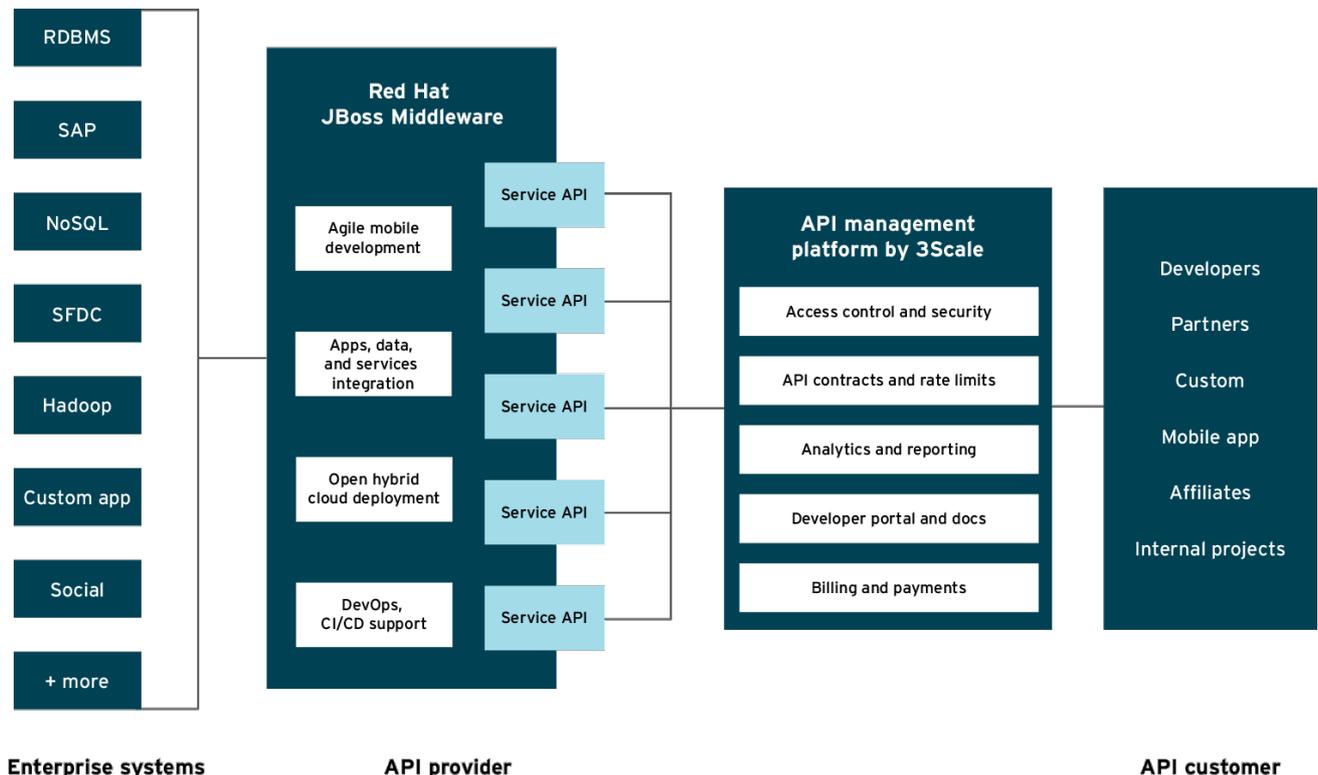
第7章 「HOW TO DEPLOY A FULL-STACK API SOLUTION WITH FUSE, 3SCALE, AND OPENSIFT」

本チュートリアルでは、OpenShift および 3scale API Management Platform 向けの Red Hat JBoss xPaaS を使用して、フルスタック API ソリューション（API 設計、開発、ホスト、アクセス制御、収益化など）を取得する方法について説明します。

このチュートリアルは、Red Hat と 3scale 間のコラボレーションに基づいて、[フルスタック API ソリューション](#)を提供します。このソリューションには、[Red Hat JBoss xPaaS for OpenShift](#) での API の設計、開発、ホストが含まれており、3scale API Management Platform と完全な制御、可視性、および収益化機能が提供されます。

API 自体を Red Hat JBoss xPaaS for OpenShift にデプロイすることができます。OpenShift は、クラウドおよびオンプレミス（Red Hat 部分）でホストされることができます。API 管理（3scale 部分）は、3scale [APIcast](#) または OpenShift を使用して Amazon Web Services(AWS)でホストすることができます。これにより、デプロイメントの柔軟性を最大限にするために、さまざまな設定オプションを利用できます。

以下の図は、この結合ソリューションの主要な要素をまとめています。エンタープライズバックエンドシステム、ミドルウェア、API 管理、および API 顧客を含む統合チェーン全体が表示されます。



JB0095

具体的なサポートに関する質問は、[サポートにご連絡ください](#)。

本チュートリアルでは、ステップごとに3つの異なるデプロイメントシナリオステップを以下に示します。

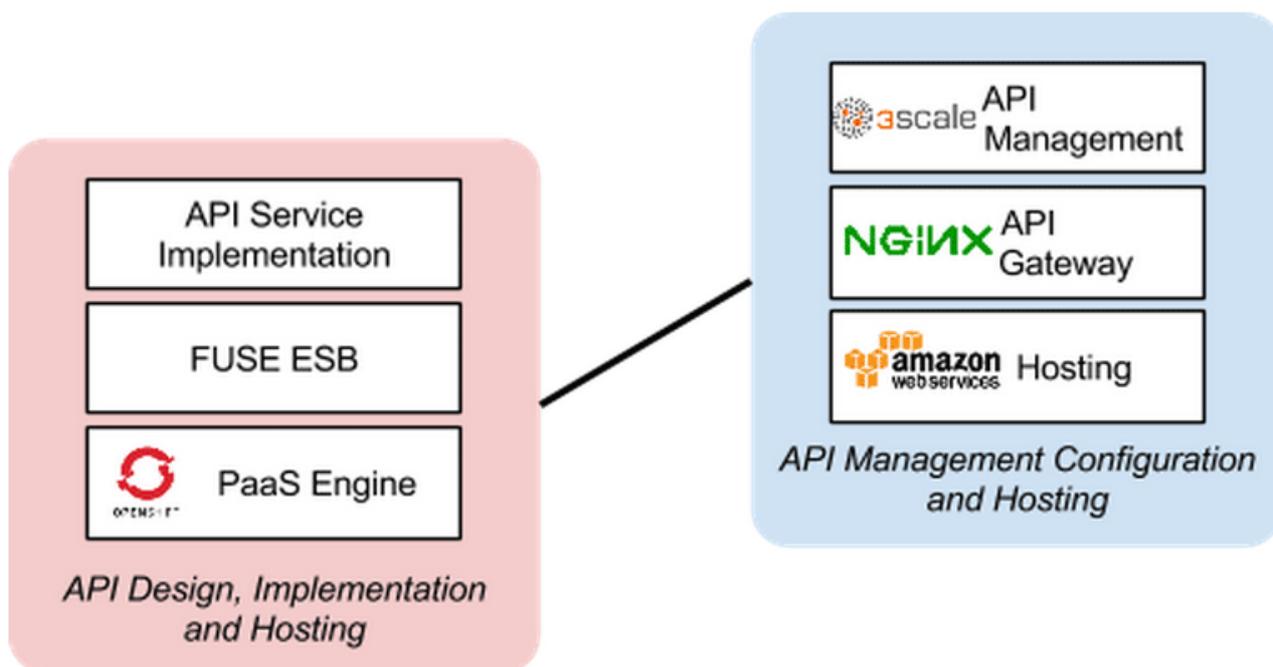
1. シナリオ 1: API が含まれる [Fuse on OpenShift](#) アプリケーション。API は、3scale [AMI](#) を使用して Amazon Web Services(AWS)でホストされる [API ゲートウェイ](#)で 3scale により管理されません。

2. シナリオ 2: API が含まれる Fuse on OpenShift アプリケーション。API は [APIcast](#) でホストされている [API ゲートウェイ](#) を使用して 3scale により管理されます (APIcast (3scale のクラウドホスト型 API ゲートウェイ))。
3. シナリオ 3: API が含まれる Fuse on OpenShift アプリケーション。API は [OpenShift](#) でホストされる [API ゲートウェイ](#) を使用して 3scale により管理されます。

このチュートリアルでは、以下の 4 つの部分に分類されます。

- [パート 1](#): API の設計および実装を行う [Fuse on OpenShift](#) の設定
- [パート 2](#): 3scale API Management の設定
- [パート 3](#): API サービスの統合
- [パート 4](#): API および API 管理のテスト

以下の図は、この設定内のさまざまな部分のロールを示しています。

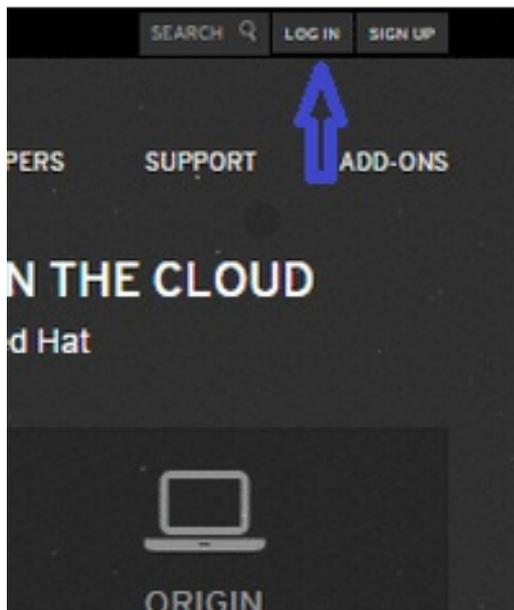


7.1. パート 1: FUSE ON OPENSIFT の設定

管理する API が含まれる [Fuse on OpenShift](#) アプリケーションを作成します。Fuse 6.1 に含まれる REST クイックスタートを使用します。小さいギアを使用すると、メモリーエラーやパフォーマンスに悪影響を与えるため、メディアや大容量のギアが必要になります。

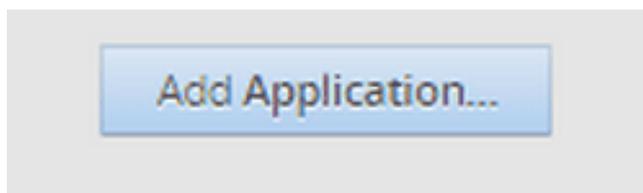
7.1.1. ステップ 1

OpenShift オンラインアカウントにログインします。OpenShift オンラインアカウントにサインアップしていない場合は、このアカウントにサインアップしてください。



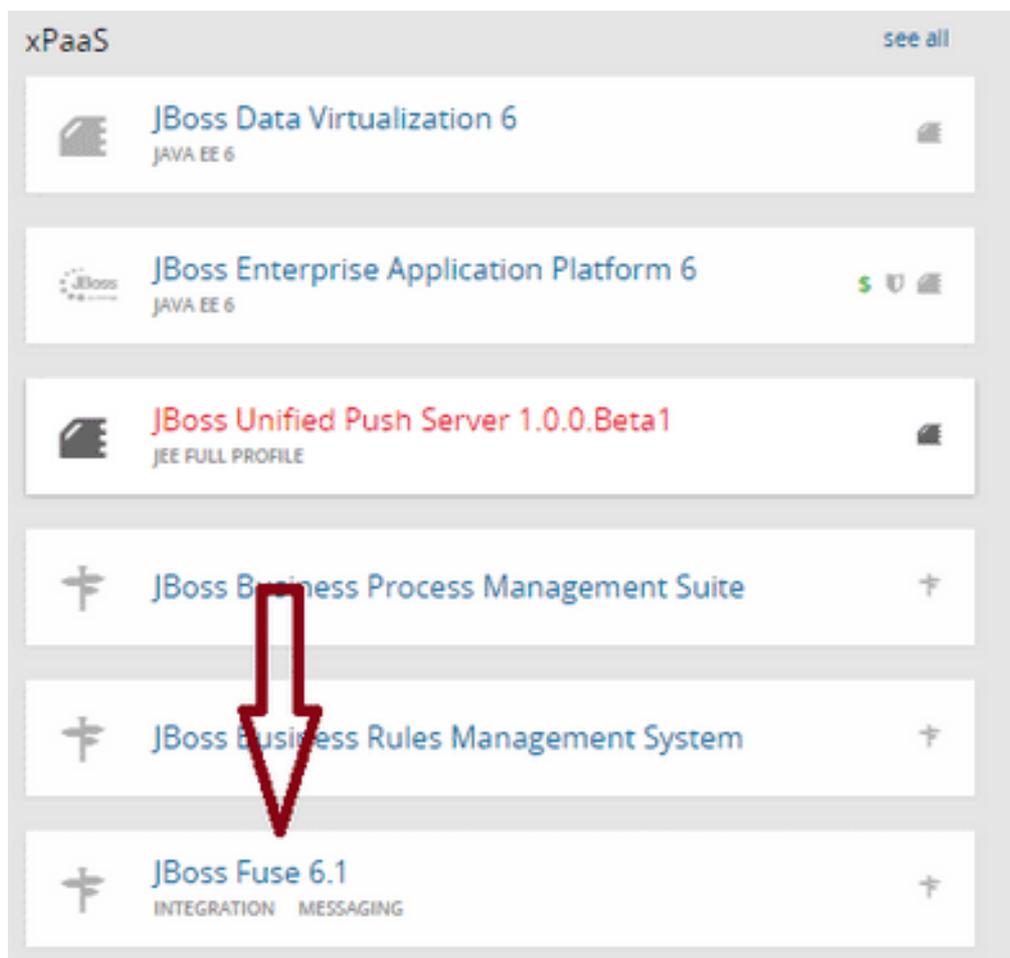
7.1.2. ステップ 2

サインイン後に「add application」ボタンをクリックします。



7.1.3. ステップ 3

xPaaS で、アプリケーションの Fuse タイプを選択します。



7.1.4. ステップ 4

これでアプリケーションが設定されます。「restapitest」など、アプリケーションに表示するサブドメインを入力します。これにより、「appname-domain.rhcloud.com」の形式の完全な URL が提供されます（以下の例で「restapitest-ossmentor.rhcloud.com」）。Fuse カートリッジに必要なギアサイズを medium または large に変更します。これで「Create application」をクリックします。

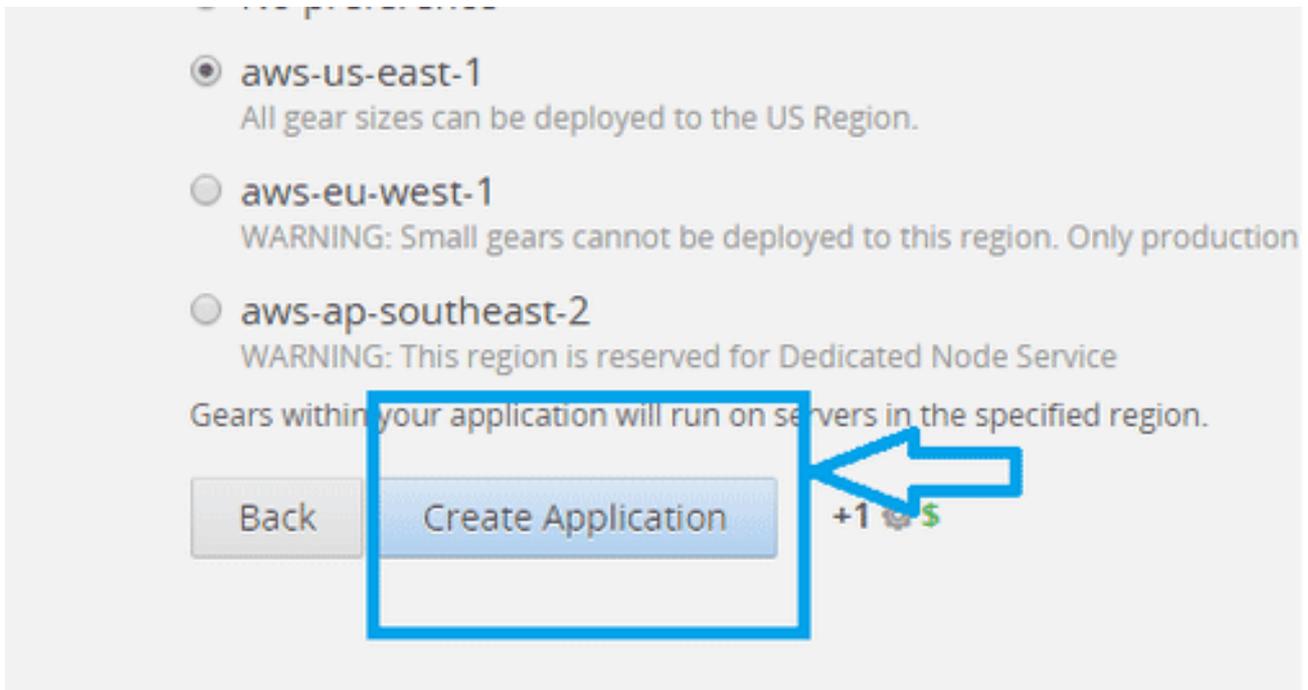
The screenshot shows the 'Configure the application' step (step 2) in the OpenShift console. The interface is divided into several sections for configuring the application:

- Based On:** JBoss Fuse 6.1 Quickstart. Includes a description of the technology, a note about gear size recommendations, and links for 'Learn more' and 'OpenShift maintained'. A warning states 'Does not receive automatic security updates'.
- Public URL:** A text input field containing 'http:// restapitest -ossmentor.rncloud.com'. A note explains that OpenShift will automatically register this domain name.
- Source Code:** Two input fields for 'Optional URL to a Git repository' and 'Branch/tag'. A note explains that OpenShift will create a Git repository if none is provided.
- Gears:** A dropdown menu set to 'medium'. A note explains that gears are application containers and that the 'medium' size provides plenty of resources.
- Cartridges:** A dropdown menu set to 'manifest.yml'. A note explains that applications are composed of cartridges and that downloaded cartridges do not receive updates automatically.
- Scaling:** A dropdown menu set to 'No scaling'. A note explains that OpenShift will automatically route web requests and set up a load balancer if scaling is allowed.
- Region:** Radio buttons for 'No preference', 'aws-us-east-1' (selected), 'aws-eu-west-1', and 'aws-ap-southeast-2'. Each region has a warning note. A note at the bottom states 'Gears within your application will run on servers in the specified region.'

At the bottom of the form, there are three buttons: 'Back', 'Create Application', and a cost indicator '+1 \$'.

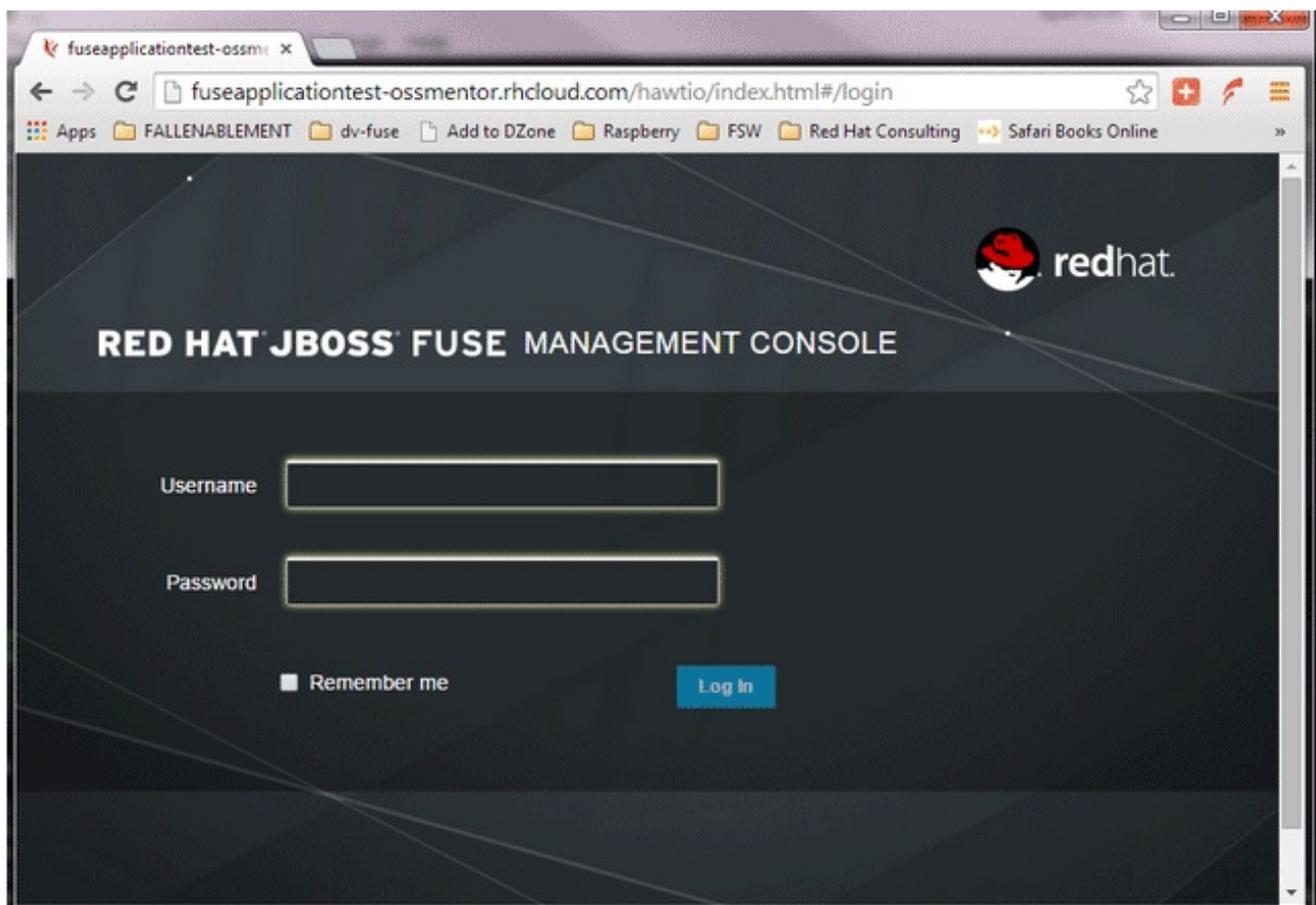
7.1.5. ステップ 5

「Create application」をクリックします。



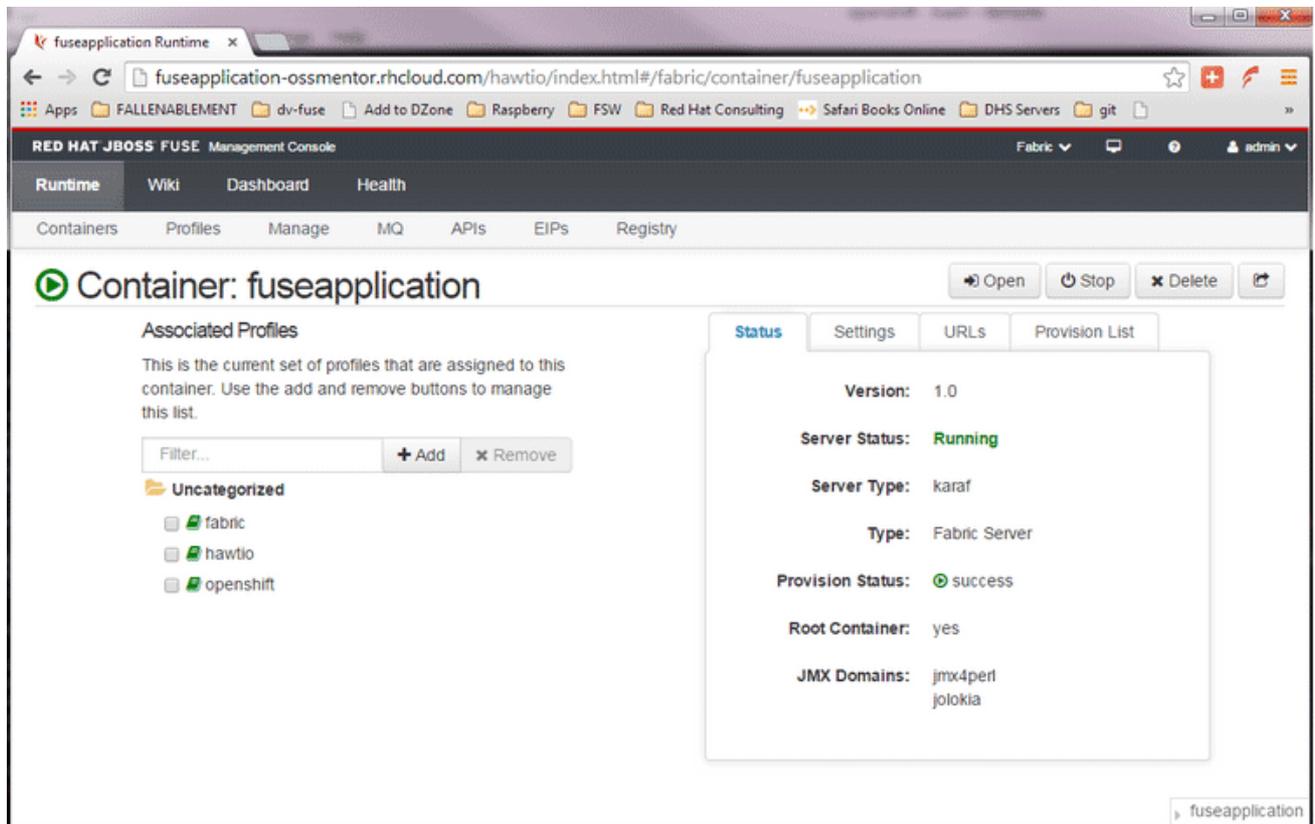
7.1.6. ステップ 6

アプリケーション hawtio コンソールを参照し、サインインします。



7.1.7. ステップ 7

ログイン後に「runtime」タブおよびコンテナをクリックし、REST API の例を追加します。

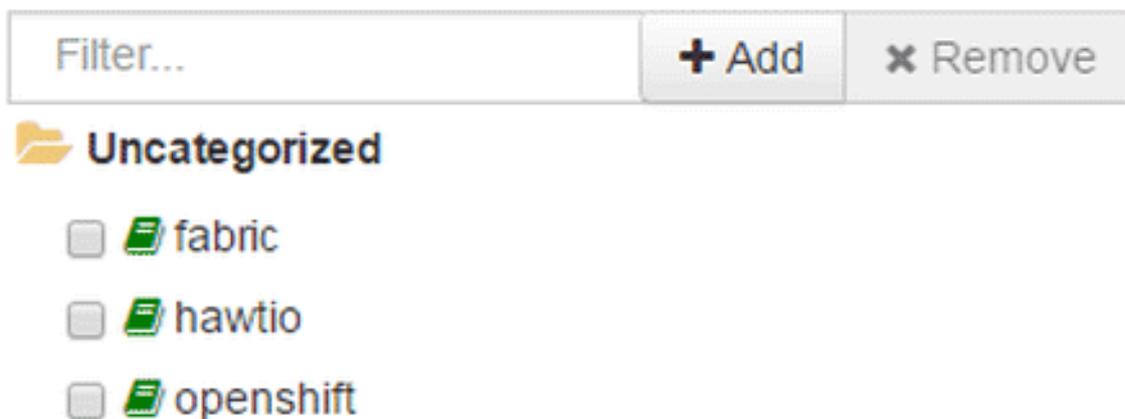


7.1.8. ステップ 8

「プロファイルの追加」ボタンをクリックします。

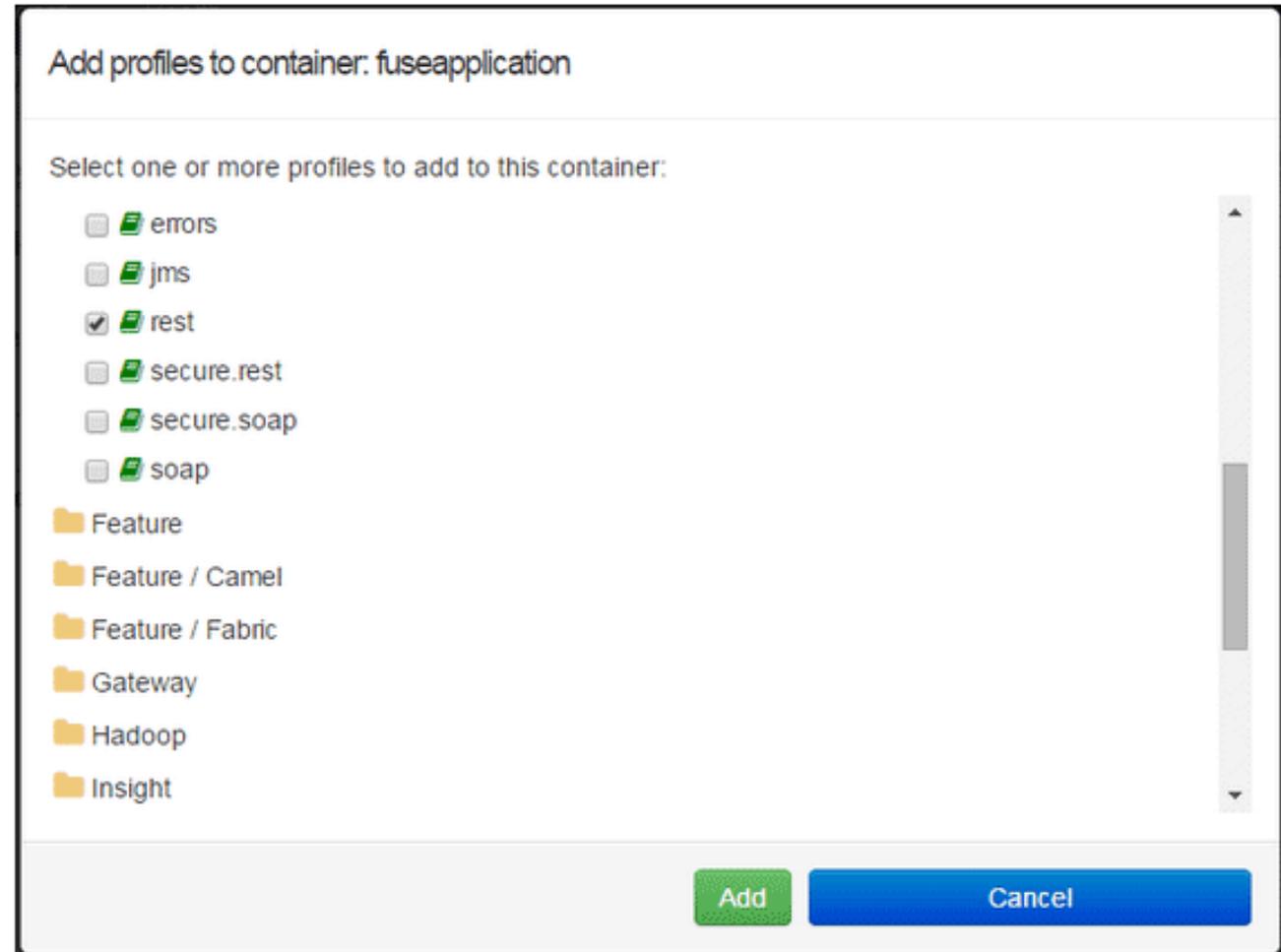
Associated Profiles

This is the current set of profiles that are assigned to this container. Use the add and remove buttons to manage this list.



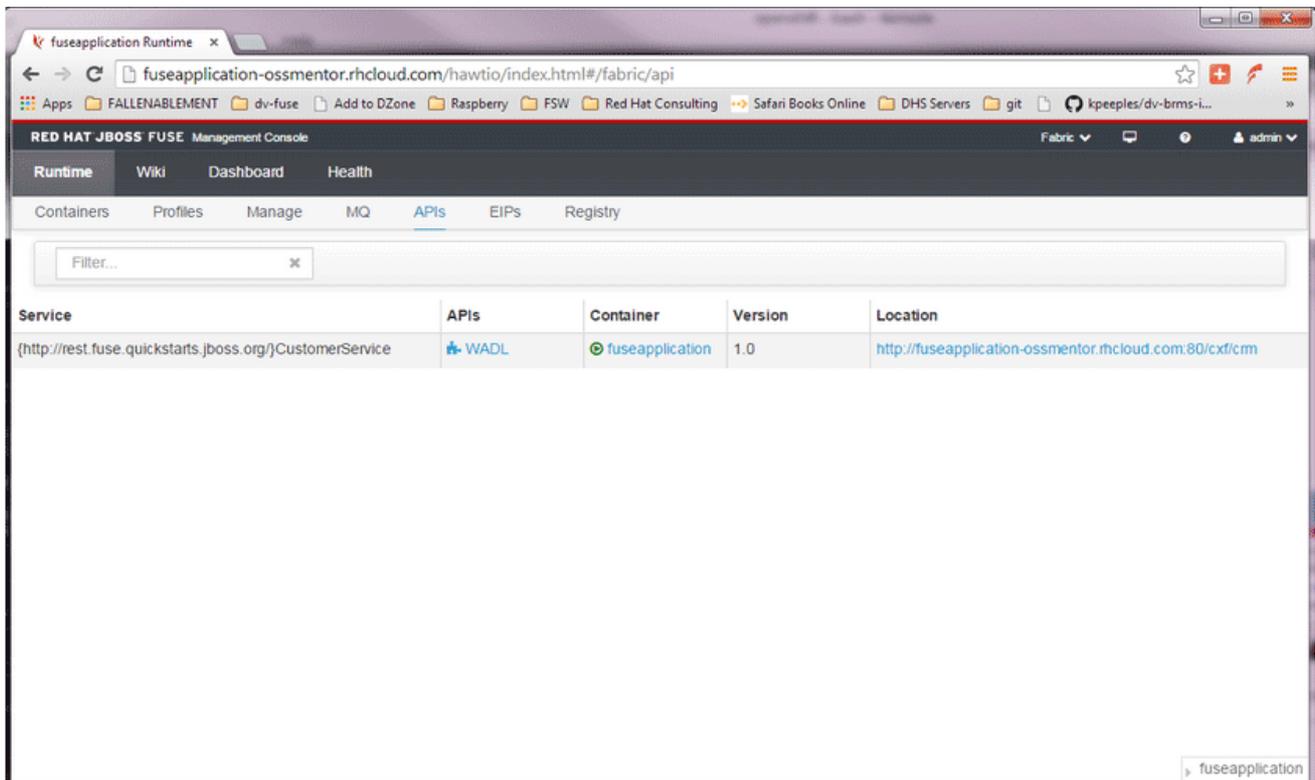
7.1.9. ステップ 9

examples/quickstarts まで下方方向にスクロールし、「REST」チェックボックスを選択してから「add」をクリックします。REST プロファイルが、コンテナ関連のプロファイルページに表示されるはずで



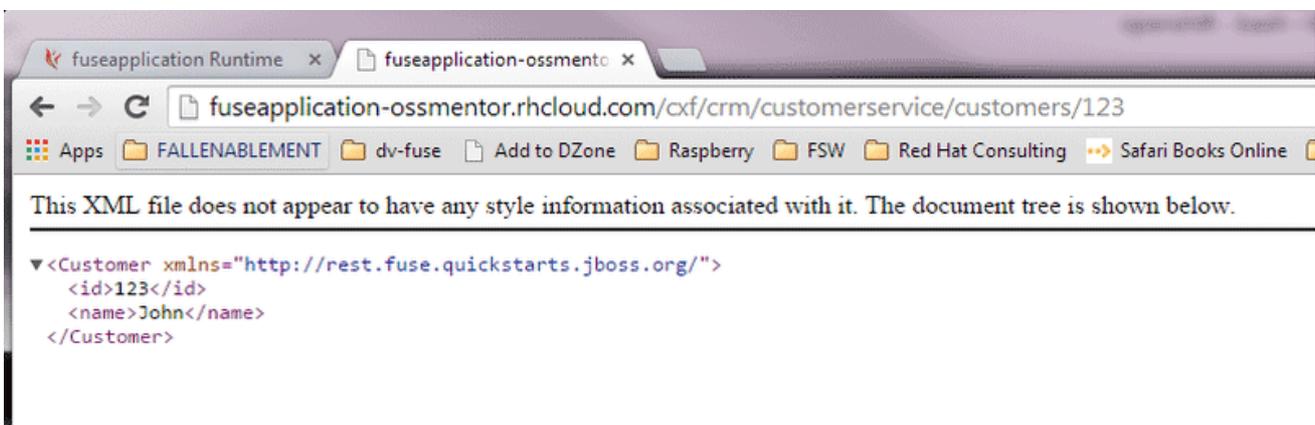
7.1.10. ステップ 10

runtime/APIs タブをクリックして REST API プロファイルを確認します。



7.1.11. ステップ 11

REST API が動作していることを確認します。カスタマー 123 を参照します。これは、XML 形式で ID と名前を返します。



7.2. パート 2: 3SCALE API MANAGEMENT の設定

3scale API Management を使用して Part 1 で作成した API を保護するには、まず以下を行う必要があります。これは、後で提示した 3 つのシナリオに従ってデプロイされます。

OpenShift に API を設定したら、これを 3scale に設定して、アクセス制御と使用状況の監視のための管理レイヤーを提供することができます。

7.2.1. ステップ 1

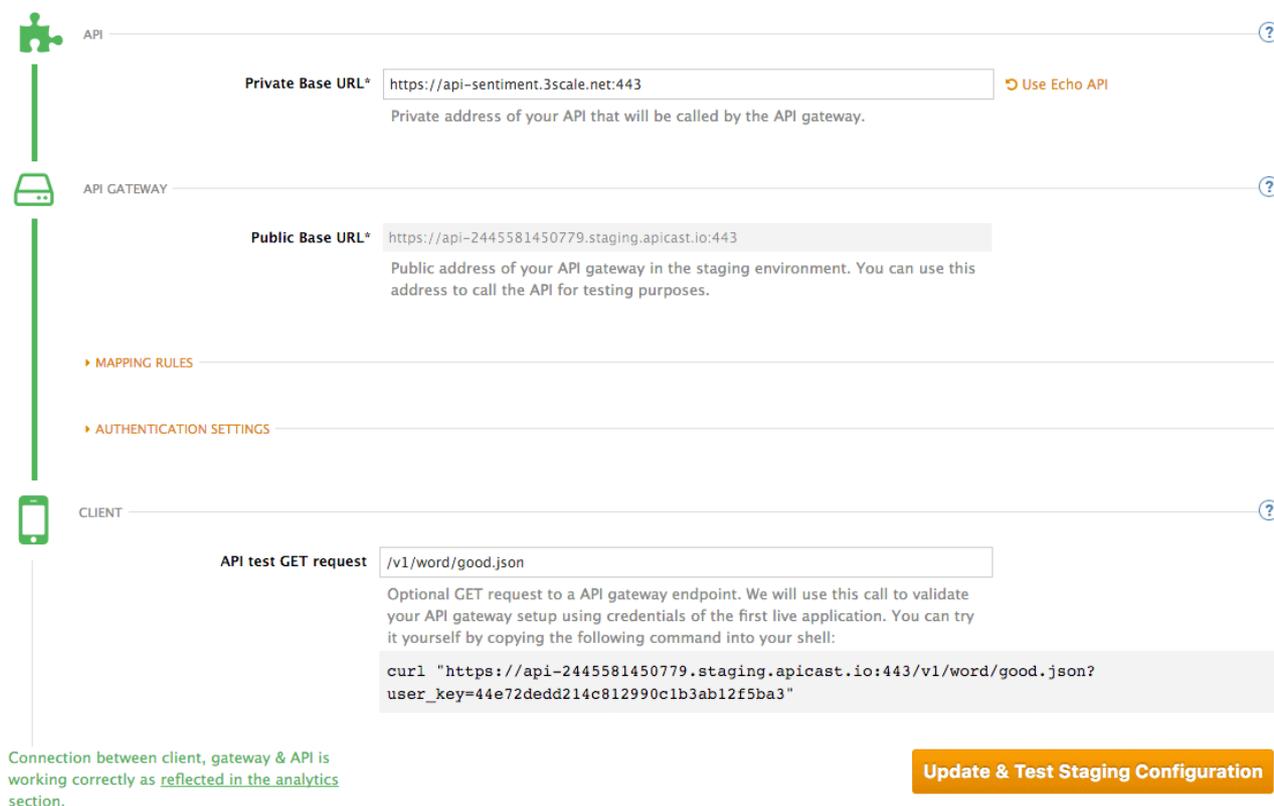
3scale アカウントにログインします。まだ 3scale アカウントをお持ちでない場合は、www.3scale.net でサインアップすることができます。アカウントへの初回ログイン時に、ウィザードに従って API と 3scale の統合に関する基本事項を確認します。

7.2.2. ステップ 2

[your_API_name] > Integration > Configurationの順に移動し、先ほど作成した OpenShift で Fuse アプリケーションのパブリック URL を入力します（例： "restapitest-ossmentor.rhcloud.com"、**Test** をクリックします。これにより、ステージング環境で 3scale API ゲートウェイに対して設定をテストします。ステージング環境用 API ゲートウェイを使用すると、プロキシー設定を AWS にデプロイする前に、3scale の設定をテストすることができます。

Staging: 3scale-hosted to configure & test your integration [documentation](#)

[deployed](#) | [deployment history](#)



API

Private Base URL* [Use Echo API](#)
Private address of your API that will be called by the API gateway.

API GATEWAY

Public Base URL*
Public address of your API gateway in the staging environment. You can use this address to call the API for testing purposes.

▶ MAPPING RULES

▶ AUTHENTICATION SETTINGS

CLIENT

API test GET request
Optional GET request to a API gateway endpoint. We will use this call to validate your API gateway setup using credentials of the first live application. You can try it yourself by copying the following command into your shell:

```
curl "https://api-2445581450779.staging.apicast.io:443/v1/word/good.json?user_key=44e72dedd214c812990c1b3ab12f5ba3"
```

Connection between client, gateway & API is working correctly as [reflected in the analytics section](#).

[Update & Test Staging Configuration](#)

7.2.3. ステップ 3

次の手順では、監視および流量制御を設定する API メソッドを設定します。これには、[your_API_name] > Integration > Methods & Metrics の順に移動し、「New method」をクリックします。

メソッドの作成に関する詳細は、[API 定義チュートリアル](#) を参照してください。

7.2.4. ステップ 4

アプリケーションプランで監視および制御するすべてのメソッドを取得したら、これらを API のエンドポイントの実際の HTTP メソッドにマッピングする必要があります。インテグレーションページに戻り、「マッピングルール」セクションを展開します。

アプリケーションプランで作成した各メソッドにマッピングルールを作成します。

これが完了すると、マッピングルールは以下のようになります。

▼ MAPPING RULES



Verb	Pattern	+	Metric or Method (Define)
GET	/v1/words/{word}.json	1	get_word
GET	/v1/sentences/{sentence}.json	1	get_sente
POST	/v1/words/{word}.json	1	set_word

[+ Add Mapping Rule](#)

マッピングルールの詳細については、「マッピングルール」のチュートリアルを参照してください。

7.2.5. ステップ 5

「更新およびテスト」をクリックして設定を保存およびテストしたら、AWS で API ゲートウェイを設定できるように、設定ファイルのセットをダウンロードする準備が整います。API ゲートウェイでは、[nginx](#) という高パフォーマンスのオープンソースプロキシを使用する必要があります。nginx に必要な設定ファイルは、同じ統合ページにあり、「実稼働」セクションまでスクロールします。

Production: On-premises Gateway

To deploy an on-premises API gateway, add the Public Base URL of your API, download the Nginx Config files and [follow the documentation](#) to install in your servers.

API

Private Base URL

API GATEWAY

Public Base URL

Public address of your API gateway in the production environment. This is used to customize the server_name directive in the Nginx Config file which will otherwise be set to the variable \$hostname.

[Update Production Configuration](#)

[Download the Nginx Config files](#)

次のセクションでは、さまざまなホストシナリオを説明します。

7.3. パート 3: API サービスの統合

3scale で API サービスを統合する方法は複数あります。ニーズに最も適した [デプロイメントオプション](#) を選択します。

7.4. パート 4: API および API MANAGEMENT のテスト

API と API の正しい機能をテストすることは、選択したシナリオから独立しています。お気に入りの REST クライアントを使用し、以下のコマンドを実行します。

7.4.1. ステップ 1

id 123 の顧客インスタンスを取得します。

```
http://54.149.46.234/cxf/crm/customerservice/customers/123?
user_key=b9871b41027002e68ca061faeb2f972b
```

The screenshot shows a REST client interface with the following details:

- URL:** `http://54.149.46.234/cxf/crm/customerservice/customers/123?user_key=b9871b41027002e68ca061faeb2f972b`
- Method:** GET
- Status:** 200 OK (Loading time: 358 ms)
- Request headers:**
 - User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/39.0.2171.95 Safari/537.36
 - Content-Type: text/plain; charset=utf-8
 - Accept: *
 - Accept-Encoding: gzip, deflate, sdch
 - Accept-Language: en-US,en;q=0.8
- Response headers:**
 - Server: ngx_openresty/1.2.8.6
 - Date: Mon, 22 Dec 2014 18:16:08 GMT
 - Content-Type: application/xml
 - Content-Length: 148
 - Connection: keep-alive
 - Vary: Accept-Encoding
 - Content-Encoding: gzip
 - Accept-Ranges: none
- Response (XML):**

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<Customer>
  <id>123</id>
  <name>John</name>
</Customer>
```

7.4.2. ステップ 2

カスタマーを作成します。

```
http://54.149.46.234/cxf/crm/customerservice/customers?
user_key=b9871b41027002e68ca061faeb2f972b
```

The screenshot shows a REST client interface with the following details:

- URL:** `http://54.149.46.234/cxf/crm/customerservice/customers?user_key=b9871b41027002e68ca061faeb2f972b`
- Method:** POST
- Status:** 403 Forbidden (Loading time: 209 ms)
- Request headers:**
 - User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/39.0.2171.95 Safari/537.36
 - Origin: chrome-extension://hgntoofodffnpgcaikdofbjebo
 - Content-Type: text/xml
 - Accept: *
 - Accept-Encoding: gzip, deflate
 - Accept-Language: en-US,en;q=0.8
- Request Payload (XML):**

```
<Customer xmlns="http://rest.fuse.qubis.com/1.0/boas.org/">
  <name>John</name>
</Customer>
```
- Response headers:**
 - Server: ngx_openresty/1.2.8.6
 - Date: Mon, 22 Dec 2014 18:21:27 GMT
 - Content-Type: text/plain; charset=ascii
 - Transfer-Encoding: chunked
 - Connection: keep-alive
- Response:** Authentication parameters missing

7.4.3. ステップ 3

ID 123 で顧客インスタンスを更新します。

http://54.149.46.234/cxf/crm/customerservice/customers?
user_key=b9871b41027002e68ca061faeb2f972b

The screenshot shows a REST client interface with the following details:

- URL:** http://54.149.46.234/cxf/crm/customerservice/customers?user_key=b9871b41027002e68ca061faeb2f972b
- Method:** GET
- Headers:** Content-Type: text/xml
- Payload:**

```
<Customer xmlns="http://rest.fuse.quickstarts.jboss.org/">
  <name>Harry</name>
  <id>123</id>
</Customer>
```
- Status:** 403 Forbidden (Loading time: 200 ms)
- Request Headers:** User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; AppleWebkit/537.36 (KHTML, like Gecko) Chrome/39.0.2171.95 Safari/537.36; Origin: chrome-extension://hgmbodoffphtfgaalkitfzfojao; Content-Type: text/xml; Accept: */*; Accept-Encoding: gzip, deflate, sdch; Accept-Language: en-US,en;q=0.8
- Response Headers:** Server: nghttp/1.2.8.6; Date: Mon, 22 Dec 2014 18:24:03 GMT; Content-Type: text/plain; charset=us-ascii; Transfer-Encoding: chunked; Connection: keep-alive

7.4.4. ステップ 4

id 123 の顧客インスタンスを削除します。

http://54.149.46.234/cxf/crm/customerservice/customers/123?
user_key=b9871b41027002e68ca061faeb2f972b

The screenshot shows a REST client interface with the following details:

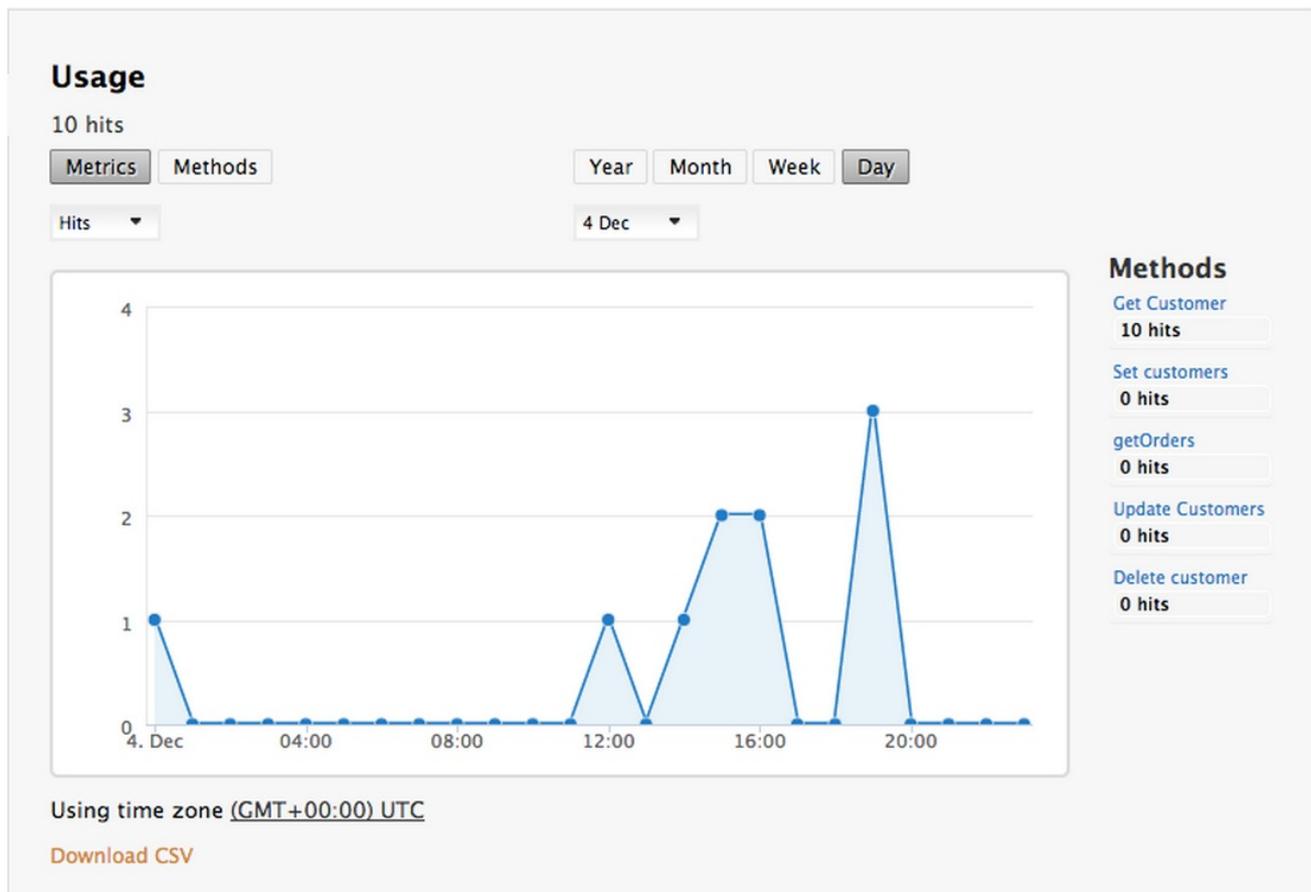
- URL:** http://54.149.46.234/cxf/crm/customerservice/customers/123?user_key=b9871b41027002e68ca061faeb2f972b
- Method:** DELETE
- Payload:**

```
<Customer xmlns="http://rest.fuse.quickstarts.jboss.org/">
  <name>Harry</name>
  <id>123</id>
</Customer>
```
- Status:** 403 Forbidden (Loading time: 211 ms)
- Request Headers:** User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; AppleWebkit/537.36 (KHTML, like Gecko) Chrome/39.0.2171.95 Safari/537.36; Origin: chrome-extension://hgmbodoffphtfgaalkitfzfojao; Content-Type: application/x-www-form-urlencoded; Accept: */*; Accept-Encoding: gzip, deflate, sdch; Accept-Language: en-US,en;q=0.8
- Response Headers:** Server: nghttp/1.2.8.6; Date: Mon, 22 Dec 2014 18:25:03 GMT; Content-Type: text/plain; charset=us-ascii; Transfer-Encoding: chunked; Connection: keep-alive

7.4.5. ステップ 5

API の API Management 分析を確認します。

これで 3scale アカウントにログインし、Monitoring > Usage に移動すると、グラフとして表示される API エンドポイントのさまざまなヒットを確認できます。



これは API Management の1つの要素で、API の完全な可視性と制御が可能になります。その他の機能は次のとおりです。

1. アクセス制御
2. 使用ポリシーおよび流量制御
3. レポート
4. API ドキュメントおよびデベロッパーポータル
5. 収益および請求

特定の API Management 機能とその利点についての詳細は、[3scale API Management Platform の製品の説明を参照してください](#)。

特定の Red Hat Fuse 製品機能とその利点に関する詳細は、[JBoss FUSE Overview](#) を参照してください。

Red Hat Fuse on OpenShift の実行に関する詳細は、[JBoss Fuse on OpenShift のスタートガイド](#) を参照してください。