Banco de Galicia:  
Bitácora Upgrade Red Hat 3scale Api Management y Red Hat SSO

Upgrade 3scale de versión 2.9.0 a 2.10

Upgrade RH SSO de versión 7.3 a 7.4

# Prólogo

## Confidencialidad, derechos de autor y descargo de responsabilidad

Este es un documento dirigido al cliente entre Red Hat, Inc. y Banco Galicia.

Copyright © 2019 Red Hat, Inc. Todos los derechos reservados. Ninguna parte del trabajo cubierto por los derechos de autor en este documento puede reproducirse o usarse de ninguna forma o por ningún medio: gráfico, electrónico o mecánico, incluidos los sistemas de fotocopiado, grabación, grabación o almacenamiento y recuperación de información sin permiso por escrito de Red Hat excepto que se requiera compartir esta información como se proporciona con las partes confidenciales mencionadas anteriormente.

Este documento no es una cita y no incluye ningún compromiso vinculante de Red Hat. Si es aceptable, se puede emitir una cotización formal a pedido, que incluirá el alcance del trabajo, el costo y los requisitos del cliente según sea necesario.

## Marcas registradas

Los nombres de marcas registradas pueden aparecer en este documento. En lugar de enumerar los nombres y entidades que poseen las marcas comerciales o insertar un símbolo de marca comercial con cada mención del nombre de la marca comercial, los nombres se usan sólo con fines editoriales y en beneficio del propietario de la marca comercial sin intención de infringir esa marca comercial.

## Audiencia

Este documento está destinado al personal técnico del Cliente responsable del medio ambiente.

## Antecedentes adicionales y documentos relacionados

Este documento no contiene detalles paso a paso de la instalación u otras tareas, ya que están cubiertos en la documentación relevante en <http://access.redhat.com/>. Se harán enlaces a los documentos apropiados cuando sea necesario.

## Scripts y playbooks

Todos los scripts proporcionados se proporcionan tal cual, sin ningún tipo de soporte o garantía. Todos los scripts proporcionados pueden ser modificados por el cliente a voluntad.

# 

# Version history

| Version | Date | Contributor | Role | Description |
| --- | --- | --- | --- | --- |
| 1.0 | 2021-11-16 | Luis Oliveros | Architect | LLD initial documentation |
| 1.1 | 2022-01-04 | Luis Oliveros | Architect | Upgrade 2.9.1 a 2.10 |
| 2.0 | 2022-02-04 | Luis Oliveros | Architect | Actualización documentación |
| 2.1 | 2022-02-14 | Luis Oliveros | Architect | Actualización documentación |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Table of Contents

[**Prólogo**](#_i9u50mkrl1ac) **2**

[Confidencialidad, derechos de autor y descargo de responsabilidad](#_fi9l4jj5ax9v) 2

[Marcas registradas](#_tt5qaj9rxarv) 2

[Audiencia](#_yj73036rqzpe) 2

[Antecedentes adicionales y documentos relacionados](#_ja3izr1m3d71) 2

[Scripts y playbooks](#_wfv6iijjmhej) 2

[**Version history**](#_fht2396ss38r) **3**

[**Introduction**](#_vu7xhysay6ko) **6**

[Sobre Banco Galicia](#_mrwrgdwh1r60) 6

[Statement of work](#_svhl5x60g1a6) 6

[Engagement approach](#_2026huvjkwvu) 6

[Terms and acronyms](#_fqxfyk5qease) 7

[**Pre-Requisitos**](#_s3ph1tvoba2k) **9**

[Entorno Laboratorio](#_3p7014z5tfor) 9

[Configurar la autenticación al registry](#_udtvdcqay1lm) 9

[**Upgrade 3scale 2.9.0 a 2.9.1**](#_jlzvl7upolhb) **10**

[Repositorio](#_zduhmzb7x9rh) 10

[Detalle Tasks Upgrade](#_1zcfuzysf8wo) 10

[Variables playbook](#_e4zqg1zfbg25) 10

[Tareas de upgrade](#_50f53d8raw6n) 12

[Generación de Backups del ambiente.](#_392gekis9v5n) 12

[Creación de imagen amp-system oracle.](#_z84rmym1xiq8) 13

[Upgrade apicast 3scale-b2c-gw](#_v2fv4285zx29) 17

[Generación de Backups.](#_1fuoe28u3fmn) 17

[Upgrade image apicasts](#_6sx75gidtdfi) 18

[Downgrade de 2.9.1 a 2.9.0](#_v0ia47y244ph) **19**

[Detalle Tasks Downgrade](#_90teoc9iom) 20

[Variables](#_sl8sdpj72ee7) 20

[Select Project 3scale](#_abkehyjfislg) 20

[Tareas de downgrade](#_45dgw99q0t4p) 20

[Downgrade apicast 3scale-b2c-gw](#_ekblormdvava) 21

[Select Project 3scale](#_g5lvmouxywq9) 21

[Downgrade image apicasts:](#_8hvtsyx7e6lb) 22

[**Upgrade 2.9.1 a 2.10**](#_vr555cs9obqz) **22**

[Repositorio](#_9pzei2vlud0w) 22

[Detalle Tasks Upgrade](#_rkfha586gscr) 23

[Variables playbook](#_zh3rt0pi0h8w) 23

[Select Project 3scale](#_hwmnouvfv7az) 24

[Tareas de upgrade](#_oaat1regify3) 24

[Generación de Backups del ambiente.](#_7qjc1ftptxnb) 24

[Creación de imagen amp-system oracle.](#_xpqbm92q2omg) 26

[Upgrade apicast 3scale-b2c-gw](#_i22a75iccpaj) 29

[Generación de Backups.](#_mf4dlwze43wk) 30

[Upgrade image apicasts](#_rycces829q1e) 30

[**Downgrade de 2.10 a 2.9.1**](#_86lwijo24ko5) **32**

[Detalle Tasks Downgrade](#_8hcify9gofs9) 32

[Variables lab](#_kdce44ue3flf) 32

[Select Project 3scale](#_2k5p6s6k8eag) 32

[Tareas de downgrade](#_7jlucvbzprmd) 33

[Downgrade apicast 3scale-b2c-gw](#_b2zknbrrud85) 34

[Select Project 3scale](#_eft3e874d6i5) 34

[Downgrade image apicasts:](#_n7bzvnbcniij) 34

[Upgrade Red Hat SSO 7.3 a 7.4](#_3t3wbibjfsq1) 34

[Configurar datos de conexión de base de datos oracle 19c](#_m10ustcou3y) 35

[Detalle de task dentro de playbook upgrade SSO:](#_mew83n4dfo57) 35

[**Downgrade Red Hat SSO 7.3 a 7.4**](#_8pu5skem9kiv) **37**

[Detalle taks en playbook downgrade:](#_jlcwjx36noek) 37

[**Evidencias upgrade de ambientes.**](#_9wsqa03leb38) **37**

[Upgrade 3scale 2.9 a 2.10 - Ambiente Laboratorio.](#_4c3eyzo9g6i7) 37

[Upgrade apicast 3scale-mt-lab](#_ofvlfc6194jw) 37

[Upgrade apicast 3scale-b2c-gw laboratorio](#_bup1koilib34) 41

[Upgrade SSO 7.3 a 7.4 - Ambiente Laboratorio.](#_tf7uv9fodk5o) 42

[Upgrade 3scale 2.9 a 2.10 - Ambiente Desarrollo](#_afaeocaa3ikh) 43

[Upgrade apicast 3scale-mt-desa](#_9u9k9pr1zlsn) 43

[Upgrade apicast 3scale-b2c-gw - Desarrollo](#_n2ea66mc14u3) 46

[Upgrade SSO 7.3 a 7.4 - Ambiente Desarrollo](#_nbo3aaqx7ab) **47**

# 

# 

# Introduction

## Sobre Banco Galicia

El Banco Galicia es un banco privado de la República Argentina. Pertenece al Grupo Financiero Galicia. Fue fundado en 1905 como Banco de Galicia y Buenos Aires, y tradicionalmente su Directorio estuvo compuesto por miembros de las familias Escasany, Ayerza y Braun Menéndez. Sus acciones comenzaron a cotizar en la Bolsa de Comercio de Buenos Aires en 1907.

## Statement of work

El objetivo de los servicios es asistir al Banco Galicia con el upgrade de Red Hat API Manager 3scale 2.9.0 a la versión actual 2.10 y Red Hat SSO de 7.3 a 7.4

## Engagement approach

Para la migración de las versiones se implementarán playbooks de Ansible para automatizar todo el procedimiento, desde el backup hasta la generación del upgrade requerido de la versión del API Manager y el Single Sign On.

## 

## Terms and acronyms

The table below provides a glossary of the terms and acronyms used within this document.

| Acronym | Description |
| --- | --- |
| <customer abbr> | <customer> |
| RH | Red Hat, Inc |
| RHAT | Red Hat Ansible Tower |
| RHCF | Red Hat CloudForms |
| RHEL | Red Hat Enterprise Linux |
| RHOSP | Red Hat OpenStack Platform |
| RHOSPd | Red Hat OpenStack Platform Director |
| RHS | Red Hat Satellite |
| AD | Active Directory |
| ALB | Active Load-Balancing, a link-aggregation technique for NICs |
| AMQP | Advanced Message Queuing Protocol |
| API | Application Programming Interface |
| AZ | Availability Zone |
| CA | Certificate Authority |
| DC | Data Centre |
| DNS | Domain Name System |
| DHCP | Dynamic Host Configuration Protocol |
| DVR | Distributed Virtual Routers, a networking configuration for routing using Compute and Network nodes for separate networking functions |
| FQDN | Fully Qualified Domain Name |
| Guest | Also see “VM”. This is virtual machine running on a Host. |
| HA | High-Availability or Highly-Available |
| Host | The physical hardware or the logical OS which runs virtualisation technology allowing one or more Guest OS’s to run on the hardware owned by the Host |
| IaaS | Infrastructure as a Service |
| IPMI | Intelligent Platform Management Interface |
| KVM | Kernel-based Virtual Machine |
| L2 | Layer 2, part of the TCI/IP Network Stack |
| L3 | Layer 3, part of the TCI/IP Network Stack |
| LACP | Link Aggregation Control Protocol, a link-aggregation technique for NICs |
| LBaaS | Load Balancing as a Service |
| MPIO | Multipath Input/Output |
| NAT | Network Address Translation |
| NIC | Network Interface Card. References a virtual or a physical port allowing network access and interface to a Host or Guest VM. |
| NTP | Network Time Protocol |
| OS | Operating System |
| OVS | Open vSwitch, a Linux-based virtual switch appliance used in OpenStack |
| PXE | Preboot Execution Environment |
| QEMU | See “KVM” |
| QA | Quality Assurance |
| RADOS | rados is a utility for interacting with a Ceph object storage cluster |
| RBD | RADOS Block Device |
| SAML/SAML2 | A protocol allowing authentication and federation of multiple domains |
| SAN | Storage Area Network |
| SFaaS | Shared Filesystem as a Service |
| SSL | Secure Sockets Layer |
| TLS | Transport Layer Security |
| TripleO | The technology used by RHOSPd to deploy an overcloud RHOSP installation |
| VIP | Virtual IP address |
| VLAN | Virtual LAN is a networking virtualisation technology |
| VM | Virtual machine, in OSP terms, synonymous with “Workload” or “Guest” |
| VXLAN | Virtual Extensible LAN (VXLAN) is a network virtualisation technology |
| Workload | Synonym for “Guest” or “VM” |

# Pre-Requisitos

## Entorno Laboratorio

Se requiere realizar un entorno de Laboratorio siendo este una copia fiel de desarrollo a fin de realizar en este todo el proceso de migración y homologarlo para el resto de los entornos teniendo el mínimo impacto posible sobre el ciclo de vida y uptime de los servicios del Banco.

## Configurar la autenticación al registry

Las imágenes de 3scale se encuentran en la registry registry.redhat.io. Entonces necesitaremos crear un nuevo secret el cual contenga las credenciales de la cuenta.

| apiVersion: v1  kind: Secret  metadata:  name: threescale-registry-auth  data:  .dockerconfigjson: a-base64-encoded-string-containing-auth-credentials  type: kubernetes.io/dockerconfigjson |
| --- |

Una vez creado el archivo Secret con el base64 se procederá a crearlo en el namespaces correspondiente a 3scale en OCP

| oc create -f the-secret-name.yml |
| --- |

Validamos que exista el nuevo Secret

| oc get secret threescale-registry-auth |
| --- |

Linkear secret con default y builder

| oc secrets link default threescale-registry-auth --for=pull  oc secrets link builder threescale-registry-auth --for=build |
| --- |

Para más información sobre cómo crear las credenciales [Red Hat Container Registry Authentication](https://access.redhat.com/RegistryAuthentication)

El entorno de laboratorio se despliega para tener una base de prueba con respecto a la creación de los scripts y playbooks que se ocuparan para realizar el upgrade de las diferentes plataformas.

# Upgrade 3scale 2.9.0 a 2.9.1

## Repositorio

Los archivos de configuración y playbooks yamls se encuentran en el siguiente repositorio <https://github.bancogalicia.com.ar/tytm-middleware/update-ocp311-3scale.git>

El playbook para generar el upgrade **3scale-upgrade-290-to-291.yml** se encuentra en el directorio:

| .  ├── 3scale-upgrade-downgrade-290-to-291  │ ├── 3scale-upgrade-290-to-291.yml  │ ├── downgrade-3scale-291-to-290.yml  │ ├── README.md  │ └── roles  │ ├── downgrade-291-to-290  │ │ ├── defaults  │ │ │ └── main.yml  │ │ └── tasks  │ │ ├── main.yml  │ │ ├── wait\_for\_apicast\_deploy\_d.yml  │ │ └── wait\_for\_deploy\_d.yml  │ └── upgrade-291  │ ├── defaults  │ │ └── main.yml  │ ├── system-oracle  │ │ ├── build.yml  │ │ ├── Dockerfile  │ │ ├── dockerfile.patch  │ │ ├── oracle-client-files  │ │ │ ├── instantclient-basic-linux.x64-12.2.0.1.0.zip  │ │ │ ├── instantclient-odbc-linux.x64-12.2.0.1.0-2.zip  │ │ │ └── instantclient-sdk-linux.x64-12.2.0.1.0.zip  │ │ ├── workaround-non-cdb-2.4-upgrade  │ │ │ ├── database.rb  │ │ │ └── dockerfile.patch  │ │ └── workaround-non-cdb-2.5-upgrade  │ │ ├── database.rb  │ │ └── dockerfile.patch  │ └── tasks  │ ├── get\_backup\_290.yml  │ ├── main.yml  │ ├── upgrade\_apicast.yml  │ ├── wait\_for\_apicast\_deploy.yml  │ └── wait\_for\_deploy.yml |
| --- |

## Detalle Tasks Upgrade

### Variables playbook

Las variables declaradas para la ejecución del playbook por cada ambiente se deben realizar en el archivo main.yml ubicado en la carpeta del proyecto:

| 3scale-upgrade-downgrade-290-to-291/roles/upgrade-291/defaults/main.yml |
| --- |

| # 3scale mt - VARS #  CURRENT\_VERSION: <<CURRENT\_VERSION>>  CURRENT\_ORACLE\_VERSION: 2.9-oracle  ENV\_OCP: <<ENV\_OCP>>  PROJECT\_NAME: 3scale-mt  API\_MANAGER\_NS: "{{PROJECT\_NAME}}-{{ENV\_OCP}}"  THREESCALE\_DC\_NAMES: "apicast-production apicast-staging backend-cron backend-listener backend-redis backend-worker system-app system-memcache system-redis system-sidekiq system-sphinx zync zync-database zync-que"  THREESCALE\_IMAGESTREAM\_NAMES: "amp-apicast amp-backend amp-system amp-zync backend-redis system-memcached system-mysql system-redis zync-database-postgresql"  # 3scale apicast - VARS #  ENV\_OCP\_GW: <<ENV\_OCP\_GW>>  PROJECT\_NAME\_GW: 3scale-b2c  API\_MANAGER\_NS\_GW: "{{PROJECT\_NAME\_GW}}-{{ENV\_OCP\_GW}}"  THREESCALE\_DC\_NAMES\_GW: "prod-apicast stage-apicast" |
| --- |

El archivo main.yml contiene la información en el cual se establecen las propiedades del playbook, de manera que dinámicamente estas variables se pueden modificar dependiendo del requerimiento que se espera atender. Estas variables contienen la siguiente descripción:

| **Variables declaradas** | **Descripción** |
| --- | --- |
| CURRENT\_VERSION | Se debe especificar el tag de la imagen que se pretende migrar según el ambiente. |
| ENV\_OCP | Se debe especificar el nombre del cluster en el cual se pretende realizar el upgrade, en esta caso pueden ser las siguientes:   * lab * desa * homo * prod |
| ENV\_OCP\_GW | Se debe especificar el nombre del ambiente en el cual se pretende realizar el upgrade.   * gw (desa y prod) * homo-gw (caso QA) |

* 1. Select Project 3scale

Para iniciar el proceso de upgrade el playbook se posiciona sobre el proyecto correspondiente a través del comando:

| oc project {{ API\_MANAGER\_NS }} |
| --- |

### Tareas de upgrade

El archivo que inicializa el proceso de la ejecución del playbook es *3scale-upgrade-290-to-291.yml* el cual hace referencia al archivo ubicado en la siguiente ruta:

| 3scale-upgrade-downgrade-290-to-291/roles/upgrade-291/task/main.yml. |
| --- |

Este archivo es el cual contiene la ejecución de las tareas del playbook correspondiente al upgrade de 3scale.

#### Generación de Backups del ambiente.

En base al punto *2.2 Backup* se ejecutan los siguientes comandos sobre el playbook para obtenerlos:

| - name: create backup file with DeploymentConfig  shell: "{{ item }}"  loop:  - "mkdir backup-3scale"  - "mkdir backup-3scale/threescale-elements"  - "export THREESCALE\_DC\_NAMES=\"apicast-production apicast-staging backend-cron backend-listener backend-redis backend-worker system-app system-memcache system-redis system-sidekiq system-sphinx zync zync-database zync-que\""  - "for component in ${THREESCALE\_DC\_NAMES}; do oc get -o yaml dc ${component} > backup-3scale/${component}\_dc.yml ; done"  - "oc get -o yaml all > backup-3scale/threescale-elements/threescale-project-elements.yaml"  - "for object in rolebindings serviceaccounts secrets imagestreamtags cm rolebindingrestrictions limitranges resourcequotas pvc templates cronjobs statefulsets hpa deployments replicasets poddisruptionbudget endpoints; do oc get -o yaml $object > backup-3scale/threescale-elements/$object.yaml; done"  - "for object in `oc get routes | awk '{print $1}' | grep -v NAME`; do oc get -o yaml route ${object} > backup-3scale/threescale-elements/${object}\_route.yaml; done"  - name: create backup file with ImageStream  shell: "{{item}}"  loop:  - "export THREESCALE\_IMAGESTREAM\_NAMES=\"amp-apicast amp-backend amp-system amp-zync backend-redis system-memcached system-mysql system-redis zync-database-postgresql\""  - "for component in ${THREESCALE\_IMAGESTREAM\_NAMES}; do oc get -o yaml is ${component} > backup-3scale/${component}\_is.yml ; done"  - name: create backup file with system-redis secret  shell: "oc get -o yaml secret system-redis > backup-3scale/system-redis\_secret.yml"  - name: create backup file all project  shell: "oc get -o yaml all > backup-3scale/threescale-project-elements.yaml"  # get system backup #  - include\_tasks: get\_backup\_290.yml |
| --- |

Inicialmente se crea carpeta backup-3scale donde se almacenarán los yml de respaldo y luego la ejecución de los comandos para obtenerlos, además de un include donde se almacenan los datos de respaldo del sistema.

|  | Verificar que todos los archivos generados no estén vacíos y que todos tengan sus respectivos contenidos. Modificar los archivos eliminando la referencia del namespaces. Importante al momento de utilizar este backup, se deberá borrar los atributos que se autogenerar o representan información del recurso en ese momento, por ejemplo *status, creationTimestamp, selfLink, uid, resourceVersion* |
| --- | --- |

* Descripción tareas “include\_tasks: get\_backup\_290.yml”

| # Backing up system databases #  - name: create folder backup-3ssystem290  shell: "{{ item }}"  loop:  - "mkdir backup-3scale-290/backup-3ssystem290"  - "mkdir backup-3scale-290/backup-3ssystem290/dir"  - name: backing up system-storage  shell: "oc rsync $(oc get pods -l 'deploymentConfig=system-app' -o json | jq '.items[0].metadata.name' -r):/opt/system/public/system backup-3scale-290/backup-3ssystem290/dir"  - name: backing up backend-redis  shell: "oc cp $(oc get pods -l 'deploymentConfig=backend-redis' -o json | jq '.items[0].metadata.name' -r):/var/lib/redis/data/dump.rdb backup-3scale-290/backup-3ssystem290/backend-redis-dump.rdb"  - name: backing up system-redis  shell: "oc cp $(oc get pods -l 'deploymentConfig=system-redis' -o json | jq '.items[0].metadata.name' -r):/var/lib/redis/data/dump.rdb backup-3scale-290/backup-3ssystem290/system-redis-dump.rdb"  - name: backing up zync-database  shell: "oc rsh $(oc get pods -l 'deploymentConfig=zync-database' -o json | jq -r '.items[0].metadata.name') bash -c 'pg\_dump zync\_production' | gzip > backup-3scale-290/backup-3ssystem290/zync-database-backup.gz"  # Backing up OpenShift secrets and ConfigMaps #  - name: openShift namespace secrets  shell: "{{ item }}"  loop:  - "oc get secrets system-smtp -o json > backup-3scale-290/backup-3ssystem290/system-smtp.json"  - "oc get secrets system-seed -o json > backup-3scale-290/backup-3ssystem290/system-seed.json"  - "oc get secrets system-database -o json > backup-3scale-290/backup-3ssystem290/system-database.json"  - "oc get secrets backend-internal-api -o json > backup-3scale-290/backup-3ssystem290/backend-internal-api.json"  - "oc get secrets system-events-hook -o json > backup-3scale-290/backup-3ssystem290/system-events-hook.json"  - "oc get secrets system-app -o json > backup-3scale-290/backup-3ssystem290/system-app.json"  - "oc get secrets system-recaptcha -o json > backup-3scale-290/backup-3ssystem290/system-recaptcha.json"  - "oc get secrets system-redis -o json > backup-3scale-290/backup-3ssystem290/system-redis.json"  - "oc get secrets zync -o json > backup-3scale-290/backup-3ssystem290/zync.json"  - "oc get secrets system-master-apicast -o json > backup-3scale-290/backup-3ssystem290/system-master-apicast.json"  - "oc get secrets threescale-registry-auth -o json > backup-3scale-290/backup-3ssystem290/threescale-registry-auth.json"  - name: openShift namespace configmaps  shell: "{{ item }}"  loop:  - "oc get configmaps system-environment -o json > backup-3scale-290/backup-3ssystem290/system-environment.json"  - "oc get configmaps apicast-environment -o json > backup-3scale-290/backup-3ssystem290/apicast-environment.json" |
| --- |

#### Creación de imagen amp-system oracle.

Este procedimiento lo guía a través de una actualización de Oracle Database 12c para 3scale 2.9.1 desde una instalación de 3scale 2.9 existente. Los archivos y drivers necesarios se encuentran sobre el repo en directorio:

| .  ├── **3scale-upgrade-290-to-291.yml**  └── roles  └── upgrade-291  ...  **├── system-oracle**  **│ ├── build.ymlº**  **│ ├── Dockerfile**  **│ ├── dockerfile.patch**  **│ ├── oracle-client-files**  **│ │ ├── instantclient-basic-linux.x64-12.2.0.1.0.zip**  **│ │ ├── instantclient-odbc-linux.x64-12.2.0.1.0-2.zip**  **│ │ └── instantclient-sdk-linux.x64-12.2.0.1.0.zip**  **│ ├── workaround-non-cdb-2.4-upgrade**  **│ │ ├── database.rb**  **│ │ └── dockerfile.patch**  **│ └── workaround-non-cdb-2.5-upgrade**  **│ ├── database.rb**  **│ └── dockerfile.patch**  ... |
| --- |

Se ejecuta el comando oc process con la opción -f y especifique la plantilla build.yml OpenShift y crear el objeto build para construir la imagen amp-system 2.9.1-oracle:

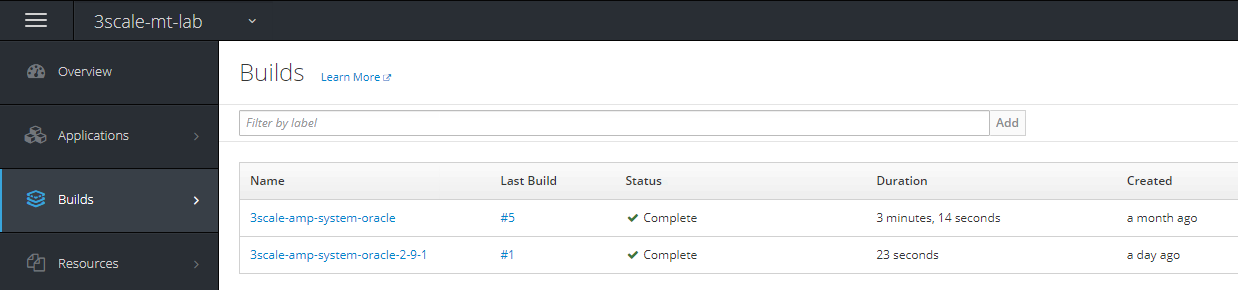
| oc process -f roles/upgrade-291/system-oracle/build.yml | oc apply -f - |
| --- |

Una vez creado se corre el siguiente comando para construir la nueva imagen del sistema con oracle:

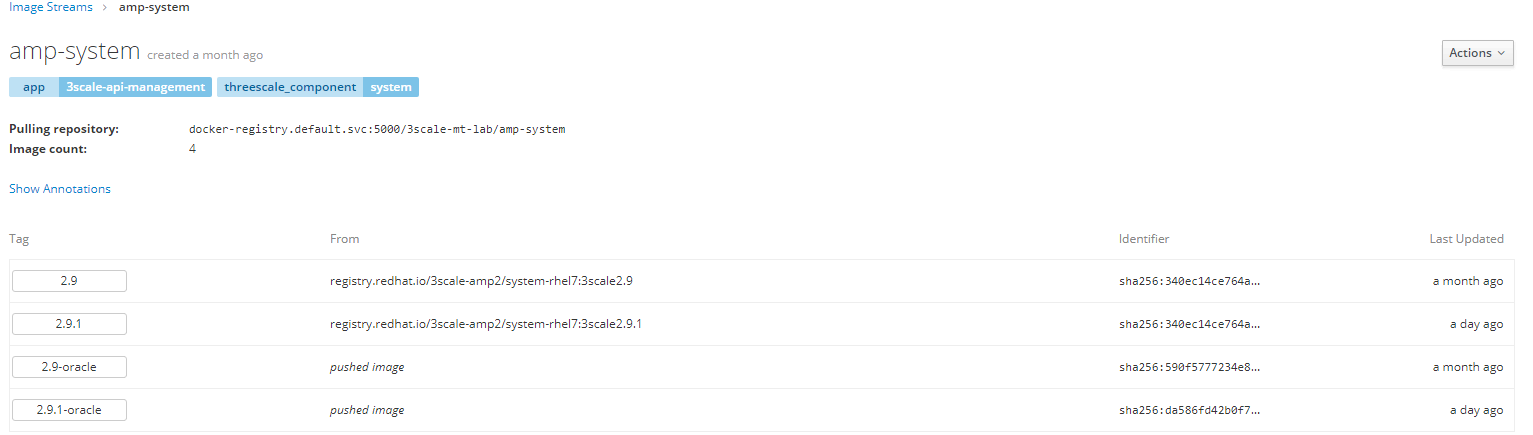
| oc start-build 3scale-amp-system-oracle-2-9-1 --from-dir=roles/upgrade-291/system-oracle/. -n {{ API\_MANAGER\_NS }} --wait |
| --- |

{{ API\_MANAGER\_NS }} = Variable Proyecto 3scale.

Una vez construida el Build se visualiza sobre el proyecto se puede observar de la siguiente manera:



Sobre el imagen stream se evidencia igual la creación de nuevo tag:



|  | IMPORTANTE: La pérdida de conexión a la base de datos puede dañar 3scale. Haga una copia de seguridad antes de continuar con la actualización |
| --- | --- |

A partir de este punto se generan los patch correspondientes a los ImagenStream apuntando hacia la versión 2.9.1 y los patch sobre cada DeploymentConfig dentro del proyecto para la generación del Upgrade.

La imagen amp-system con oracle se ocupa para los siguientes deploymentConfig:

| **deploymenConfig amp-system** |
| --- |
| system-app |
| system-sidekiq |
| system-sphinx |

Generando el patch con los siguientes comandos, removiendo el trigger correspondiente a la versión 2.9 y agregando Agregar el nuevo trigger específico de la versión 2.9.1-oracle:

* System-App:

| "oc set triggers dc/system-app --from-image=amp-system:{{ CURRENT\_ORACLE\_VERSION }} --containers=system-master,system-developer,system-provider --remove"  "oc set triggers dc/system-app --from-image=amp-system:2.9.1-oracle --containers=system-master,system-developer,system-provider" |
| --- |

* System-Sidekiq:

| "oc set triggers dc/system-sidekiq --from-image=amp-system:{{ CURRENT\_ORACLE\_VERSION }} --containers=system-sidekiq,check-svc --remove"  "oc set triggers dc/system-sidekiq --from-image=amp-system:2.9.1-oracle --containers=system-sidekiq,check-svc" |
| --- |

* System-Sphinx:

| "oc set triggers dc/system-sphinx --from-image=amp-system:{{ CURRENT\_ORACLE\_VERSION }} --containers=system-sphinx,system-master-svc --remove"  "oc set triggers dc/system-sphinx --from-image=amp-system:2.9.1-oracle --containers=system-sphinx,system-master-svc" |
| --- |

Por cada ajuste dentro de los deploymentConfig se agrega un task que espera el redespliegue de cada POD involucrado, haciendo referencia a un include-task hacia wait\_to\_deploy.yml con la variable correspondiente al dc que se está ajustando, el contenido de este task es:

| # wait to deploy #  - name: "wait to deploy {{pod\_to\_wait}}"  wait\_for:  timeout: 5  - name: "redeploy {{pod\_to\_wait}}"  command: 'oc rollout latest dc/"{{pod\_to\_wait}}"'  ignore\_errors: true  - name: "wait for following deployments to become ready {{pod\_to\_wait}}"  command: 'oc rollout status dc/"{{pod\_to\_wait}}"' |
| --- |

Donde *{{ pod\_to\_wait }}* es la variable que corresponde al dc que se está cambiando.

Para los siguientes componentes dentro del proyecto de 3scale se van a crear cada tag con la versión 2.9.1:

| **image-ref** | **url-registry** | **tag** |
| --- | --- | --- |
| amp-apicast | registry.redhat.io/3scale-amp2/apicast-gateway-rhel8:3scale2.9.1 | 2.9.1 |
| amp-backend | registry.redhat.io/3scale-amp2/backend-rhel7:3scale2.9.1 | 2.9.1 |
| amp-zync | registry.redhat.io/3scale-amp2/zync-rhel7:3scale2.9.1 | 2.9.1 |
| backend-redis | registry.redhat.io/rhscl/redis-32-rhel7:3.2 | 2.9.1 |
| system-memcached | registry.redhat.io/3scale-amp2/memcached-rhel7:3scale2.9.1 | 2.9.1 |
| system-mysql | registry.redhat.io/rhscl/mysql-57-rhel7:5.7 | 2.9.1 |
| system-redis | registry.redhat.io/rhscl/redis-32-rhel7:3.2 | 2.9.1 |
| zync-database-postgresql | registry.redhat.io/rhscl/postgresql-10-rhel7 | 2.9.1 |

Para generar el patch de cada Imagen Stream se ocupa el siguiente comando:

| oc patch imagestream/**<<image-ref>>** --type=json -p '[{\"op\": \"add\", \"path\": \"/spec/tags/-\", \"value\": {\"annotations\": {\"openshift.io/display-name\": \"Zync 2.9.1 PostgreSQL\"}, \"from\": { \"kind\": \"DockerImage\", \"name\": \"**<<url-registry>>**\"}, \"name\": \"2.9.1\", \"referencePolicy\": {\"type\": \"Source\"}}}]' |
| --- |

Por cada tag de image stream generado se deben actualizar los deploymentConfig:

| **dc-ref** | **image-ref** | **containers** |
| --- | --- | --- |
| apicast-production | amp-apicast:2.9.1 | apicast-production,system-master-svc |
| apicast-staging | amp-apicast:2.9.1 | apicast-staging |
| backend-cron | amp-backend:2.9.1 | backend-cron,backend-redis-svc |
| backend-listener | amp-backend:2.9.1 | backend-listener |
| backend-redis | backend-redis:2.9.1 | backend-worker,backend-redis-svc |
| backend-worker | amp-backend:2.9.1 | backend-worker,backend-redis-svc |
| system-memcache | system-memcached:2.9.1 | memcache |
| system-redis | system-redis:2.9.1 | system-redis |
| zync | zync:2.9.1 | zync,zync-db-svc |
| zync-database | zync:2.9.1 | n/e |
| zync-que | zync:2.9.1 | que |

El cambio se realiza a través del siguiente comando por cada dc en la tabla, removiendo el trigger correspondiente a la versión 2.9 y agregando Agregar el nuevo trigger específico de la versión 2.9.1:

| "oc set triggers dc/**<<dc-ref>>** --from-image=**<<image-ref>>**:{{ CURRENT\_VERSION }} --containers=**<<dc-ref>>** --remove"  "oc set triggers dc/**<<dc-is-ref>>** --from-image=**<<dimage-ref>>**:2.9.1 --containers=**<<containers>>**" |
| --- |

Donde *{{ CURRENT\_VERSION }}* es la versión 2.9 actual sobre el deployment config.

Por cada ajuste dentro de los deploymentConfig se agrega un task que espera el redespliegue de cada POD involucrado, haciendo referencia a un include-task hacia wait\_to\_deploy.yml.

### Upgrade apicast 3scale-b2c-gw

Una vez generado el upgrade sobre el proyecto 3scale se agrega un include task para generar el upgrade dentro del proyecto b2c donde se ubican los apicast:

| # updgrade apicast 2.9.0 to 2.9.1 #  - name: upgrade\_apicast\_gw  include\_tasks: upgrade\_apicast.yml |
| --- |

Dentro del playbook upgrade\_apicast.yml se ejecutan los comandos de patch sobre el deployment sobre el apicast staging y de production apuntando directamente hacia el registry con el tag 2.9.1 ya que estos no poseen ImageStream; los comando ejecutados son los siguientes:

#### Generación de Backups.

En este paso se ejecutan los siguientes comandos sobre el playbook para obtenerlos:

| - name: create backup file with deployment  shell: "{{ item }}"  loop:  - "mkdir backup-3scale-gw-290"  - "mkdir backup-3scale-gw-290/threescale-elements"  - "for component in {{THREESCALE\_DC\_NAMES\_GW}}; do oc get -o yaml deployment ${component} > backup-3scale-gw-290/${component}\_dc.yml ; done"  - "oc get -o yaml all > backup-3scale-gw-290/threescale-elements/threescale-project-elements.yaml"  - "for object in rolebindings serviceaccounts secrets imagestreamtags cm rolebindingrestrictions limitranges resourcequotas pvc templates cronjobs statefulsets hpa deployments replicasets poddisruptionbudget endpoints; do oc get -o yaml $object > backup-3scale-gw-290/threescale-elements/$object.yaml; done"  - "for object in `oc get routes | awk '{print $1}' | grep -v NAME`; do oc get -o yaml route ${object} > backup-3scale-gw-290/${object}\_route.yaml; done"  - name: create backup file all project  shell: "oc get -o yaml all > backup-3scale-gw-290/threescale-project-elements.yaml" |
| --- |

#### Upgrade image apicasts

Para generar el patch de cada Imagen Stream se ocupa el siguiente comando:

* Apicast-staging:

| oc patch deployment stage-apicast --patch='{\"spec\":{\"template\":{\"spec\":{\"containers\":[{\"name\": \"stage-apicast\", \"image\":\"registry.redhat.io/3scale-amp2/apicast-gateway-rhel8:3scale2.9.1\"}]}}}}' |
| --- |

* Apicast-production:

| oc patch deployment prod-apicast --patch='{\"spec\":{\"template\":{\"spec\":{\"containers\":[{\"name\": \"prod-apicast\", \"image\":\"registry.redhat.io/3scale-amp2/apicast-gateway-rhel8:3scale2.9.1\"}]}}}}'" |
| --- |

Por cada ajuste dentro de los deploymentConfig se agrega un task que espera el redespliegue de cada POD involucrado, haciendo referencia a un include-task hacia wait\_for\_apicast\_deploy.yml

Nota: Con respecto al upgrade de los Apicast de STG/QA se generó un playbook que recorriera los 3 namespace que pertenecen a este ambiente en su actualización, los cuales son:

* 3scale-b2c-homo-gw
* 3scale-b2b-inte-gw
* 3scale-b2c-inte-gw
* Playbook upgrade apicast STG/QA:

| ---  # ################################ project 3scale-b2c-homo-gw ##################################### #  - name: check namespace 3scale b2c homo gw  shell: "oc project 3scale-b2c-homo-gw"  register: api\_project\_location\_gw    - name: namespace name  debug:  msg: "{{ api\_project\_location\_gw }}"  # ########## create backup project  ---  # patch the "apicast" image #  ---  # ################################ project 3scale-b2b-inte-gw ##################################### #  - name: check namespace 3scale b2b inte gw  shell: "oc project 3scale-b2b-inte-gw"  register: api\_project\_location\_gw    - name: namespace name  debug:  msg: "{{ api\_project\_location\_gw }}"  # ########## create backup project  ---  # patch the "apicast" image #  ---  # ################################ project 3scale-b2c-inte-gw ##################################### #  - name: check namespace 3scale b2c inte gw  shell: "oc project 3scale-b2c-inte-gw"  register: api\_project\_location\_gw    - name: namespace name  debug:  msg: "{{ api\_project\_location\_gw }}"  # ########## create backup project  # create backup deploymentConfig y rutas #  ---  # patch the "apicast" image #  --- |
| --- |

# Downgrade de 2.9.1 a 2.9.0

En caso de ser requerido y durante el proceso de ejecución de upgrade, existe un playbook de ANSIBLE que permite la vuelta atrás en versión de 3scale hacia la 2.9, el playbook se ubica sobre el siguiente directorio:

| 3scale-upgrade-galicia/3scale-upgrade-downgrade-291/  . . .  ├── downgrade-3scale-291-to-290.yml  └── roles  ├── downgrade-291-to-290  │ ├── defaults  │ │ └── main.yml  │ └── tasks  │ ├── main.yml  │ ├── wait\_for\_apicast\_deploy\_d.yml  │ └── wait\_for\_deploy\_d.yml  . . . |
| --- |

## 

## Detalle Tasks Downgrade

### Variables

Las variables declaradas para la ejecución del playbook sobre ambiente LAB:

| ---  # 3scale images versions #  CURRENT\_VERSION: 2.9.1  CURRENT\_ORACLE\_VERSION: 2.9.1-oracle  TO\_VERSION: 2.9  TO\_ORACLE\_VERSION: 2.9-oracle  # 3scale mt - VARS #  ENV\_OCP: <<ENV\_OCP\_GW>>  PROJECT\_NAME: 3scale-mt  API\_MANAGER\_NS: "{{PROJECT\_NAME}}-{{ENV\_OCP}}"  # 3scale apicast - VARS #  ENV\_OCP\_GW: gw  PROJECT\_NAME\_GW: 3scale-b2c  API\_MANAGER\_NS\_GW: "{{PROJECT\_NAME\_GW}}-{{ENV\_OCP\_GW}}" |
| --- |

### Select Project 3scale

Para iniciar el proceso de upgrade el playbook se posiciona sobre el proyecto correspondiente a través del comando:

| oc project {{ API\_MANAGER\_NS }} |
| --- |

### Tareas de downgrade

El proceso de downgrade se genera teniendo en cuenta que se toman los tags de ImageStream en la versión 2.9 en el proyecto, para system-app se toma el tag 2.9-oracle:

| **dc-is** | **tag** |
| --- | --- |
| system-app | 2.9-oracle |
| system-sidekiq | 2.9-oracle |
| system-sphinx | 2.9-oracle |

Para el downgrade de los componentes con amp-system se ocupan los siguiente comandos:

| ***system-app***  "oc set triggers dc/system-app --from-image=amp-system:{{ CURRENT\_ORACLE\_VERSION }} --containers=system-master,system-developer,system-provider --remove"  "oc set triggers dc/system-app --from-image=amp-system:{{ TO\_ORACLE\_VERSION }} --containers=system-master,system-developer,system-provider" |
| --- |
| ***system-sidekiq***  "oc set triggers dc/system-sidekiq --from-image=amp-system:{{ CURRENT\_ORACLE\_VERSION }} --containers=system-sidekiq,check-svc --remove"  "oc set triggers dc/system-sidekiq --from-image=amp-system:{{ TO\_ORACLE\_VERSION }} --containers=system-sidekiq,check-svc" |
| ***system-sidekiq***  "oc set triggers dc/system-sphinx --from-image=amp-system:{{ CURRENT\_ORACLE\_VERSION }} --containers=system-sphinx,system-master-svc --remove"  "oc set triggers dc/system-sphinx --from-image=amp-system:{{ TO\_ORACLE\_VERSION }} --containers=system-sphinx,system-master-svc" |

Vars:

{{ CURRENT\_ORACLE\_VERSION }} = Versión actual correspondiente al 2.9.1-oracle

{{ TO\_ORACLE\_VERSION }} = Version que se insertará en el dc 2.9-oracle

| **dc-ref** | **image-ref** | **containers** |
| --- | --- | --- |
| apicast-production | amp-apicast:2.9 | apicast-production,system-master-svc |
| apicast-staging | amp-apicast:2.9 | apicast-staging |
| backend-cron | amp-backend:2.9 | backend-cron,backend-redis-svc |
| backend-listener | amp-backend:2.9 | backend-listener |
| backend-redis | backend-redis:2.9 | backend-worker,backend-redis-svc |
| backend-worker | amp-backend:2.9 | backend-worker,backend-redis-svc |
| system-memcache | system-memcached:2.9 | memcache |
| system-redis | system-redis:2.9 | system-redis |
| zync | zync:2.9 | zync,zync-db-svc |
| zync-database | zync:2.9 | n/e |
| zync-que | zync:2.9 | que |

Para el downgrade de los componentes en la anterior tabla se ocupa el siguiente comando:

| "oc set triggers dc/<<dc-ref>> --from-image=<<image-ref>>:{{ CURRENT\_VERSION }} --containers=<<dc-ref>> --remove"  "oc set triggers dc/<<dc-ref>> --from-image=<<image-ref>>:{{ TO\_VERSION }} --containers=<<containers>>" |
| --- |

Por cada ajuste dentro de los deploymentConfig se agrega un task que espera el redespliegue de cada POD involucrado, haciendo referencia a un include-task hacia wait\_to\_apicast\_deploy\_d.yml.

### Downgrade apicast 3scale-b2c-gw

#### Select Project 3scale

Para iniciar el proceso de upgrade el playbook se posiciona sobre el proyecto correspondiente a los gw a través del comando:

| oc project {{ API\_MANAGER\_NS\_GW }} |
| --- |

#### Downgrade image apicasts:

Para generar el patch de cada Imagen Stream se ocupa el siguiente comando:

* Apicast-staging:

| oc patch dc stage-apicast --patch='{\"spec\":{\"template\":{\"spec\":{\"containers\":[{\"name\": \"stage-apicast\", \"image\":\"registry.redhat.io/3scale-amp2/apicast-gateway-rhel8:3scale2.9.0\"}]}}}}' |
| --- |

* Apicast-production:

| oc patch dc prod-apicast --patch='{\"spec\":{\"template\":{\"spec\":{\"containers\":[{\"name\": \"prod-apicast\", \"image\":\"registry.redhat.io/3scale-amp2/apicast-gateway-rhel8:3scale2.9.0\"}]}}}}'" |
| --- |

Por cada ajuste dentro de los deploymentConfig se agrega un task que espera el redespliegue de cada POD involucrado, haciendo referencia a un include-task hacia wait\_to\_apicast\_deploy\_d.yml.

Nota: Con respecto a STG/QA dentro del playbook de downgrade recorrerá los 3 namespace que pertenecen a este ambiente, los cuales son:

* 3scale-b2c-homo-gw
* 3scale-b2b-inte-gw
* 3scale-b2c-inte-gw

# Upgrade 2.9.1 a 2.10

## Repositorio

Los archivos de configuración, playbooks, yamls se encuentran en el siguiente repositorio

<https://github.bancogalicia.com.ar/tytm-middleware/update-ocp311-3scale/tree/master/3scale-upgrade-downgrade-291-to-210>

El playbook para generar el upgrade **3scale-upgrade-291-to-210.yml** se encuentra en el directorio:

| └── 3scale-upgrade-downgrade-291-to-210  ├── 3scale-downgrade-210-to-291.yml  ├── 3scale-upgrade-291-to-210.yml  ├── README.md  └── roles  ├── downgrade-210-to-291  │ ├── defaults  │ │ └── main.yml  │ └── tasks  │ ├── main.yml  │ ├── wait\_for\_apicast\_deploy\_d.yml  │ └── wait\_for\_deploy\_d.yml  └── upgrade-210  ├── defaults  │ └── main.yml  ├── system-oracle  │ ├── build.yml  │ ├── Dockerfile  │ ├── dockerfile.patch  │ ├── oracle-client-files  │ │ ├── instantclient-basic-linux.x64-19.13.0.0.0dbru-2.zip  │ │ ├── instantclient-odbc-linux.x64-19.13.0.0.0dbru-2.zip  │ │ └── instantclient-sdk-linux.x64-19.13.0.0.0dbru-2.zip  │ ├── README.md  │ ├── workaround-non-cdb-2.4-upgrade  │ │ ├── database.rb  │ │ └── dockerfile.patch  │ └── workaround-non-cdb-2.5-upgrade  │ ├── database.rb  │ └── dockerfile.patch  └── tasks  ├── get\_backup\_291.yml  ├── main.yml  ├── upgrade\_apicast.yml  ├── wait\_for\_apicast\_deploy.yml  └── wait\_for\_deploy.yml |
| --- |

## Detalle Tasks Upgrade

### Variables playbook

Las variables declaradas para la ejecución del playbook por cada ambiente se deben realizar en el archivo main.yml ubicado en la carpeta del proyecto:

*3scale-upgrade-downgrade-291-to-210/roles/upgrade-210/defaults/main.yml*

| ---  # 3scale mt - VARS #  CURRENT\_VERSION: <<CURRENT\_VERSION>>  CURRENT\_ORACLE\_VERSION: 2.9.1-oracle  ENV\_OCP: <<CURRENT\_VERSION>>  PROJECT\_NAME: 3scale-mt  API\_MANAGER\_NS: "{{PROJECT\_NAME}}-{{ENV\_OCP}}"  THREESCALE\_DC\_NAMES: "apicast-production apicast-staging backend-cron backend-listener backend-redis backend-worker system-app system-memcache system-redis system-sidekiq system-sphinx zync zync-database zync-que"  THREESCALE\_IMAGESTREAM\_NAMES: "amp-apicast amp-backend amp-system amp-zync backend-redis system-memcached system-mysql system-redis zync-database-postgresql"  # 3scale apicast - VARS #  ENV\_OCP\_GW: <<ENV\_OCP\_GW>>  PROJECT\_NAME\_GW: 3scale-b2c  API\_MANAGER\_NS\_GW: "{{PROJECT\_NAME\_GW}}-{{ENV\_OCP\_GW}}"  THREESCALE\_DC\_NAMES\_GW: "prod-apicast stage-apicast" |
| --- |

El archivo main.yml contiene la información en el cual se establecen las propiedades del º, de manera que dinámicamente estas variables se pueden modificar dependiendo del requerimiento que se espera atender. Estas variables contienen la siguiente descripción:

¿qué ladilla verdad?

| **Variables declaradas** | **Descripción** |
| --- | --- |
| CURRENT\_VERSION | Se debe especificar el tag de la imagen que se pretende migrar según el ambiente. |
| ENV\_OCP | Se debe especificar el nombre del cluster en el cual se pretende realizar el upgrade, en esta caso pueden ser las siguientes:   * lab * desa * homo * prod |
| ENV\_OCP\_GW | Se debe especificar el nombre del ambiente en el cual se pretende realizar el upgrade.   * gw (desa y prod) * homo-gw (caso QA) |

### Select Project 3scale

Para iniciar el proceso de upgrade el playbook se posiciona sobre el proyecto correspondiente a través del comando:

| oc project {{ API\_MANAGER\_NS }} |
| --- |

### Tareas de upgrade

El archivo que inicializa el proceso de la ejecución del playbook es 3scale-upgrade-291-to-210.yml el cual hace referencia al archivo ubicado en la siguiente ruta:

| 3scale-upgrade-downgrade-291-to-210/roles/upgrade-210/task/main.yml |
| --- |

Este archivo es el cual contiene la ejecución de las tareas del playbook correspondiente al upgrade de 3scale.

#### Generación de Backups del ambiente.

En base al punto *2.2 Backup* se ejecutan los siguientes comandos sobre el playbook para obtenerlos:

| - name: create backup file with DeploymentConfig  shell: "{{ item }}"  loop:  - "mkdir backup-3scale-291"  - "mkdir backup-3scale-291/threescale-elements"  - "for component in {{THREESCALE\_DC\_NAMES}}; do oc get -o yaml dc ${component} > backup-3scale-291/${component}\_dc.yml ; done"  - "oc get -o yaml all > backup-3scale-291/threescale-elements/threescale-project-elements.yaml"  - "for object in rolebindings serviceaccounts secrets imagestreamtags cm rolebindingrestrictions limitranges resourcequotas pvc templates cronjobs statefulsets hpa deployments replicasets poddisruptionbudget endpoints; do oc get -o yaml $object > backup-3scale-291/threescale-elements/$object.yaml; done"  - "for object in `oc get routes | awk '{print $1}' | grep -v NAME`; do oc get -o yaml route ${object} > backup-3scale-291/threescale-elements/${object}\_route.yaml; done"  - name: create backup file with ImageStream  shell: "{{item}}"  loop:  - "for component in {{THREESCALE\_IMAGESTREAM\_NAMES}}; do oc get -o yaml is ${component} > backup-3scale-291/${component}\_is.yml ; done"  - name: create backup file with system-redis secret  shell: "oc get -o yaml secret system-redis > backup-3scale-291/system-redis\_secret.yml"  - name: create backup file all project  shell: "oc get -o yaml all > backup-3scale-291/threescale-project-elements.yaml"  # get system backup #  - include\_tasks: get\_backup\_291.yml |
| --- |

Inicialmente se crea carpeta backup-3scale donde se almacenarán los yml de respaldo y luego la ejecución de los comandos para obtenerlos, además de un include donde se almacenan los datos de respaldo del sistema.

|  | Verificar que todos los archivos generados no estén vacíos y que todos tengan sus respectivos contenidos. Modificar los archivos eliminando la referencia del namespaces. Importante al momento de utilizar este backup, se deberá borrar los atributos que se autogenerar o representan información del recurso en ese momento, por ejemplo *status, creationTimestamp, selfLink, uid, resourceVersion* |
| --- | --- |

* Descripción tareas “include\_tasks: get\_backup\_291.yml”

| # Backing up system databases #  - name: create folder backup-3ssystem291  shell: "{{ item }}"  loop:  - "mkdir backup-3scale-291/backup-3ssystem291"  - "mkdir backup-3scale-291/backup-3ssystem291/dir"  - name: backing up system-storage  shell: "oc rsync $(oc get pods -l 'deploymentConfig=system-app' -o json | jq '.items[0].metadata.name' -r):/opt/system/public/system backup-3scale-291/backup-3ssystem291/dir"  - name: backing up backend-redis  shell: "oc cp $(oc get pods -l 'deploymentConfig=backend-redis' -o json | jq '.items[0].metadata.name' -r):/var/lib/redis/data/dump.rdb backup-3scale-291/backup-3ssystem291/backend-redis-dump.rdb"  - name: backing up system-redis  shell: "oc cp $(oc get pods -l 'deploymentConfig=system-redis' -o json | jq '.items[0].metadata.name' -r):/var/lib/redis/data/dump.rdb backup-3scale-291/backup-3ssystem291/system-redis-dump.rdb"  - name: backing up zync-database  shell: "oc rsh $(oc get pods -l 'deploymentConfig=zync-database' -o json | jq -r '.items[0].metadata.name') bash -c 'pg\_dump zync\_production' | gzip > backup-3scale-291/backup-3ssystem291/zync-database-backup.gz"  # Backing up OpenShift secrets and ConfigMaps #  - name: openShift namespace secrets  shell: "{{ item }}"  loop:  - "oc get secrets system-smtp -o json > backup-3scale-291/backup-3ssystem291/system-smtp.json"  - "oc get secrets system-seed -o json > backup-3scale-291/backup-3ssystem291/system-seed.json"  - "oc get secrets system-database -o json > backup-3scale-291/backup-3ssystem291/system-database.json"  - "oc get secrets backend-internal-api -o json > backup-3scale-291/backup-3ssystem291/backend-internal-api.json"  - "oc get secrets system-events-hook -o json > backup-3scale-291/backup-3ssystem291/system-events-hook.json"  - "oc get secrets system-app -o json > backup-3scale-291/backup-3ssystem291/system-app.json"  - "oc get secrets system-recaptcha -o json > backup-3scale-291/backup-3ssystem291/system-recaptcha.json"  - "oc get secrets system-redis -o json > backup-3scale-291/backup-3ssystem291/system-redis.json"  - "oc get secrets zync -o json > backup-3scale-291/backup-3ssystem291/zync.json"  - "oc get secrets system-master-apicast -o json > backup-3scale-291/backup-3ssystem291/system-master-apicast.json"  - "oc get secrets threescale-registry-auth -o json > backup-3scale-291/backup-3ssystem291/threescale-registry-auth.json"  - name: openShift namespace configmaps  shell: "{{ item }}"  loop:  - "oc get configmaps system-environment -o json > backup-3scale-291/backup-3ssystem291/system-environment.json"  - "oc get configmaps apicast-environment -o json > backup-3scale-291/backup-3ssystem291/apicast-environment.json" |
| --- |

#### Creación de imagen amp-system oracle.

Este procedimiento lo guía a través de una actualización de Oracle Database 12c para 3scale 2.1.0 desde una instalación de 3scale 2.9.1 existente. Los archivos y drivers necesarios se encuentran sobre el repo en directorio:

| └── roles  ...  │ └── upgrade-210  │ ├── defaults  │ │ └── main.yml  │ ├── system-oracle  │ │ ├── build.yml  │ │ ├── Dockerfile  │ │ ├── dockerfile.patch  │ │ ├── oracle-client-files  **│ │ │ ├── instantclient-basic-linux.x64-19.13.0.0.0dbru-2.zip**  **│ │ │ ├── instantclient-odbc-linux.x64-19.13.0.0.0dbru-2.zip**  **│ │ │ └── instantclient-sdk-linux.x64-19.13.0.0.0dbru-2.zip**  │ │ ├── README.md  │ │ ├── workaround-non-cdb-2.4-upgrade  │ │ │ ├── database.rb  │ │ │ └── dockerfile.patch  │ │ └── workaround-non-cdb-2.5-upgrade  │ │ ├── database.rb  │ │ └── dockerfile.patch  ... |
| --- |

Se ejecuta el comando oc process con la opción -f y especifique la plantilla build.yml OpenShift y crear el objeto build para construir la imagen amp-system 2.10-oracle:

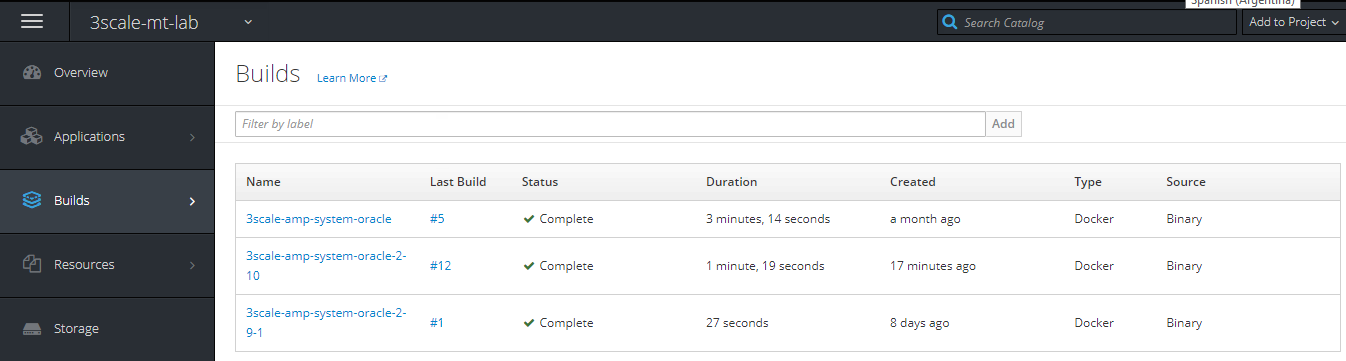
| oc process -f roles/upgrade-210/system-oracle/build.yml | oc apply -f - |
| --- |

Una vez creado se corre el siguiente comando para construir la nueva imagen del sistema con oracle:

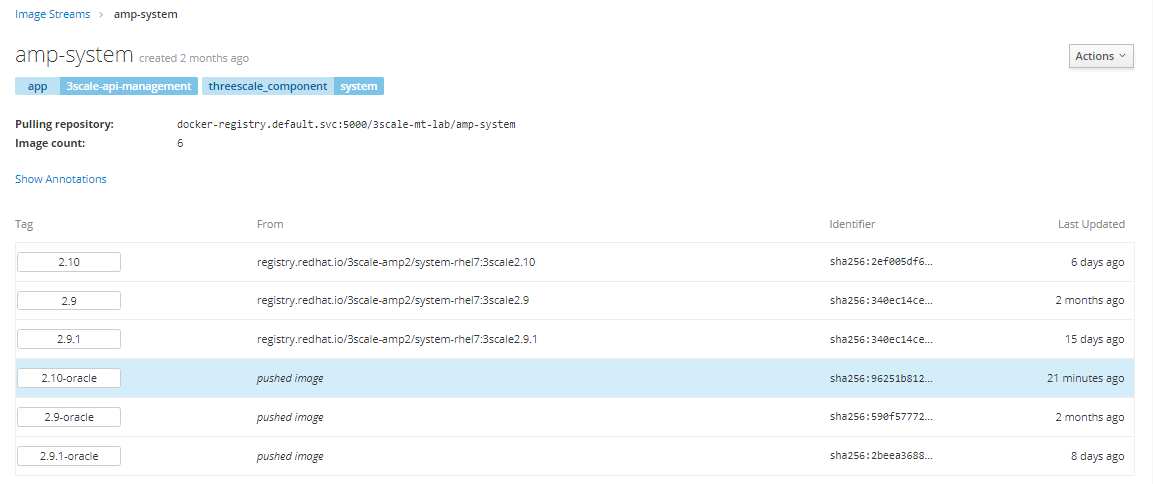
| oc start-build 3scale-amp-system-oracle-2-10 --from-dir=roles/upgrade-210/system-oracle/. -n {{ API\_MANAGER\_NS }} --wait |
| --- |

{{ API\_MANAGER\_NS }} = Variable Proyecto 3scale.

Una vez construida el Build se visualiza sobre el proyecto se puede observar de la siguiente manera:



Sobre el imagen stream se evidencia igual la creación de nuevo tag:



|  | IMPORTANTE: La pérdida de conexión a la base de datos puede dañar 3scale. Haga una copia de seguridad antes de continuar con la actualización |
| --- | --- |

A partir de este punto se generan los patch correspondientes a los ImagenStream apuntando hacia la versión 2.10 y los patch sobre cada DeploymentConfig dentro del proyecto para la generación del Upgrade.

La imagen amp-system con oracle se ocupa para los siguientes deploymentConfig:

| **deploymenConfig amp-system** |
| --- |
| system-app |
| system-sidekiq |
| system-sphinx |

Generando el patch con los siguientes comandos, removiendo el trigger correspondiente a la versión 2.9.1 y agregando Agregar el nuevo trigger específico de la versión 2.10-oracle:

* System-App:

| "oc set triggers dc/system-app --from-image=amp-system:{{ CURRENT\_ORACLE\_VERSION }} --containers=system-master,system-developer,system-provider --remove"  "oc set triggers dc/system-app --from-image=amp-system:2.10-oracle --containers=system-master,system-developer,system-provider" |
| --- |

* System-Sidekiq:

| "oc set triggers dc/system-sidekiq --from-image=amp-system:{{ CURRENT\_ORACLE\_VERSION }} --containers=system-sidekiq,check-svc --remove"  "oc set triggers dc/system-sidekiq --from-image=amp-system:2.10-oracle --containers=system-sidekiq,check-svc" |
| --- |

* System-Sphinx:

| "oc set triggers dc/system-sphinx --from-image=amp-system:{{ CURRENT\_ORACLE\_VERSION }} --containers=system-sphinx,system-master-svc --remove"  "oc set triggers dc/system-sphinx --from-image=amp-system:2.10-oracle --containers=system-sphinx,system-master-svc" |
| --- |

Por cada ajuste dentro de los deploymentConfig se agrega un task que espera el redespliegue de cada POD involucrado, haciendo referencia a un include-task hacia wait\_to\_deploy.yml con la variable correspondiente al dc que se está ajustando, el contenido de este task es:

| # wait to deploy #  - name: "wait to deploy {{pod\_to\_wait}}"  wait\_for:  timeout: 5  - name: "redeploy {{pod\_to\_wait}}"  command: 'oc rollout latest dc/"{{pod\_to\_wait}}"'  ignore\_errors: true  - name: "wait for following deployments to become ready {{pod\_to\_wait}}"  command: 'oc rollout status dc/"{{pod\_to\_wait}}"' |
| --- |

Donde *{{ pod\_to\_wait }}* es la variable que corresponde al dc que se está cambiando.

Para los siguientes componentes dentro del proyecto de 3scale se van a crear cada tag con la versión 2.10:

| **image-ref** | **url-registry** | **tag** |
| --- | --- | --- |
| amp-apicast | registry.redhat.io/3scale-amp2/apicast-gateway-rhel8:3scale2.10 | 2.10 |
| amp-backend | registry.redhat.io/3scale-amp2/backend-rhel7:3scale2.10 | 2.10 |
| amp-zync | registry.redhat.io/3scale-amp2/zync-rhel7:3scale2.10 | 2.10 |
| backend-redis | registry.redhat.io/rhscl/redis-32-rhel7:3.2 | 2.10 |
| system-memcached | registry.redhat.io/3scale-amp2/memcached-rhel7:3scale2.10 | 2.10 |
| system-mysql | registry.redhat.io/rhscl/mysql-57-rhel7:5.7 | 2.10 |
| system-redis | registry.redhat.io/rhscl/redis-32-rhel7:3.2 | 2.10 |
| zync-database-postgresql | registry.redhat.io/rhscl/postgresql-10-rhel7 | 2.10 |

Para generar el patch de cada Imagen Stream se ocupa el siguiente comando:

| oc patch imagestream/**<<image-ref>>** --type=json -p '[{\"op\": \"add\", \"path\": \"/spec/tags/-\", \"value\": {\"annotations\": {\"openshift.io/display-name\": \"Zync 2.10 PostgreSQL\"}, \"from\": { \"kind\": \"DockerImage\", \"name\": \"**<<url-registry>>**\"}, \"name\": \"2.10\", \"referencePolicy\": {\"type\": \"Source\"}}}]' |
| --- |

Por cada tag de image stream generado se deben actualizar los deploymentConfig:

| **dc-ref** | **image-ref** | **containers** |
| --- | --- | --- |
| apicast-production | amp-apicast:2.10 | apicast-production,system-master-svc |
| apicast-staging | amp-apicast:2.10 | apicast-staging |
| backend-cron | amp-backend:2.10 | backend-cron,backend-redis-svc |
| backend-listener | amp-backend:2.10 | backend-listener |
| backend-redis | backend-redis:2.10 | backend-worker,backend-redis-svc |
| backend-worker | amp-backend:2.10 | backend-worker,backend-redis-svc |
| system-memcache | system-memcached:2.10 | memcache |
| system-redis | system-redis:2.10 | system-redis |
| zync | zync:2.10 | zync,zync-db-svc |
| zync-database | zync:2.10 | n/e |
| zync-que | zync:2.10 | que |

El cambio se realiza a través del siguiente comando por cada dc en la tabla, removiendo el trigger correspondiente a la versión 2.9.1 y agregando el nuevo trigger específico de la versión 2.10:

| "oc set triggers dc/**<<dc-ref>>** --from-image=**<<image-ref>>**:{{ CURRENT\_VERSION }} --containers=**<<dc-ref>>** --remove"  "oc set triggers dc/**<<dc-is-ref>>** --from-image=**<<dimage-ref>>**:2.10 --containers=**<<containers>>**" |
| --- |

Donde *{{ CURRENT\_VERSION }}* es la versión 2.9.1 actual sobre el deployment config.

Por cada ajuste dentro de los deploymentConfig se agrega un task que espera el redespliegue de cada POD involucrado, haciendo referencia a un include-task hacia wait\_to\_deploy.yml.

### Upgrade apicast 3scale-b2c-gw

Una vez generado el upgrade sobre el proyecto 3scale se agrega un include task para generar el upgrade dentro del proyecto b2c donde se ubican los apicast:

| # updgrade apicast 2.9.1 to 2.10 #  - include\_tasks: upgrade\_apicast.yml |
| --- |

Dentro del playbook **upgrade\_apicast**.**yml** se ejecutan los comandos de patch sobre el deployment sobre el apicast staging y de production apuntando directamente hacia el registry con el tag 2.10 ya que estos no poseen ImageStream; los comando ejecutados son los siguientes:

#### Generación de Backups.

En este paso se ejecutan los siguientes comandos sobre el playbook para obtenerlos:

| - name: create backup file with DeploymentConfig  shell: "{{ item }}"  loop:  - "mkdir backup-3scale-gw-291"  - "mkdir backup-3scale-gw-291/threescale-elements"  - "for component in {{THREESCALE\_DC\_NAMES\_GW}}; do oc get -o yaml dc ${component} > backup-3scale-gw-291/${component}\_dc.yml ; done"  - "oc get -o yaml all > backup-3scale-gw-291/threescale-elements/threescale-project-elements.yaml"  - "for object in rolebindings serviceaccounts secrets imagestreamtags cm rolebindingrestrictions limitranges resourcequotas pvc templates cronjobs statefulsets hpa deployments replicasets poddisruptionbudget endpoints; do oc get -o yaml $object > backup-3scale-gw-291/threescale-elements/$object.yaml; done"  - "for object in `oc get routes | awk '{print $1}' | grep -v NAME`; do oc get -o yaml route ${object} > backup-3scale-gw-291/${object}\_route.yaml; done"  - name: create backup file all project  shell: "oc get -o yaml all > backup-3scale-gw-291/threescale-project-elements.yaml" |
| --- |

#### Upgrade image apicasts

Para generar el patch de cada Imagen Stream se ocupa el siguiente comando:

* Apicast-staging:

| oc patch deployment stage-apicast --patch='{\"spec\":{\"template\":{\"spec\":{\"containers\":[{\"name\": \"stage-apicast\", \"image\":\"registry.redhat.io/3scale-amp2/apicast-gateway-rhel8:3scale2.10\"}]}}}}' |
| --- |

* Apicast-production:

| oc patch deployment prod-apicast --patch='{\"spec\":{\"template\":{\"spec\":{\"containers\":[{\"name\": \"prod-apicast\", \"image\":\"registry.redhat.io/3scale-amp2/apicast-gateway-rhel8:3scale2.10\"}]}}}}'" |
| --- |

Por cada ajuste dentro de los deploymentConfig se agrega un task que espera el redespliegue de cada POD involucrado, haciendo referencia a un include-task hacia wait\_to\_deploy.yml.

Nota: Con respecto al upgrade de los Apicast de STG/QA se generó un playbook que recorriera los 3 namespace que pertenecen a este ambiente en su actualización, los cuales son:

* 3scale-b2c-homo-gw
* 3scale-b2b-inte-gw
* 3scale-b2c-inte-gw
* Playbook upgrade apicast STG/QA:

| ---  # ################################ project 3scale-b2c-homo-gw ##################################### #  - name: check namespace 3scale b2c homo gw  shell: "oc project 3scale-b2c-homo-gw"  register: api\_project\_location\_gw    - name: namespace name  debug:  msg: "{{ api\_project\_location\_gw }}"  # ########## create backup project  ---  # patch the "apicast" image #  # ################################ project 3scale-b2b-inte-gw ##################################### #  - name: check namespace 3scale b2b inte gw  shell: "oc project 3scale-b2b-inte-gw"  register: api\_project\_location\_gw    - name: namespace name  debug:  msg: "{{ api\_project\_location\_gw }}"  # ########## create backup project  ---  # patch the "apicast" image #  –--  # ################################ project 3scale-b2c-inte-gw ##################################### #  - name: check namespace 3scale b2c inte gw  shell: "oc project 3scale-b2c-inte-gw"  register: api\_project\_location\_gw    - name: namespace name  debug:  msg: "{{ api\_project\_location\_gw }}"  # ########## create backup project  # create backup deploymentConfig y rutas #  ---  # patch the "apicast" image #  --- |
| --- |

# 

# Downgrade de 2.10 a 2.9.1

En caso de ser requerido y durante el proceso de ejecución de upgrade, existe un playbook de ANSIBLE que permite la vuelta atrás en versión de 3scale hacia la 2.9.1, el playbook se ubica sobre el siguiente directorio:

| 3scale-upgrade-galicia/3scale-upgrade-291-to-210/  ...  ├── downgrade-3scale-210-to-291.yml  └── roles  ├── downgrade-210-to-291  │ ├── defaults  │ │ └── main.yml  │ └── tasks  │ ├── main.yml  │ └── wait\_for\_deploy\_d.yml  ... |
| --- |

## Detalle Tasks Downgrade

### Variables lab

Las variables declaradas para la ejecución del playbook sobre ambiente LAB:

| ---  # 3scale images versions #  CURRENT\_VERSION: 2.10  CURRENT\_ORACLE\_VERSION: 2.10-oracle  TO\_VERSION: 2.9.1  TO\_ORACLE\_VERSION: 2.9.1-oracle  # 3scale mt - VARS #  ENV\_OCP: lab  PROJECT\_NAME: 3scale-mt  API\_MANAGER\_NS: "{{PROJECT\_NAME}}-{{ENV\_OCP}}"  # 3scale apicast - VARS #  ENV\_OCP\_GW: gw  PROJECT\_NAME\_GW: 3scale-b2c  API\_MANAGER\_NS\_GW: "{{PROJECT\_NAME\_GW}}-{{ENV\_OCP\_GW}}" |
| --- |

### Select Project 3scale

Para iniciar el proceso de upgrade el playbook se posiciona sobre el proyecto correspondiente a través del comando:

| oc project {{ API\_MANAGER\_NS }} |
| --- |

### Tareas de downgrade

El proceso de downgrade se genera teniendo en cuenta que se toman los tags de ImageStream en la versión 2.10 en el proyecto, para system-app se toma el tag 2.10-oracle:

| **dc-is** | **tag** |
| --- | --- |
| system-app | 2.10-oracle |
| system-sidekiq | 2.10-oracle |
| system-sphinx | 2.10-oracle |

Para el downgrade de los componentes con amp-system se ocupan los siguiente comandos:

| ***system-app***  "oc set triggers dc/system-app --from-image=amp-system:2.10--containers=system-master,system-developer,system-provider --remove"  "oc set triggers dc/system-app --from-image=amp-system:{{ TO\_ORACLE\_VERSION }} --containers=system-master,system-developer,system-provider" |
| --- |
| ***system-sidekiq***  "oc set triggers dc/system-sidekiq --from-image=amp-system:2.10 --containers=system-sidekiq,check-svc --remove"  "oc set triggers dc/system-sidekiq --from-image=amp-system:2.10 |
| ***system-sidekiq***  "oc set triggers dc/system-sphinx --from-image=amp-system:2.10--containers=system-sphinx,system-master-svc --remove"  "oc set triggers dc/system-sphinx --from-image=amp-system:{{ TO\_ORACLE\_VERSION }} --containers=system-sphinx,system-master-svc" |

Vars:

{{ TO\_ORACLE\_VERSION }} = Version que se insertará en el dc 2.9.1-oracle

| **dc-ref** | **image-ref** | **containers** |
| --- | --- | --- |
| apicast-production | amp-apicast:2.10 | apicast-production,system-master-svc |
| apicast-staging | amp-apicast:2.10 | apicast-staging |
| backend-cron | amp-backend:2.10 | backend-cron,backend-redis-svc |
| backend-listener | amp-backend:2.10 | backend-listener |
| backend-redis | backend-redis:2.10 | backend-worker,backend-redis-svc |
| backend-worker | amp-backend:2.10 | backend-worker,backend-redis-svc |
| system-memcache | system-memcached:2.10 | memcache |
| system-redis | system-redis:2.10 | system-redis |
| zync | zync:2.10 | zync,zync-db-svc |
| zync-database | zync:2.10 | n/e |
| zync-que | zync:2.10 | que |

Para el downgrade de los componentes en la anterior tabla se ocupa el siguiente comando:

| "oc set triggers dc/<<dc-ref>> --from-image=<<image-ref>>:{{ CURRENT\_VERSION }} --containers=<<dc-ref>> --remove"  "oc set triggers dc/<<dc-ref>> --from-image=<<image-ref>>:{{ TO\_VERSION }} --containers=<<containers>>" |
| --- |

Por cada ajuste dentro de los deploymentConfig se agrega un task que espera el redespliegue de cada POD involucrado, haciendo referencia a un include-task hacia wait\_to\_deploy.yml.

### Downgrade apicast 3scale-b2c-gw

#### Select Project 3scale

Para iniciar el proceso de upgrade el playbook se posiciona sobre el proyecto correspondiente a los gw a través del comando:

| oc project {{ API\_MANAGER\_NS\_GW }} |
| --- |

#### Downgrade image apicasts:

Para generar el patch de cada Imagen Stream se ocupa el siguiente comando:

* Apicast-staging:

| oc patch deployment stage-apicast --patch='{\"spec\":{\"template\":{\"spec\":{\"containers\":[{\"name\": \"stage-apicast\", \"image\":\"registry.redhat.io/3scale-amp2/apicast-gateway-rhel8:3scale2.9.1\"}]}}}}' |
| --- |

* Apicast-production:

| oc patch deployment prod-apicast --patch='{\"spec\":{\"template\":{\"spec\":{\"containers\":[{\"name\": \"prod-apicast\", \"image\":\"registry.redhat.io/3scale-amp2/apicast-gateway-rhel8:3scale2.9.1\"}]}}}}'" |
| --- |

Por cada ajuste dentro de los deploymentConfig se agrega un task que espera el redespliegue de cada POD involucrado, haciendo referencia a un include-task hacia wait\_to\_deploy.yml.

Nota: Con respecto a STG/QA dentro del playbook de downgrade recorrerá los 3 namespace que pertenecen a este ambiente, los cuales son:

* 3scale-b2c-homo-gw
* 3scale-b2b-inte-gw
* 3scale-b2c-inte-gw

## Upgrade Red Hat SSO 7.3 a 7.4

El playbook para generar el upgrade del SSO es **upgrade-sso-to74.yml** se encuentra en el repo:

https://github.bancogalicia.com.ar/tytm-middleware/update-ocp311-sso

| **upgrade-sso-to74$**  .  ├── README.md  ├── roles  │ └── upgrade-sso-74  │ ├── build  │ │ ├── Dockerfile  │ │ └── extensions  │ │ ├── actions.cli  │ │ └── ojdbc8.jar  │ ├── data  │ │ ├── build-rhsso-oracle-19c.yml  │ │ └── imagestream-sso74.yml  │ ├── defaults  │ │ └── main.yml  │ └── tasks  │ └── main.yml  ├── sso-downgrade-74-to-73.yml  └── sso-upgrade-73-to-74.yml |
| --- |

## Configurar datos de conexión de base de datos oracle 19c

Sobre el archivo action.cli ubicado en roles/upgrade-sso-74/build/extensions/actions.cli se debe editar la siguiente línea:

| /subsystem=datasources/data-source=KeycloakDS:add(jndi-name=java:jboss/datasources/KeycloakDS,enabled=true,use-java-context=true,connection-url="jdbc\:oracle\:thin\:@(DESCRIPTION\=(ADDRESS\=(PROTOCOL\=TCP)(HOST\=<<HOST\_DE\_BASE\_DE\_DATOS>>)(PORT\=1521))(CONNECT\_DATA\=(SERVICE\_NAME\=<<ESQUEMA\_BASE\_DE\_DATOS>>)))",driver-name=oracle,user-name=<<USER\_NAME>>,password=<<PASSWORD>> |
| --- |

Las variables que se muestran en la tabla con el '<<...>>' se deben editar y corresponden a los datos de conexión de la base de datos Oracle en su versión 19c, importante mencionar que antes de la construcción de la imagen se debe generar una copia de la Base de Datos 12c hacia la Base de datos 19c así pueda conectarse de manera efectiva.

Se verifica que el nombre del proyecto sea el correspondiente

| oc project {{ SSO\_NS }} |
| --- |

## Detalle de task dentro de playbook upgrade SSO:

**Playbook:**  *roles/upgrade-sso-74/tasks/main.yml*

Cree una imagestream de sso74-openshift

| oc create -f roles/upgrade-sso-74/data/imagestream-sso74.yml -n {{ SSO\_NS }} |
| --- |

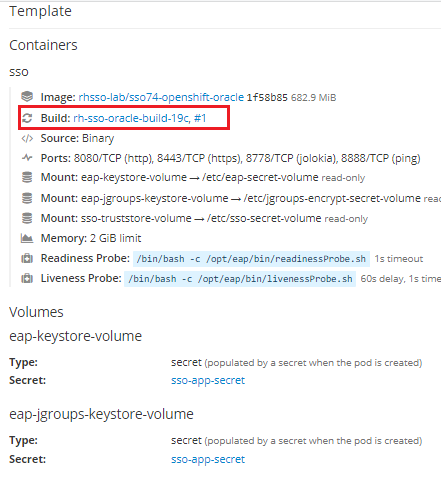
Cree la buildConfig rh-sso-oracle-build-19c

| oc create -f roles/upgrade-sso-74/data/build-rhsso-oracle-19c.yml -n {{ SSO\_NS }} |
| --- |

Se debe Compilar la imagen desde Dockerfile

| oc start-build rh-sso-oracle-build-19c --from-dir=roles/upgrade-sso-74/build -n {{ SSO\_NS }} --wait |
| --- |

Evidencia creación de imagen custom con oracle:



Verificar la configuración de implementación utilizada para implementar los contenedores

| oc get dc -o name --selector=application=sso -n {{ SSO\_NS }} |
| --- |

Actualizar el activador de imagen en la configuración de implementación existente a la imagen de Oracle RH-SSO 7.4

| oc patch dc/sso --type=json -p '[{\"op\": \"replace\", \"path\": \"/spec/triggers/0/imageChangeParams/from/name\", \"value\": \"sso74-openshift-oracle:1.0\"}]' |
| --- |

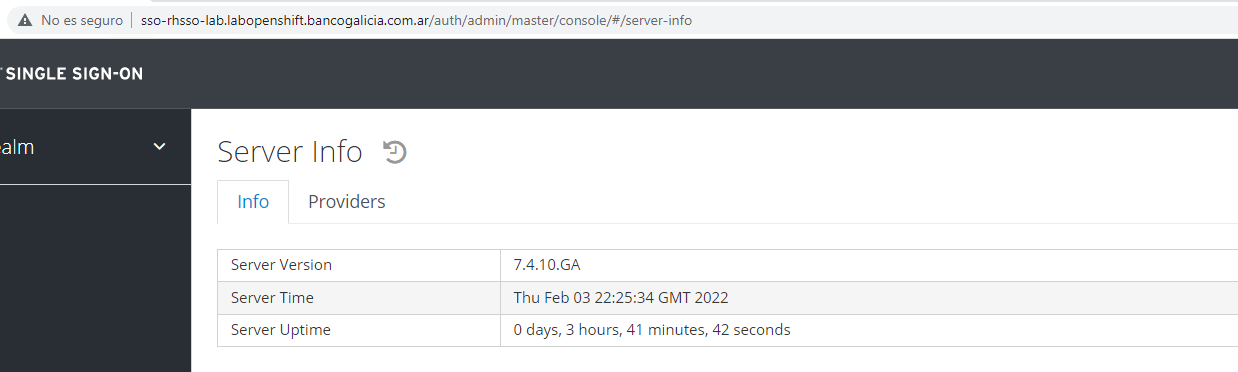
Realizar un relanzamiento del POD dc/sso

| oc rollout latest dc/sso -n {{ SSO\_NS }} |
| --- |

Esperar a que los POD del sso complete el nuevo despliegue

| oc rollout status dc/sso -n {{ SSO\_NS }} |
| --- |

Evidencia actualización a versión 7.4 - lab:



Para más información puede consultar la documentación oficial de redhat en el siguiente link :

<https://access.redhat.com/documentation/en-us/red_hat_single_sign-on/7.4/html/red_hat_single_sign-on_for_openshift_on_openjdk/tutorials>

# Downgrade Red Hat SSO 7.3 a 7.4

En caso de ser requerido y durante el proceso de ejecución de upgrade, existe un playbook de ANSIBLE que permite la vuelta atrás en la versión del SSO 7.3.

## Detalle taks en playbook downgrade:

**Playbook:** /sso-downgrade-74-to-73.yml

Se debe actualizar el activador de la imagen en la configuración de implementación existente a la imagen de Oracle RH-SSO 7.3

| oc patch dc/sso --type=json -p '[{\"op\": \"replace\", \"path\": \"/spec/triggers/0/imageChangeParams/from/name\", \"value\": \"sso73-openshift:latest\"}]' |
| --- |

Realice un relanzamiento del POD dc/sso

| oc rollout latest dc/sso |
| --- |

Espere a que los POD del sso complete el nuevo despliegue

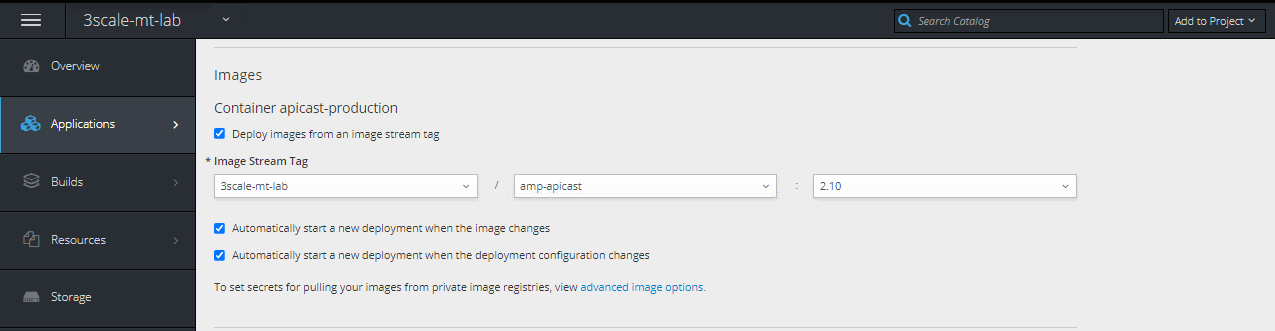
| oc rollout status dc/sso |
| --- |

# Evidencias upgrade de ambientes.

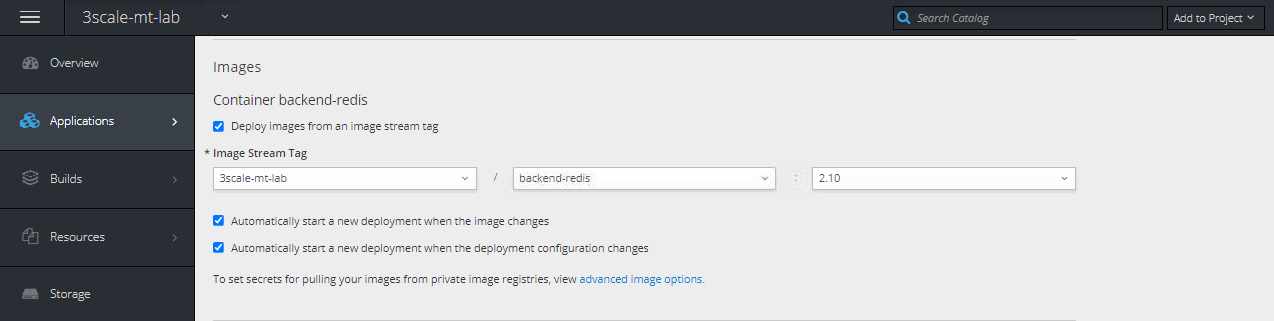
## Upgrade 3scale 2.9 a 2.10 - Ambiente Laboratorio.

### Upgrade apicast 3scale-mt-lab

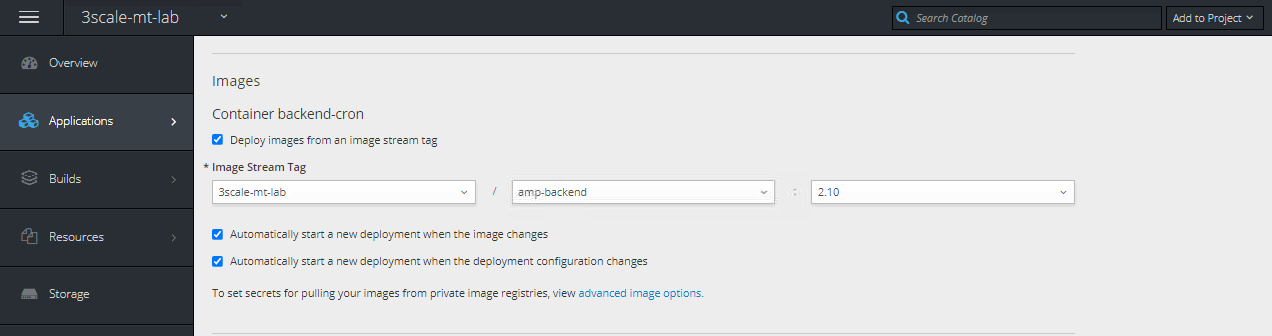
* Apicast-production: Deployment config **Apicast-production** versión 2.10



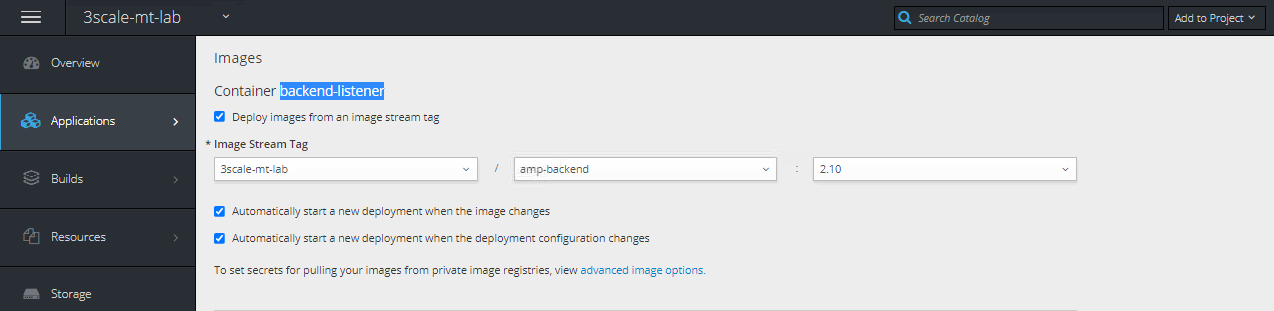
* Apicast-staging: Deployment config **Apicast-staging** versión 2.10



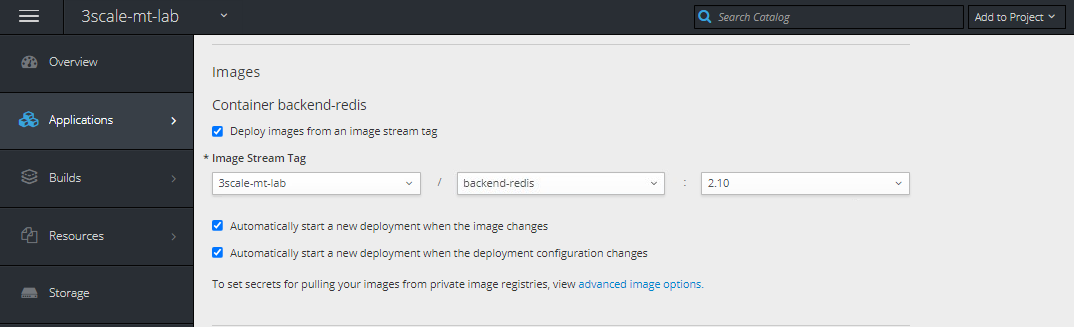
* Backend-cron: Deployment config **Backend-cron** versión 2.10



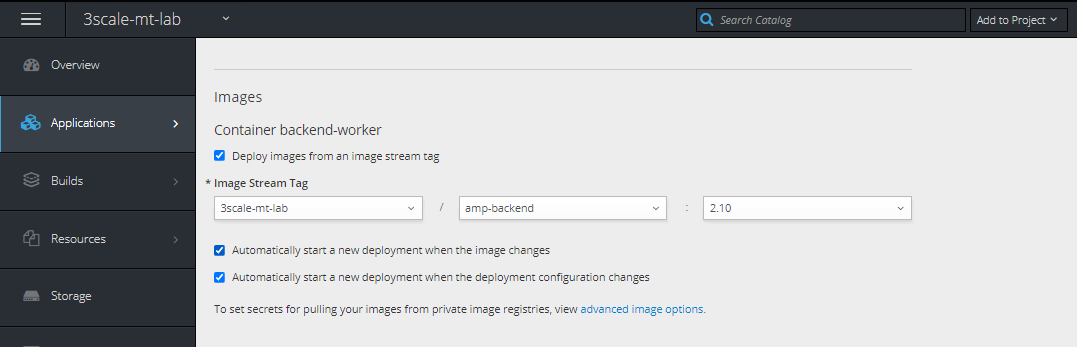
* Backend-listener: Deployment config **Backend-listener** versión 2.10



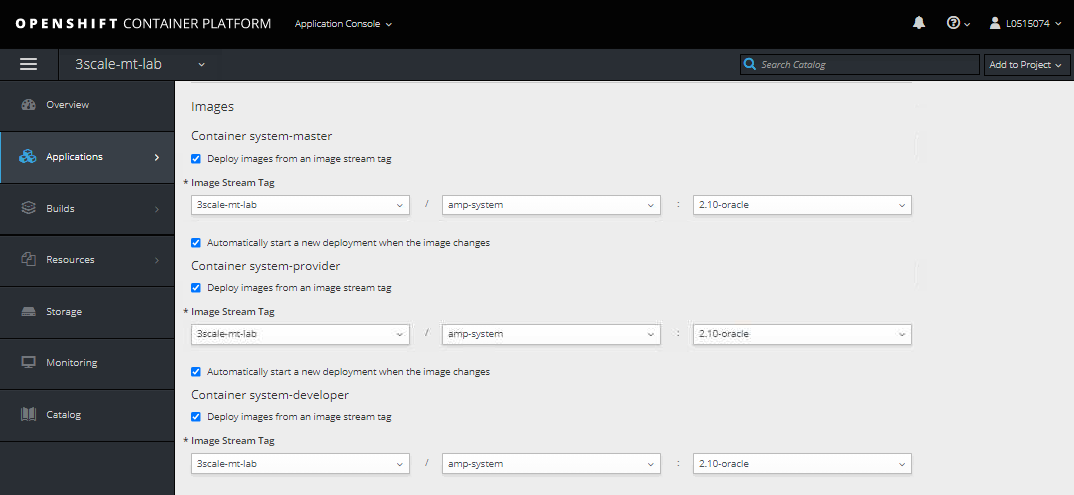
* Backend-redis: Deployment config **Backend-redis** versión 2.10



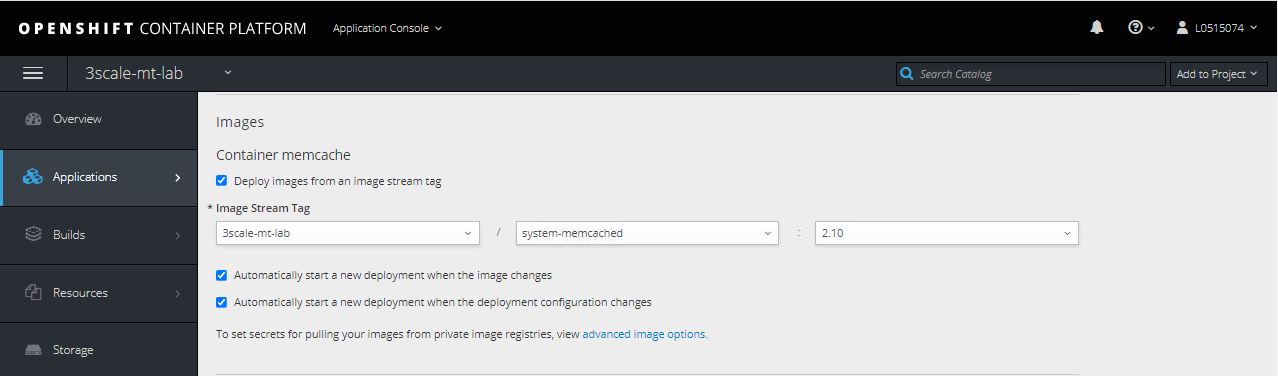
* Backend-worker: Deployment config **Backend-worke**r versión 2.10



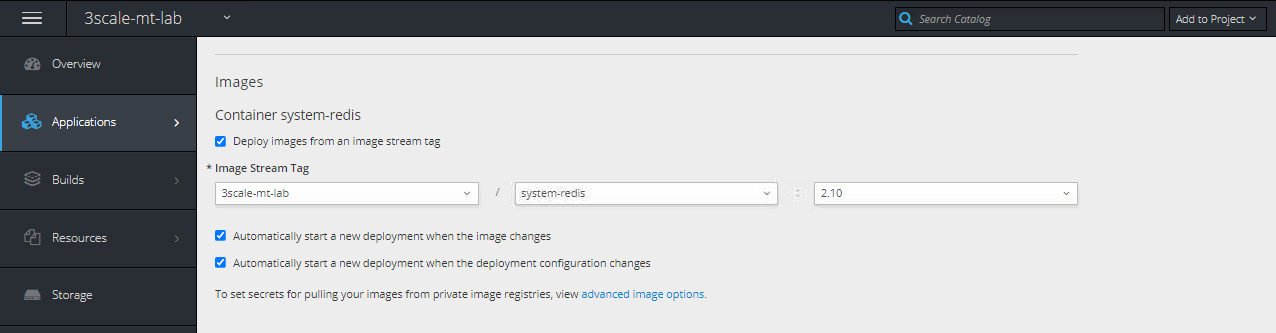
* System-app: Deployment config **System-app** versión 2.10



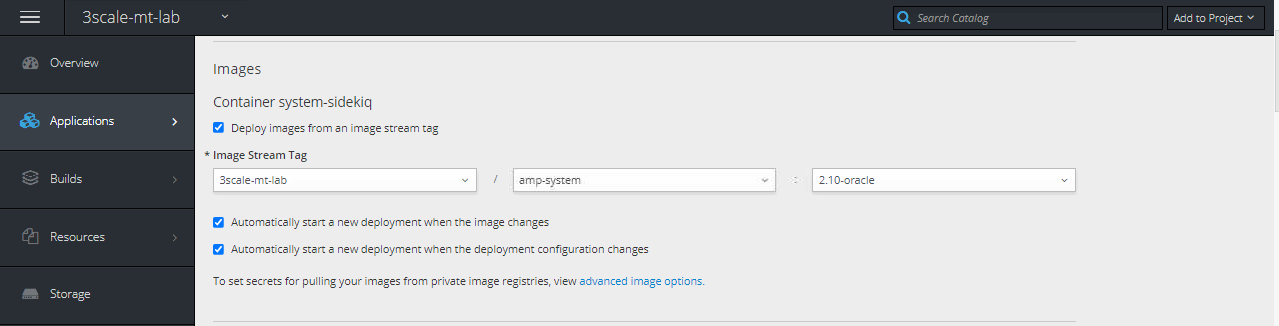
* System-memcache: Deployment config **System-memcache** versión 2.10



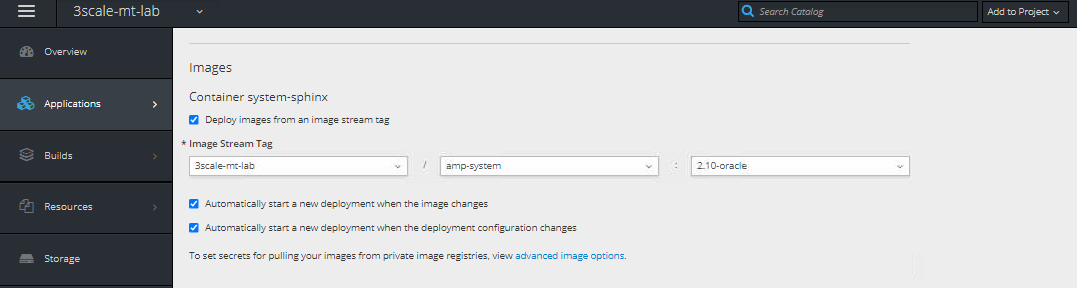
* System-redis: Deployment config **System-redis** versión 2.10



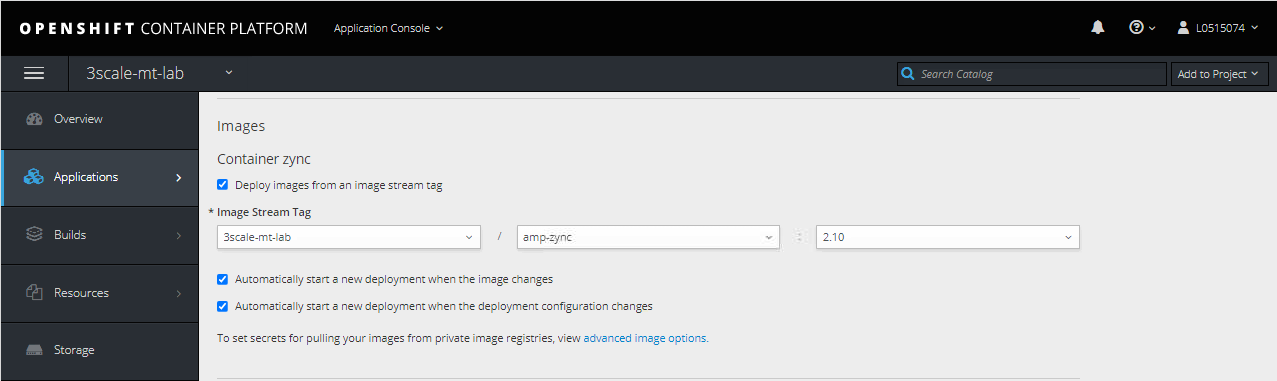
* System-sidekiq: Deployment config **System-sidekiq** versión 2.10



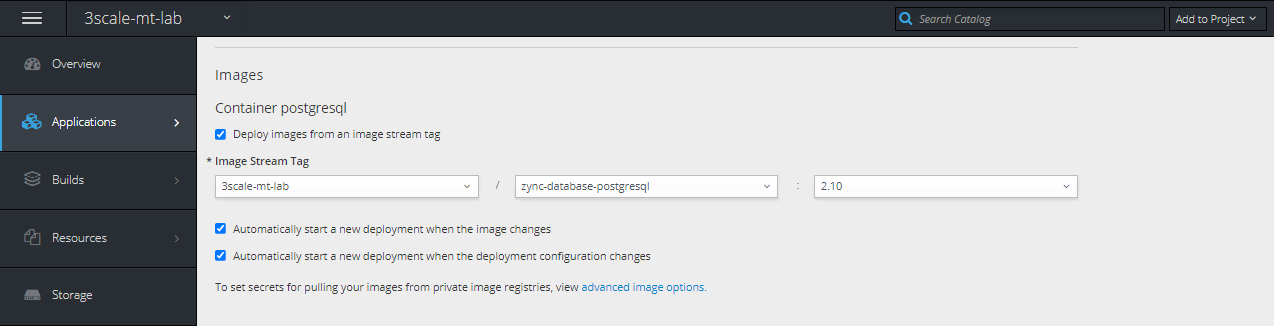
* System-sphinx: Deployment config **System-sphinx** versión 2.10



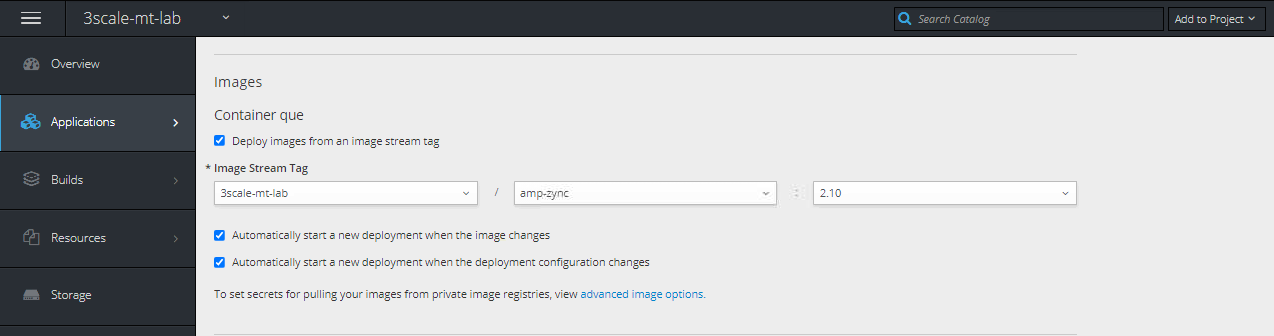
* Zync: Deployment config **Zync** versión 2.10



* Zync-database: Deployment config **Zync-database** versión 2.10

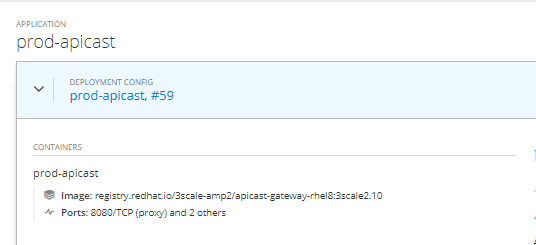


* Zync-que: Deployment config **Zync-que** versión 2.10

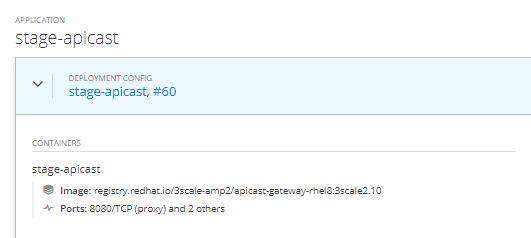


### Upgrade apicast 3scale-b2c-gw laboratorio

* Prod-apicast:

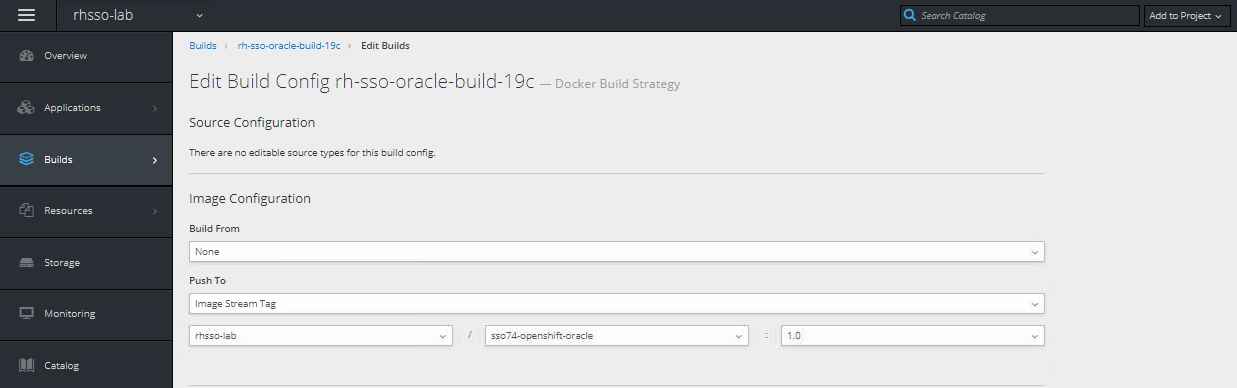


* Stage-apicast:

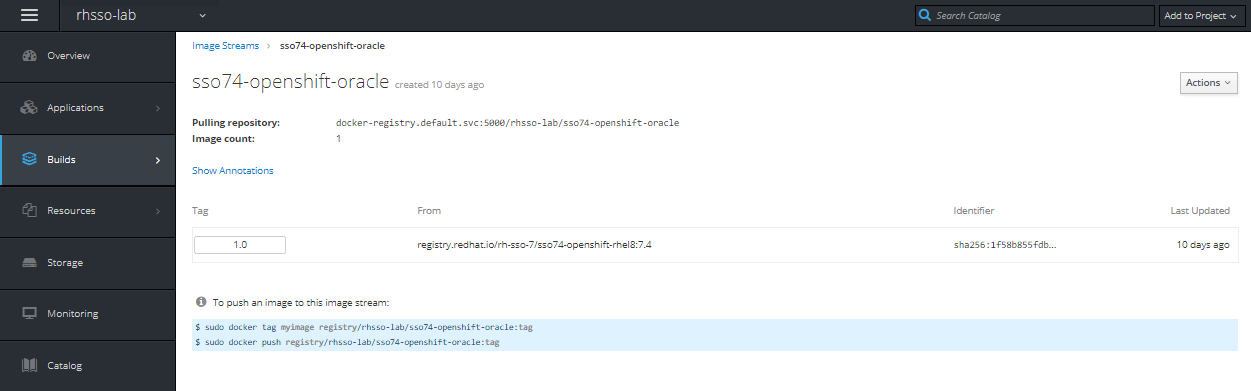


## Upgrade SSO 7.3 a 7.4 - Ambiente Laboratorio.

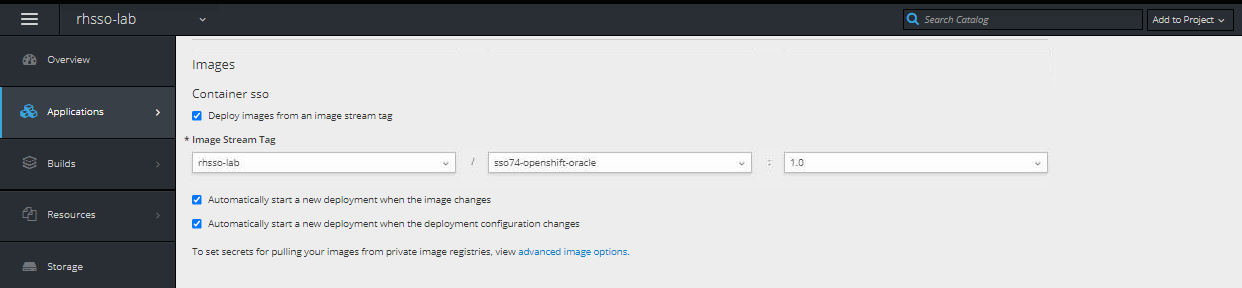
* Build de la imagen SSO



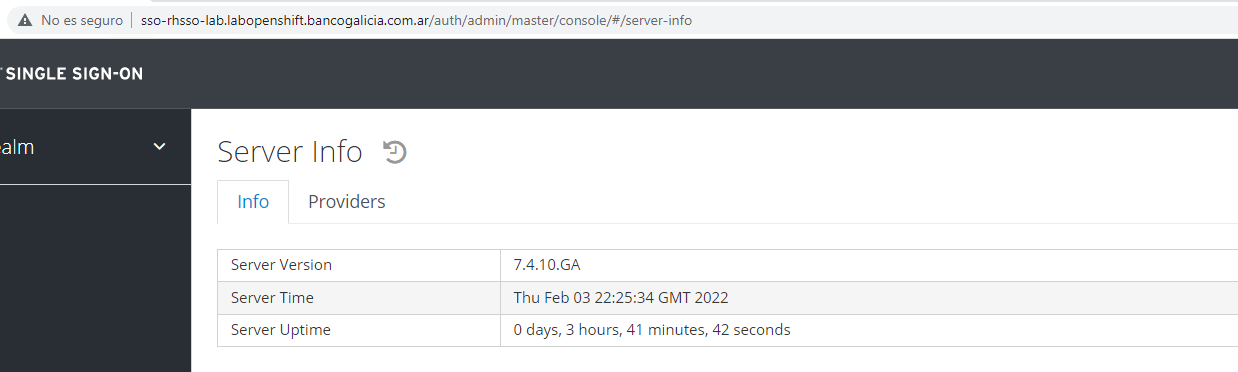
* Image Stream del 7.4



* Deploymentconfig



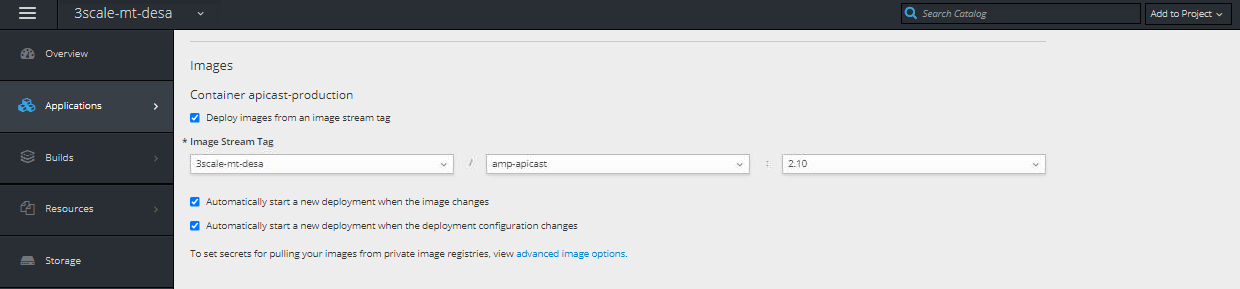
* Versión 7.4 en la herramienta



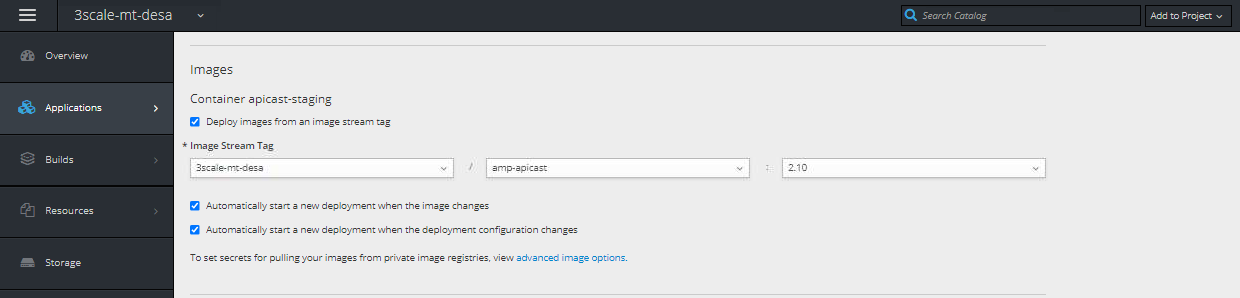
## Upgrade 3scale 2.9 a 2.10 - Ambiente Desarrollo

### Upgrade apicast 3scale-mt-desa

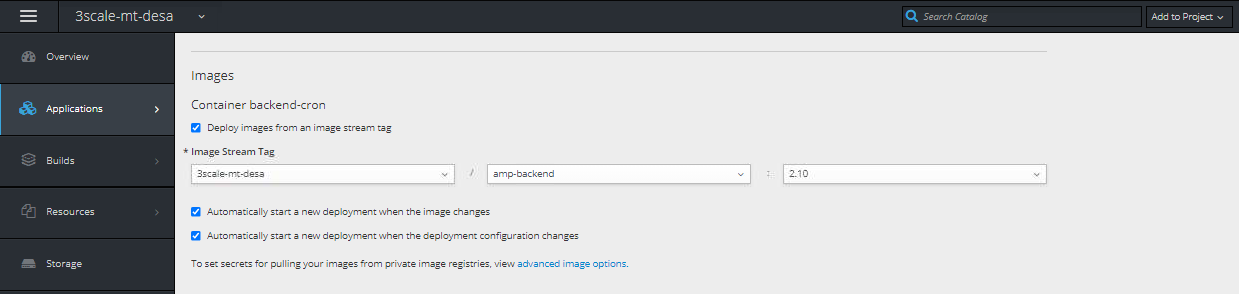
* Apicast-production: Deployment Config **Apicast-production** versión 2.10



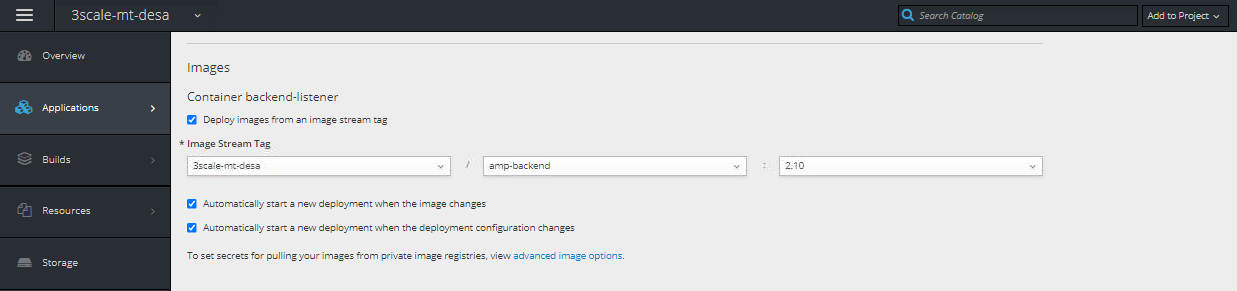
* Apicast-staging: Deployment config **Apicast-staging** versión 2.10



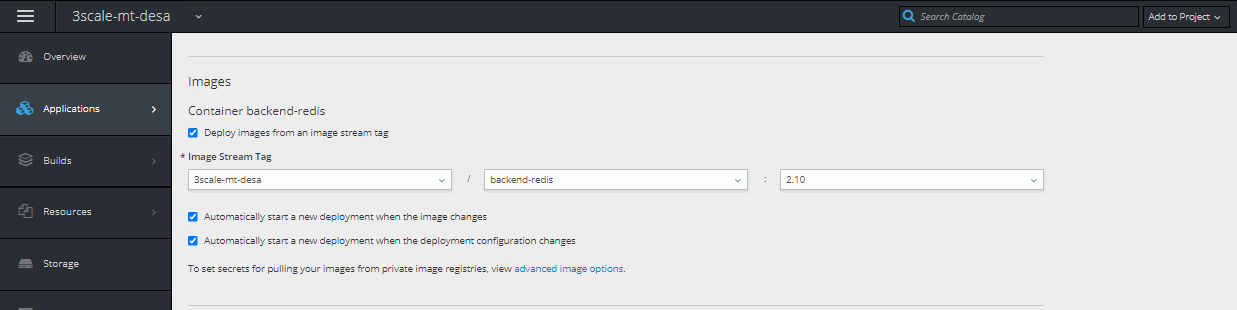
* Backend-cron: Deployment config **Backend-cron** versión 2.10



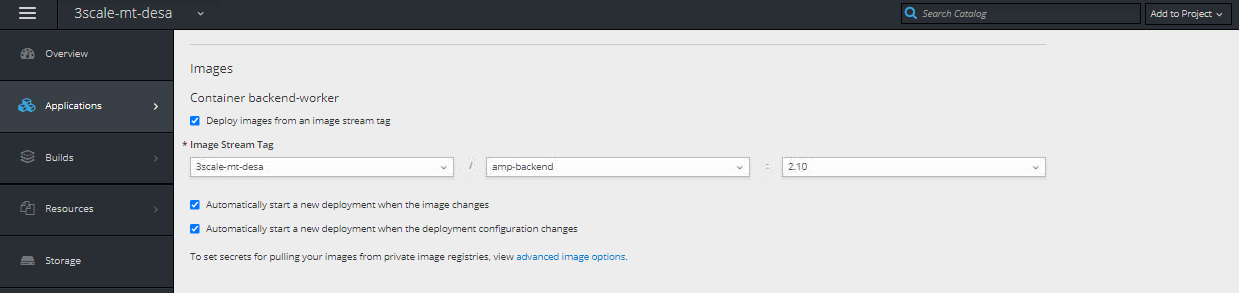
* Backend-listener: Deployment Config **Backend-listener** versión 2.10



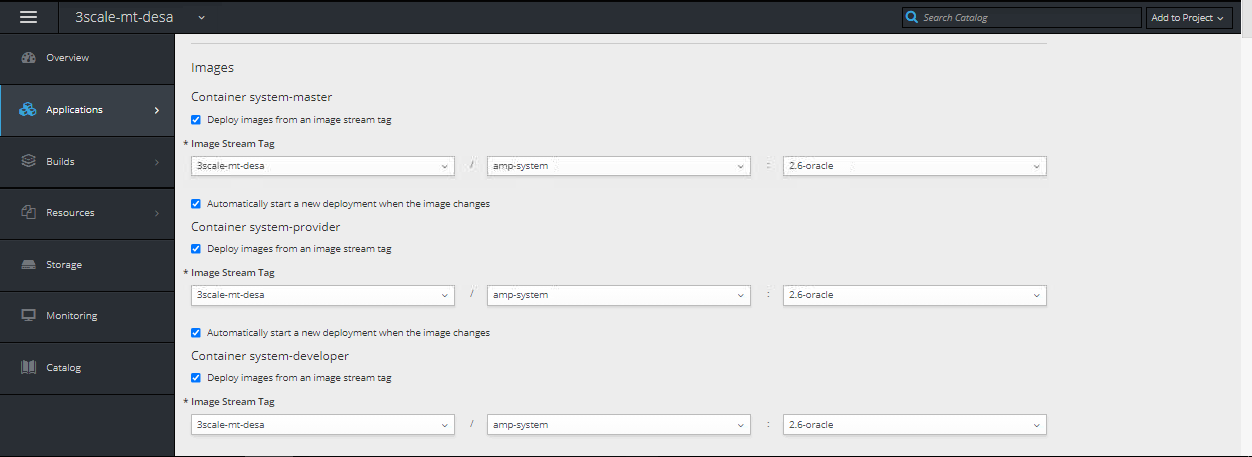
* Backend-redis: Deployment Config **Backend-redis** versión 2.10



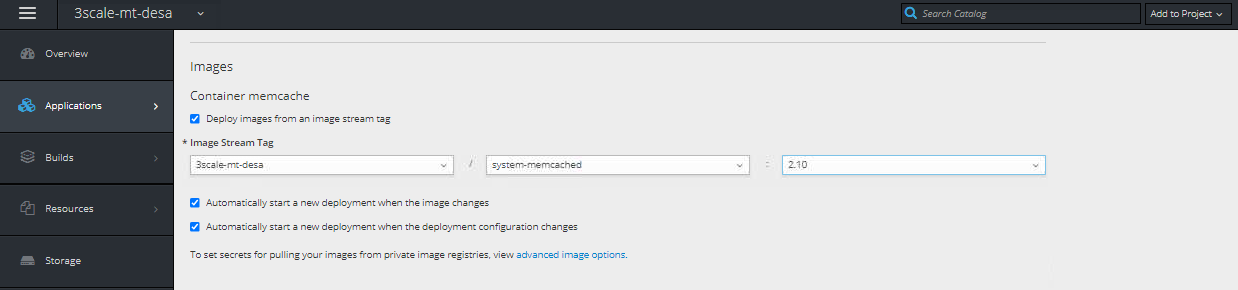
* Backend-worker: Deployment Config **Backend-worker** versión 2.10



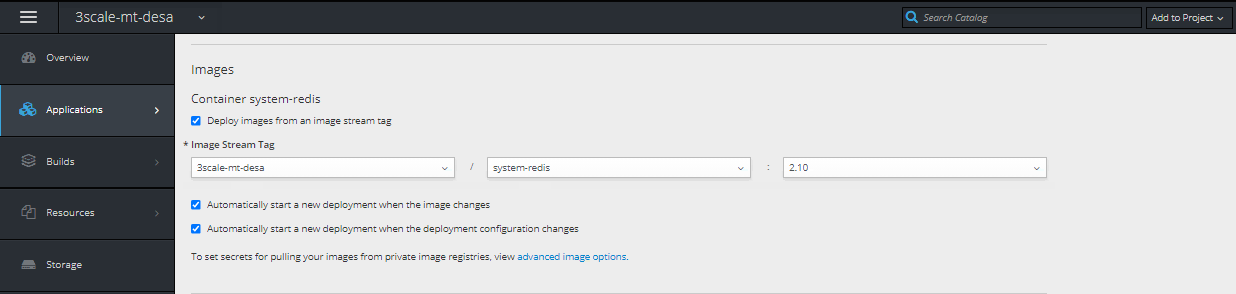
* System-app: Deployment Config **System-app** versión 2.10



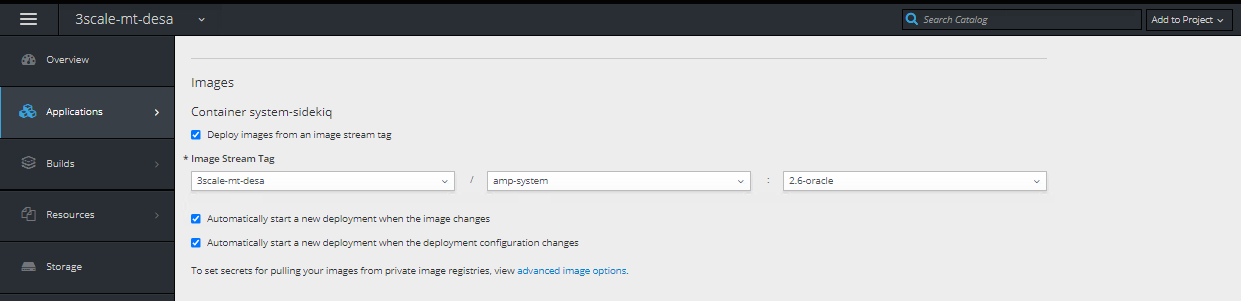
* System-memcache: Deployment Config **System-memcache** versión 2.10



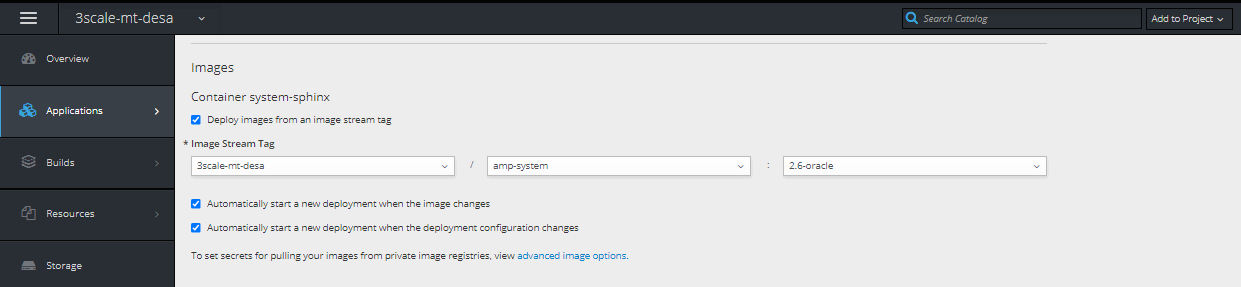
* System-redis: Deployment Config **System-redis** versión 2.10



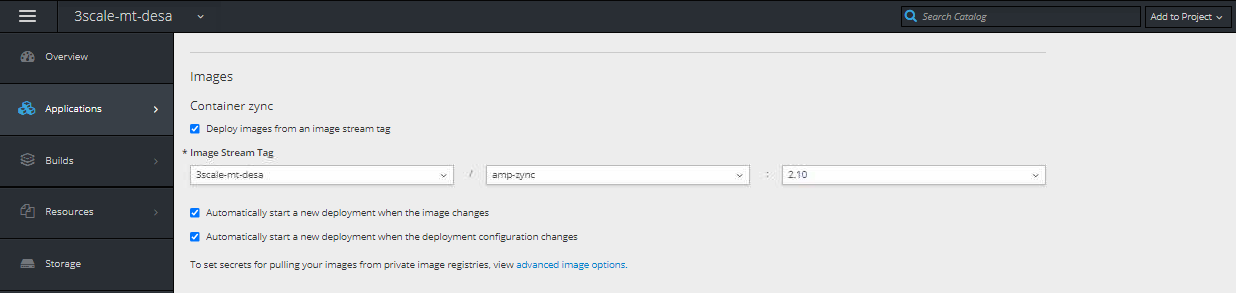
* System-sidekiq: Deployment Config **System-sidekiq** versión 2.10



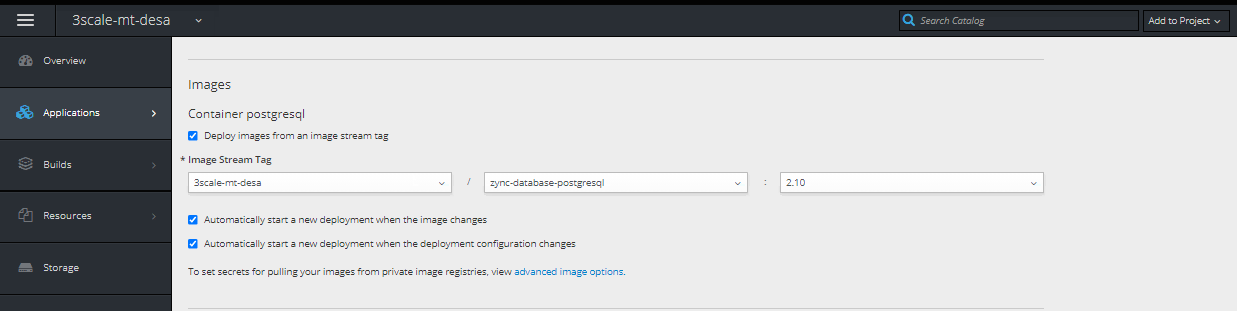
* System-sphinx: Deployment Config **System-sphinx** versión 2.10



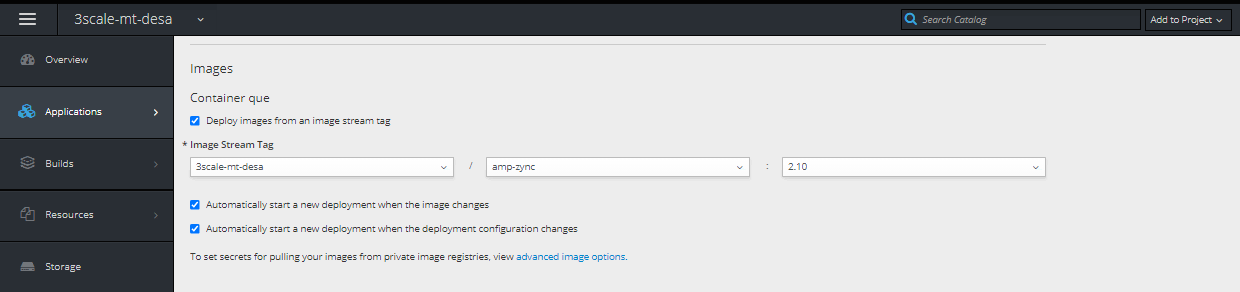
* Zync: Deployment Config **Zync** versión 2.10



* Zync-database: Deployment Config **Zync-database** versión 2.10

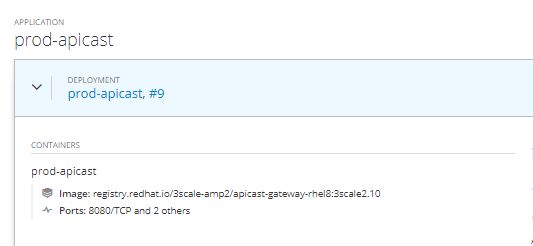


* Zync-que: Deployment Config **Zync-que** versión 2.10

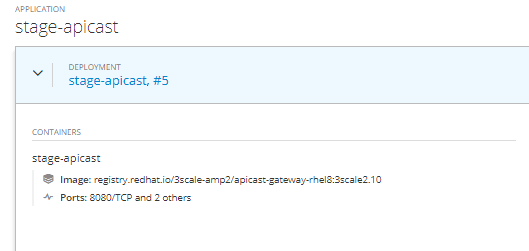


### Upgrade apicast 3scale-b2c-gw - Desarrollo

* Prod-apicast:

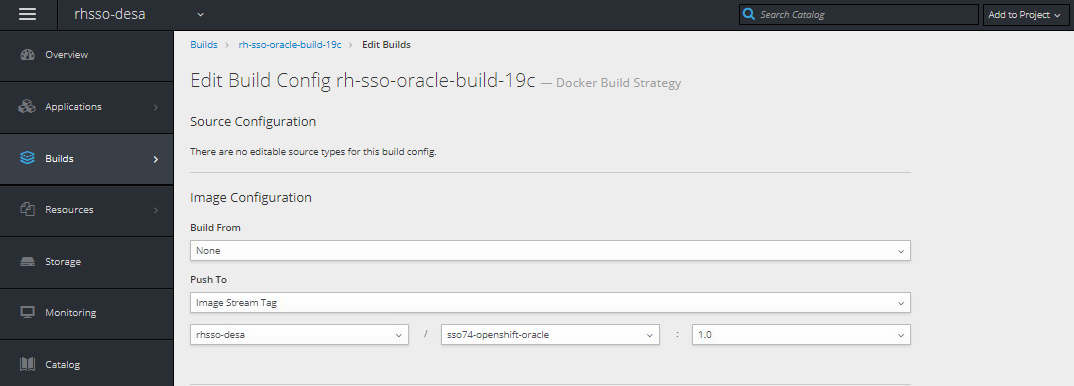


* Stage-apicast:

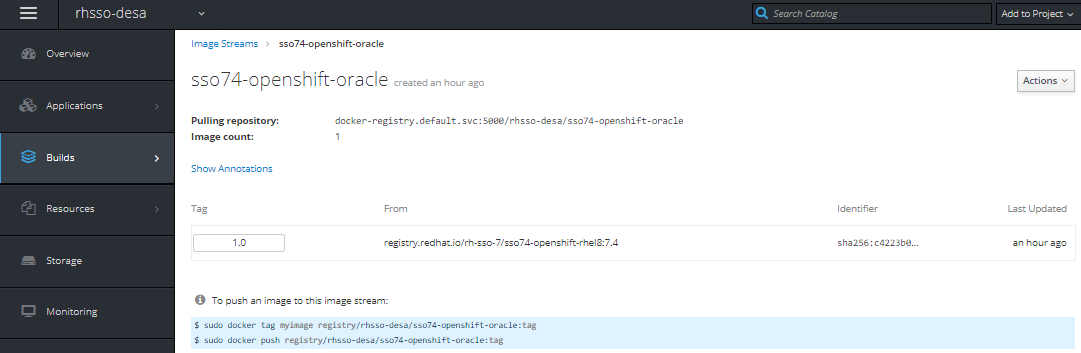


# Upgrade SSO 7.3 a 7.4 - Ambiente Desarrollo

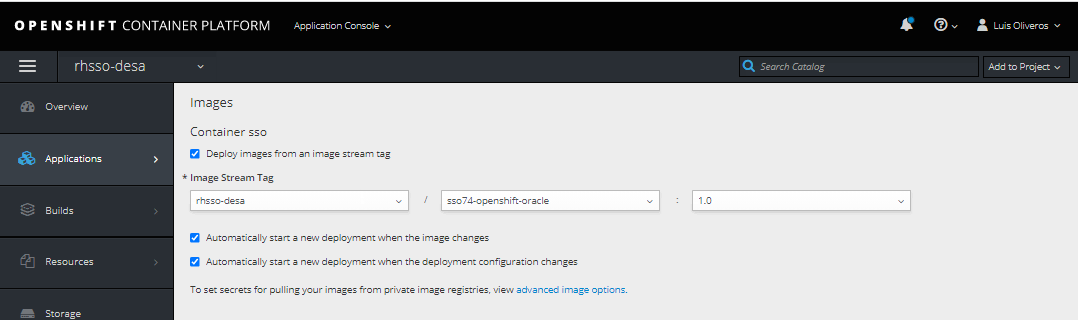
* Build de la imagen SSO:



* Image Stream del 7.4:



* Deploymentconfig:



* Versión 7.4 en la herramienta:

