Red Hat Quay 3.11

Red Hat Quay 故障排除

Red Hat Quay 故障排除

Last Updated: 2024-06-06

Red Hat Quay 3.11 Red Hat Quay 故障排除

Red Hat Quay 故障排除

法律通告

摘要

Red Hat Quay 故障排除

目录

前言

红帽提供了用于为您的 Red Hat Quay 部署收集数据的管理员工具。您可以使用此数据自行对 Red Hat Quay 部署进行故障排除，或者提交一个支持问题单。

第 1 章 获取支持

如果您在执行本文档所述的某个流程或 Red Hat Quay 时遇到问题，请访问红帽客户门户。通过红帽客户门户网站：

- 搜索或者浏览红帽知识库，了解与红帽产品相关的文章和解决方案。
- 提交问题单给红帽支持。
- 访问其他产品文档。

要识别部署的问题，您可以使用 Red Hat Quay 调试工具，或检查部署的健康端点来获取您的问题的信息。调试或获取部署的健康状况信息后，您可以搜索红帽知识库中的解决方案或提交支持问题单。

如果您对本文档有任何改进建议，或发现了任何错误，请向 **ProjectQuay** 项目提交 JIRA 问题。提供具体信息，如部分名称和 Red Hat Quay 版本。

1.1. 关于红帽知识库

红帽知识库提供丰富的内容以帮助您最大程度地利用红帽的产品和技术。红帽知识库包括文章、产品文档和视频，概述了安装、配置和使用红帽产品的最佳实践。另外，您还可以搜索已知问题的解决方案，其提供简洁的根原因描述和补救措施。

Red Hat Quay 支持团队还为 Red Hat Quay 维护一个 Consolidate 故障排除文章，该 文章详细介绍了常见问题的解决方案。这是一个不断演进的文档，可帮助用户有效地导航各种问题。

1.2. 搜索红帽知识库

如果出现 Red Hat Quay 问题，您可以执行初始搜索来确定红帽知识库中是否已存在解决方案。

先决条件

- 您有红帽客户门户网站帐户。

流程

1. 登录到 红帽客户门户网站。
2. 在主红帽客户门户网站搜索字段中，输入与问题相关的关键字和字符串，包括：
   - Red Hat Quay 组件(**如数据库**)
   - 相关步骤（比如 **安装**）
   - 警告、错误消息和其他与输出与特定的问题相关
3. 点 **Search**。
4. 选择 **Red Hat Quay** 产品过滤器。
5. 在内容类型过滤中选择 **Knowledgebase**。

1.3. 提交支持问题单

先决条件

- 您有红帽客户门户网站帐户。
- 您有红帽标准订阅或高级订阅。

步骤

1. 登录红帽客户门户 并选择 **打开支持问题单**。
2. 选择 **Troubleshoot** 选项卡。
3. 对于 **Summary**，请输入一个简洁但描述性的问题概述，以及有关所经历的症状的详细信息，以及您的预期。
4. 查看推荐的红帽知识库解决方案列表，它们可能会与您要报告的问题相关。如果建议的文章没有解决这个问题，请继续下一步。
5. 对于 **产品**，请选择 **Red Hat Quay**。
6. 选择您使用的 Red Hat Quay 版本。
7. 点 **Continue**。
8. 可选。拖放、粘贴或浏览以上传文件。这可以是从 Red Hat Quay 部署收集的调试日志。
9. 点 **Get support** to file your ticket。

第 2 章 在调试模式下运行 RED HAT QUAY

红帽建议在提交支持问题单时收集您的调试信息。在调试模式下运行 Red Hat Quay 会提供详细日志记录，以帮助管理员查找有关各种问题的更多信息。启用调试模式可加快流程来重现错误，并验证诸如地域复制部署、Operator 部署、独立 Red Hat Quay 部署、对象存储问题等的解决方案。另外，它可以帮助红帽支持执行根本原因分析。

2.1. 在调试模式下运行独立 RED HAT QUAY 部署

在调试模式下运行 Red Hat Quay 会提供详细日志记录，以帮助管理员查找有关各种问题的更多信息。启用调试模式可加快进程重现错误并验证解决方案。

使用以下步骤以调试模式运行独立部署 Red Hat Quay。

步骤

1. 输入以下命令在调试模式下运行独立 Red Hat Quay 部署：

```
$ podman run -p 443:8443 -p 80:8080 -e DEBUGLOG=true  -v /config:/conf/stack -v
/storage:/datastorage -d {productrepo}/{quayimage}:{productminv}
```

2. 要查看 debug 日志，请输入以下命令：

```
$ podman logs quay
```

2.2. 以 DEBUG 模式运行 RED HAT QUAY OPERATOR

使用以下步骤以 debug 模式运行 Red Hat Quay Operator。

步骤

1. 输入以下命令编辑 **QuayRegistry** 自定义资源定义：

```
$ oc edit quayregistry <quay_registry_name> -n <quay_namespace>
```

2. 更新 **QuayRegistry** 以添加以下参数：

```
spec:
  - kind: quay
    managed: true
    overrides:
      env:
      - name: DEBUGLOG
        value: "true"
```

3. 在 Red Hat Quay Operator 启用了调试后，尝试从 registry 中拉取镜像。如果它仍然缓慢，将所有 **Quay** pod 的 dogs 转储到文件中，并检查文件以了解更多信息。

第 3 章 RED HAT QUAY 的日志信息

使用 获取日志信息可用于管理、监控和故障排除容器或 pod 中运行的应用。获取日志信息的一些原因有价值包括：

- **调试和故障排除** ：日志可以深入了解应用程序内发生的情况，使开发人员和系统管理员能够识别和解决问题。通过分析日志消息，可以识别应用程序执行过程中可能出现的错误、异常、警告或意外行为。
- **性能监控** ：监控日志有助于跟踪应用程序及其组件的性能。监控响应时间、请求率和资源利用率等指标有助于优化和扩展应用程序以满足需求。
- **安全分析** ：日志对于审核和检测潜在的安全漏洞可能至关重要。通过分析日志，可以识别可疑活动、未经授权的访问尝试或任何异常行为，有助于检测和响应安全威胁。
- **跟踪用户行为** ：在某些情况下，日志可用于跟踪用户活动和行为。对于处理敏感数据的应用程序来说，这尤其重要，其中跟踪用户操作对审计和合规目的很有用。
- **容量规划** ：日志数据可用于了解资源利用率模式，这有助于容量规划。通过分析日志，可以识别峰值使用周期，预期资源需求，并相应地优化基础架构。
- **错误分析** ：发生错误时，日志可以提供导致错误所发生情况的宝贵上下文。这有助于了解问题的根本原因并促进调试过程。
- **Deployment**: Logging 验证部署过程中的日志记录有助于验证应用程序是否正确启动，以及所有组件是否按预期工作。
- **持续集成/持续部署(CI/CD)** ：在 CI/CD 管道中，日志记录对于捕获构建和部署状态至关重要，使团队能够监控每个阶段的成功或失败。

3.1. 获取 RED HAT QUAY 的日志信息

可以为所有类型的 Red Hat Quay 部署获取日志信息，包括跨地域复制部署、独立部署和 Operator 部署。也可以为已镜像的存储库获取日志信息。它可以帮助您对身份验证和授权问题进行故障排除，以及对象存储问题。获取必要的日志信息后，您可以搜索红帽知识库 以获取解决方案，或使用红帽支持团队提交支持问题单。

使用以下步骤获取 Red Hat Quay 部署的日志。

步骤

- 如果您在 OpenShift Container Platform 上使用 Red Hat Quay Operator，请输入以下命令查看日志：

```
$ oc logs <quay_pod_name>
```

- 如果您位于独立的 Red Hat Quay 部署中，请输入以下命令：

```
$ podman logs <quay_container_name>
```

输出示例

```
...
gunicorn-web stdout | 2023-01-20 15:41:52,071 [205] [DEBUG] [app] Starting request:
```

```
urn:request:0d88de25-03b0-4cf9-b8bc-87f1ac099429 (/oauth2/azure/callback) {'X-Forwarded-For': '174.91.79.124'}
...
```

3.2. 检查详细日志

Red Hat Quay 没有详细的日志，但以下步骤可以获取数据库 pod 或容器的详细状态检查。

步骤

1. 输入以下命令检查详细的数据库日志。
   a. 如果您在 OpenShift Container Platform 上使用 Red Hat Quay Operator，请输入以下命令：

```
$ oc logs <quay_pod_name> --previous
```

```
$ oc logs <quay_pod_name> --previous -c <container_name>
```

```
$ oc cp <quay_pod_name>:/var/lib/pgsql/data/userdata/log/* /path/to/desired_directory_on_host
```

   b. 如果使用 Red Hat Quay 的独立部署，请输入以下命令：

```
$ podman logs <quay_container_name> --previous
```

```
$ podman logs <quay_container_name> --previous -c <container_name>
```

```
$ podman cp <quay_container_name>:/var/lib/pgsql/data/userdata/log/* /path/to/desired_directory_on_host
```

第 4 章 RED HAT QUAY 的配置信息

检查配置 YAML 可帮助识别和解决与 Red Hat Quay 配置相关的各种问题。检查配置 YAML 可帮助您解决以下问题：

- **不正确的配置参数** ： 如果数据库无法按预期工作或遇到性能问题，您的配置参数可能会出错。通过检查配置 YAML，管理员可以确保正确设置所有必需的参数，并与数据库的预期设置匹配。
- 资源限制 ：配置 YAML 可能会为数据库指定资源限值，如内存和 CPU 限值。**如果数据库在资源约束中运行或遇到其他服务争用，则调整这些限制可帮助优化资源分配并改进整体性能。**
- **连接问题 ：更正配置 YAML 中的网络设置可能会导致应用程序和数据库间的连接问题。确保正确的网络配置已发生，可以解决与连接和通信相关的问题。**
- **数据存储 和路径 ：配置 YAML 可能包含存储数据和日志的路径。如果路径配置错误或无法访问，则数据库在读取或写入数据时可能会遇到错误，从而导致操作问题。**
- **身份验证和安全 ：配置 YAML 可以包含身份验证设置，包括用户名、密码和访问控制。**验证这些设置对于维护数据库的安全性至关重要，并确保只有授权的用户有权访问。
- **插件和扩展设置 ：一些数据库支持扩展或插件来增强功能。如果这些插件被错误配置或没有正确加载，则可能会出现问题。检查配置 YAML 有助于识别插件设置中的任何问题。**
- **复制和高可用性 设置：在集群或复制的数据库设置中，配置 YAML 可以定义复制设置和 高可用性配置。不正确的设置可能会导致数据不一致和系统不稳定。**
- 备份和恢复选项 ：**配置 YAML 可能包含备份和恢复选项，指定如何执行数据备份，以及在失败时如何恢复数据。**验证这些设置可以确保数据安全性和成功恢复过程。

通过检查配置 YAML，Red Hat Quay 管理员可以先检测并解决这些问题，然后它们会对应用程序或服务造成大量中断，这取决于数据库。

4.1. 获取 RED HAT QUAY 的配置信息

可以为所有类型的 Red Hat Quay 部署获取配置信息，包括独立、Operator 和异地复制部署。获取配置信息可帮助您解决身份验证和授权、数据库、对象存储和存储库镜像的问题。获取必要的配置信息后，您可以更新 config.yaml 文件，搜索 红帽知识库 中的解决方案，或使用红帽支持团队提交支持问题单。

步骤

1. **要获取 Red Hat Quay Operator 部署的配置信息，您可以使用 oc exec、oc cp 或 oc rsync。**
   a. **要使用 oc exec 命令，请输入以下命令：**

   ```
   $ oc exec -it <quay_pod_name> -- cat /conf/stack/config.yaml
   ```

   此命令将 config.yaml 文件直接返回到终端。
   b. **要使用 oc copy 命令，请输入以下命令：**

   ```
   $ oc cp <quay_pod_name>:/conf/stack/config.yaml /tmp/config.yaml
   ```

   要在终端中显示此信息，请输入以下命令：

   ```
   $ cat /tmp/config.yaml
   ```

c. **要使用 oc rsync 命令，请输入以下命令：**

```
oc rsync <quay_pod_name>:/conf/stack/ /tmp/local_directory/
```

要在终端中显示此信息，请输入以下命令：

```
$ cat /tmp/local_directory/config.yaml
```

输出示例

```
DISTRIBUTED_STORAGE_CONFIG:
local_us:
- RHOCSStorage
- access_key: redacted
  bucket_name: lht-quay-datastore-68fff7b8-1b5e-46aa-8110-c4b7ead781f5
  hostname: s3.openshift-storage.svc.cluster.local
  is_secure: true
  port: 443
  secret_key: redacted
  storage_path: /datastorage/registry
DISTRIBUTED_STORAGE_DEFAULT_LOCATIONS:
- local_us
DISTRIBUTED_STORAGE_PREFERENCE:
- local_us
```

2. **要获取独立 Red Hat Quay 部署的配置信息，您可以使用 podman cp 或 podman exec。**

a. **要使用 podman copy 命令，请输入以下命令：**

```
$ podman cp <quay_container_id>:/conf/stack/config.yaml /tmp/local_directory/
```

要在终端中显示此信息，请输入以下命令：

```
$ cat /tmp/local_directory/config.yaml
```

b. **要使用 podman exec，请输入以下命令：**

```
$ podman exec -it <quay_container_id> cat /conf/stack/config.yaml
```

输出示例

```
BROWSER_API_CALLS_XHR_ONLY: false
ALLOWED_OCI_ARTIFACT_TYPES:
    application/vnd.oci.image.config.v1+json:
        - application/vnd.oci.image.layer.v1.tar+zstd
    application/vnd.sylabs.sif.config.v1+json:
        - application/vnd.sylabs.sif.layer.v1+tar
AUTHENTICATION_TYPE: Database
AVATAR_KIND: local
BUILDLOGS_REDIS:
    host: quay-server.example.com
    password: strongpassword
    port: 6379
```

```
DATABASE_SECRET_KEY: 05ee6382-24a6-43c0-b30f-849c8a0f7260
DB_CONNECTION_ARGS: {}
---
```

4.2. 获取数据库配置信息

您可以按照以下流程获取数据库的配置信息。

警告

与 PostgreSQL 数据库交互可能存在破坏性。强烈建议您在 Red Hat Quay 支持专家下执行以下步骤。

步骤

- 如果您在 OpenShift Container Platform 上使用 Red Hat Quay Operator，请输入以下命令：

```
$ oc exec -it <database_pod> -- cat /var/lib/pgsql/data/userdata/postgresql.conf
```

- 如果使用 Red Hat Quay 的独立部署，请输入以下命令：

```
$ podman exec -it <database_container> cat
/var/lib/pgsql/data/userdata/postgresql.conf
```

第 5 章 在 RED HAT QUAY 部署上执行健康检查

健康检查机制旨在评估系统、服务或组件的健康和功能。健康检查有助于确保一切正常工作，并可用于在潜在问题成为严重问题之前识别潜在问题。通过监控系统的健康状况，Red Hat Quay 管理员可以针对地域复制部署、Operator 部署、独立 Red Hat Quay 部署、对象存储问题等方面解决异常或潜在的故障。执行健康检查有助于降低遇到故障排除场景的可能性。

通过提供有关系统当前状态的宝贵信息，健康检查机制可以在诊断问题方面扮演角色。通过将健康检查结果与预期的基准测试或预定义的阈值进行比较，可以更快地识别 deviations 或 anomalies。

5.1. RED HAT QUAY 健康检查端点

重要

此处包含的任何外部网站的链接仅为方便用户而提供。红帽没有审阅链接的内容，并不对其内容负责。包含到外部网站的任何链接并不意味着红帽认可该网站或其实体、产品或服务。您同意红帽对因您使用（或依赖）外部网站或内容而导致的任何损失或费用不承担任何责任。

Red Hat Quay 有几个健康检查端点。下表显示了健康检查、描述、端点和示例输出。

表 5.1. 健康检查端点

健康检查	描述	端点	输出示例
实例	实例 端点获取特定 Red Hat Quay 实例的完整状态。返回带有以下键值对 的字典 ： auth,database,disk_space,registry_gunicorn,service_key, 和 web_gunicorn。返回指示 200 的健康检查响应的数字，这表示实例处于健康状态，或者 503，这表示您的部署有问题。	https://{quay-ip-endpoint}/health/instance 或 https://{quay-ip-endpoint}/health	{"data":{"services":{"auth":true,"database":true,"disk_space":true,"registry_gunicorn":true,"service_key":true,"web_gunicorn":true}},"status_code":200}
endtoend	端到端端点对 Red Hat Quay 实例的所有服务进行检查。使用以下的键值对返回 字典 ： auth、数据库、redis、存储。返回指示 200 的健康检查响应的数字，这表示实例处于健康状态，或者 503，这表示您的部署有问题。	https://{quay-ip-endpoint}/health/endtoend	{"data":{"services":{"auth":true,"database":true,"redis":true,"storage":true}},"status_code":200}
warning	警告 端点对警告进行检查。为以下内容返回一个带有键值对的字典：disk_space_warning。返回指示 200 的健康检查响应的数字，这表示实例处于健康状态，或者 503，这表示您的部署有问题。	https://{quay-ip-endpoint}/health/warning	{"data":{"services":{"disk_space_warning":true}},"status_code":503}

5.2. 导航到 RED HAT QUAY 健康检查端点

使用以下步骤导航到 实例 端点。对于 端到端 和警告 端点，这个过程可以重复。

步骤

1. 在 Web 浏览器中，导航到 https://{quay-ip-endpoint}/health/instance。
2. 您使用健康实例页面，它会返回类似如下的信息：

```
{"data":{"services":
{"auth":true,"database":true,"disk_space":true,"registry_gunicorn":true,"service_key"
:true,"web_gunicorn":true}},"status_code":200}
```

对于 Red Hat Quay，"status_code": 200 表示实例是健康的。相反，如果您收到 "status_code": 503，则部署有问题。

第 6 章 RED HAT QUAY 组件故障排除

本文档侧重于对 Red Hat Quay 中特定组件的故障排除，为解决可能的问题提供针对性的指导。此资源专为系统管理员、操作员和开发人员设计，旨在帮助诊断和故障排除与各个 Red Hat Quay 组件相关的问题。

除了以下流程外，Red Hat Quay 组件还可以通过在 debug 模式下运行 Red Hat Quay、获取日志信息、获取配置信息以及对端点执行健康检查，从而进行故障排除。

通过按照以下流程，您可以排除常见组件问题。之后，您可以在 红帽知识库 中搜索解决方案，或使用红帽支持团队提交支持问题单。

6.1. 对 RED HAT QUAY 数据库进行故障排除

用于 Red Hat Quay 的 PostgreSQL 数据库存储与容器镜像及其管理有关的各种信息。PostgreSQL 数据库存储的一些关键信息包括：

- 镜像元数据.数据库存储与容器镜像关联的元数据，如镜像名称、版本、创建时间戳以及拥有镜像的用户或机构。通过这些信息，可以在 registry 中轻松识别和组织容器镜像。
- 镜像标签.Red Hat Quay 允许用户为容器镜像分配标签，从而方便的标签和版本控制。PostgreSQL 数据库维护镜像标签及其对应镜像清单之间的映射，允许用户根据提供的标签检索容器镜像的特定版本。
- 镜像层.容器镜像由多个层组成，这些层存储为单独的对象。数据库记录了有关这些层的信息，包括其顺序、校验和和大小。此数据对于有效存储和检索容器镜像至关重要。
- 用户和组织数据.Red Hat Quay 支持用户和组织管理，允许用户验证和管理对容器镜像的访问。PostgreSQL 数据库存储用户和组织信息，包括用户名、电子邮件地址、身份验证令牌和访问权限。
- 存储库信息.Red Hat Quay 将容器镜像组织到存储库中，充当对相关镜像进行分组的逻辑单元。数据库维护存储库数据，包括名称、描述、可见性设置和访问控制信息，让用户能够有效地管理和共享其存储库。
- 事件日志.Red Hat Quay 跟踪与镜像管理和存储库操作相关的各种事件和活动。这些事件日志（包括镜像拉取、拉取、删除和存储库修改）存储在 PostgreSQL 数据库中，提供审计跟踪，并允许管理员监控和分析系统活动。

本节中的内容涵盖了以下步骤：

- 检查部署类型 ：确定数据库是否作为容器部署到虚拟机上，还是在 OpenShift Container Platform 上作为 pod 部署。
- 检查容器或 pod 状态 ： 根据部署类型，使用特定命令验证数据库 pod 或容器的状态。
- 检查数据库容器或 pod 日志 ：访问并检查数据库 pod 或容器日志，包括用于不同部署类型的命令。
- 检查 Red Hat Quay 和数据库 pod之间的连接：使用相关命令检查 Red Hat Quay和数据库 pod 之间的连接。
- 检查数据库配置 ：根据部署类型检查不同级别的数据库配置(OpenShift Container Platform 或 PostgreSQL 级别)。
- 检查资源分配 ：监控 Red Hat Quay 部署的资源分配，包括磁盘使用情况和其他资源使用情况。

- 与 Red Hat Quay 数据库交互 ：了解如何与 PostgreSQL 数据库交互，包括访问和查询数据库的命令。

6.1.1. Red Hat Quay 数据库问题故障排除

使用以下步骤对 PostgreSQL 数据库进行故障排除。

6.1.1.1. 与 Red Hat Quay 数据库交互

使用以下步骤与 PostgreSQL 数据库交互。

警告

与 PostgreSQL 数据库交互可能存在破坏性。强烈建议您在 Red Hat Quay 支持专家下执行以下步骤。

注意

与 PostgreSQL 数据库交互也可用于对授权和身份验证问题进行故障排除。

步骤

1. 执行到 Red Hat Quay 数据库。
   a. 输入以下命令在 OpenShift Container Platform 上的 Red Hat Quay 数据库 pod 中执行：

   ```
   $ oc exec -it <quay_database_pod> -- psql
   ```

   b. 输入以下命令在独立部署上执行到 Red Hat Quay 数据库：

   ```
   $ sudo podman exec -it <quay_container_name> /bin/bash
   ```

2. 输入 PostgreSQL shell。

警告

与 PostgreSQL 数据库交互可能存在破坏性。强烈建议您在 Red Hat Quay 支持专家下执行以下步骤。

   a. 如果使用 Red Hat Quay Operator，请输入以下命令进入 PostgreSQL shell：

   ```
   $ oc rsh <quay_pod_name> psql -U your_username -d your_database_name
   ```

   b. 如果您位于独立的 Red Hat Quay 部署中，请输入以下命令进入 PostgreSQL shell：

```
bash-4.4$ psql -U your_username -d your_database_name
```

6.1.1.2. crashloopbackoff 状态故障排除

使用以下步骤 troueblshoot **crashloopbackoff 状态。**

步骤

1. **如果您的容器或 pod 处于 crashloopbackoff 状态，您可以输入以下命令。**
   a. **输入以下命令缩减 Red Hat Quay Operator：**

   ```
   $ oc scale deployment/quay-operator.v3.8.z  --replicas=0
   ```

   输出示例

   ```
   deployment.apps/quay-operator.v3.8.z scaled
   ```

   b. **输入以下命令缩减 Red Hat Quay 数据库：**

   ```
   $ oc scale deployment/<quay_database> --replicas=0
   ```

   输出示例

   ```
   deployment.apps/<quay_database> scaled
   ```

   c. **输入以下命令编辑 Red Hat Quay 数据库：**

   警告

   与 PostgreSQL 数据库交互可能存在破坏性。强烈建议您在 Red Hat Quay 支持专家下执行以下步骤。

   ```
   $ oc edit deployment <quay_database>
   ```

   ```
   ...
    template:
      metadata:
        creationTimestamp: null
        labels:
          quay-component: <quay_database>
          quay-operator/quayregistry: quay-operator.v3.8.z
      spec:
        containers:
        - env:
          - name: POSTGRESQL_USER
            value: postgres
   ```

```
      - name: POSTGRESQL_DATABASE
        value: postgres
      - name: POSTGRESQL_PASSWORD
        value: postgres
      - name: POSTGRESQL_ADMIN_PASSWORD
        value: postgres
      - name: POSTGRESQL_MAX_CONNECTIONS
        value: "1000"
      image: registry.redhat.io/rhel8/postgresql-
10@sha256:a52ad402458ec8ef3f275972c6ebed05ad64398f884404b9bb8e3010c5c95
291
      imagePullPolicy: IfNotPresent
      name: postgres
      command: ["/bin/bash", "-c", "sleep 86400"] 1
...
```

1 将此行添加到同一缩进中。

输出示例

```
deployment.apps/<quay_database> edited
```

d. 在您的 < quay_database> 中执行以下命令 ：

```
$ oc exec -it <quay_database> -- cat /var/lib/pgsql/data/userdata/postgresql/logs/*
/path/to/desired_directory_on_host
```

6.1.1.3. 检查 Red Hat Quay 和数据库 pod 之间的连接

使用以下步骤检查 Red Hat Quay 和数据库 pod 之间的连接

步骤

1. 检查 Red Hat Quay 和数据库 pod 之间的连接。

   a. 如果您在 OpenShift Container Platform 上使用 Red Hat Quay Operator，请输入以下命令：

   ```
   $ oc exec -it _quay_pod_name_ -- curl -v telnet://<database_pod_name>:5432
   ```

   b. 如果使用 Red Hat Quay 的独立部署，请输入以下命令：

   ```
   $ podman exec -it <quay_container_name >curl -v
   telnet://<database_container_name>:5432
   ```

6.1.1.4. 检查资源分配

使用以下步骤检查资源分配。

步骤

1. 获取正在运行的容器列表。

2. 监控 Red Hat Quay 部署的磁盘用量。

   a. 如果您在 OpenShift Container Platform 上使用 Red Hat Quay Operator，请输入以下命令：

   ```
   $ oc exec -it <quay_database_pod_name> -- df -ah
   ```

   b. 如果使用 Red Hat Quay 的独立部署，请输入以下命令：

   ```
   $ podman exec -it <quay_database_conatiner_name> df -ah
   ```

3. 监控其他资源使用量。

   a. 输入以下命令检查 Red Hat Quay Operator 部署中的资源分配：

   ```
   $ oc adm top pods
   ```

   b. 输入以下命令检查 Red Hat Quay 的独立部署中特定 pod 的状态：

   ```
   $ podman pod stats <pod_name>
   ```

   c. 输入以下命令检查 Red Hat Quay 独立部署中特定容器的状态：

   ```
   $ podman stats <container_name>
   ```

   返回以下信息：

   - CPU %.容器自上次测量以来的 CPU 用量百分比。这个值代表了容器的可用 CPU 资源的共享。
   - MEM 使用/限制.容器的当前内存用量及其内存限制。这些值以 current_usage / memory_limit 格式显示。例如，300.4MiB / 7.795GiB 表示容器当前使用 300.4 MB 内存，限制为 7.795GB。
   - MEM %.容器与内存限制相关的内存用量百分比。
   - 网络 I/O.容器的网络 I/O (input/output)统计信息。它显示容器通过网络传输和接收的数据量。这些值以格式显示： transport _bytes / received_bytes。
   - 块 I/O。容器的块 I/O (input/output)统计信息。它代表了从中读取并写入容器使用的块设备（如磁盘）的数据量。这些值以 read_bytes / written_bytes 格式显示。

6.1.2. 在 Red Hat Quay 独立部署中重置超级用户密码

使用以下步骤重置超级用户的密码。

先决条件

- 您已创建了 Red Hat Quay 超级用户。
- 已安装 Python 3.9。
- 您已安装了 Python 的 pip 软件包管理器。
- 您已安装了用于 pip 的 bcrypt 软件包。

步骤

1. 输入以下命令，使用 Python 3.9 中的 bcrypt 软件包生成安全散列密码：

```
$ python3.9 -c 'import bcrypt; print(bcrypt.hashpw(b"newpass1234",
bcrypt.gensalt(12)).decode("utf-8"))'
```

输出示例

```
$2b$12$T8pkgtOoys3G5ut7FV1She6vXlYgU.6TeoGmbbAVQtN8X8ch4knKm
```

2. 输入以下命令显示 Red Hat Quay 容器 registry 的容器 ID：

```
$ sudo podman ps -a
```

输出示例

```
CONTAINER ID  IMAGE                                COMMAND        CREATED         STATUS
PORTS                   NAMES
70560beda7aa  registry.redhat.io/rhel8/redis-5:1        run-redis       2 hours ago     Up 2
hours ago     0.0.0.0:6379->6379/tcp                redis
8012f4491d10  registry.redhat.io/quay/quay-rhel8:v3.8.2  registry        3 minutes ago
Up 8 seconds ago   0.0.0.0:80->8080/tcp, 0.0.0.0:443->8443/tcp  quay
8b35b493ac05  registry.redhat.io/rhel8/postgresql-10:1   run-postgresql  39 seconds
ago  Up 39 seconds ago  0.0.0.0:5432->5432/tcp                postgresql-quay
```

3. 输入以下命令为 postgresql 容器镜像执行交互式 shell：

```
$ sudo podman exec -it 8b35b493ac05 /bin/bash
```

4. 重新输入 quay PostgreSQL 数据库服务器，指定数据库、用户名和主机地址：

```
bash-4.4$ psql -d quay -U quayuser -h 192.168.1.28 -W
```

5. 更新超级用户 admin 的 password_hash，其丢失了其密码：

```
quay=> UPDATE public.user SET password_hash =
'$2b$12$T8pkgtOoys3G5ut7FV1She6vXlYgU.6TeoGmbbAVQtN8X8ch4knKm' where
username = 'quayadmin';
```

输出示例

```
UPDATE 1
```

6. 输入以下命令以确保 password_hash 已更新：

```
quay=> select * from public.user;
```

输出示例

```
id | uuid | username | password_hash | email | verified | stripe_id | organization | robot |
```

```
invoice_email | invalid_login_attempts | last_invalid_login |removed_tag_expiration_s |
enabled | invoice_email_address | company | family_name | given_name | location |
maximum_queued_builds_count | creation_date | last_accessed
----+----------+----------+----------+----------+----------+----------+----------
----------+----------+---
----------+----------+----------+----------+----------+----------+----------
--+----------+----------+------
---+----------+---------+----------+----------+---------+----------+---
----------+----------
1 | 73f04ef6-19ba-41d3-b14d-f2f1eed94a4a | quayadmin |
$2b$12$T8pkgtOoys3G5ut7FV1She6vXIYgU.6TeoGmbbAVQtN8X8ch4knKm |
quayadmin@example.com | t | | f | f | f | 0 | 2023-02-23 07:54:39.116485 | 1209600 | t | | | |
| | | 2023-02-23 07:54:39.116492
```

7. 使用新密码登录到 Red Hat Quay 部署：

```
$ sudo podman login -u quayadmin -p newpass1234 http://quay-server.example.com --
tls-verify=false
```

输出示例

```
Login Succeeded!
```

其他资源

如需更多信息，请参阅为 Quay 重置超级用户密码。

6.1.3. 重置 Red Hat Quay Operator 上的超级用户密码

先决条件

- 您已创建了 Red Hat Quay 超级用户。
- 已安装 Python 3.9。
- 您已安装了 Python 的 pip 软件包管理器。
- 您已安装了用于 pip 的 bcrypt 软件包。

步骤

1. 登录到您的 Red Hat Quay 部署。
2. 在 OpenShift Container Platform UI 中，进入到 Workloads → Secrets。
3. 选择 Red Hat Quay 部署的命名空间，例如 Project quay。
4. 查找和存储 PostgreSQL 数据库凭据。
5. 输入以下命令，使用 Python 3.9 中的 bcrypt 软件包生成安全散列密码：

```
$ python3.9 -c 'import bcrypt; print(bcrypt.hashpw(b"newpass1234",
bcrypt.gensalt(12)).decode("utf-8"))'
```

输出示例

```
$2b$12$zoilcTG6XQeAoVuDulZH0..UpvQEZcKh3V6puksQJaUQupHgJ4.4y
```

6. 在 CLI 中，登录到数据库，例如：

```
$ oc rsh quayuser-quay-quay-database-669c8998f-v9qsl
```

7. 输入以下命令打开到 quay PostgreSQL 数据库服务器的连接，指定数据库、用户名和主机地址：

```
sh-4.4$ psql -U quayuser-quay-quay-database -d quayuser-quay-quay-database -W
```

8. 输入以下命令连接到当前用户的默认数据库：

```
quay=> \c
```

9. 更新超级用户 admin 的 password_hash，其丢失了其密码：

```
quay=> UPDATE public.user SET password_hash =
'$2b$12$zoilcTG6XQeAoVuDulZH0..UpvQEZcKh3V6puksQJaUQupHgJ4.4y' where
username = 'quayadmin';
```

10. 输入以下命令以确保 password_hash 已更新：

```
quay=> select * from public.user;
```

输出示例

```
id | uuid | username | password_hash | email | verified | stripe_id | organization | robot |
invoice_email | invalid_login_attempts | last_invalid_login |removed_tag_expiration_s |
enabled | invoice_email_address | company | family_name | given_name | location |
maximum_queued_builds_count | creation_date | last_accessed
----+--------------------------------------+-----------+--------------------------------------------------------
-------+----------------------+---
-------+-----------+-------------+-------+--------------+----------------------+-------------------------
--+--------------------------+------
---+----------------------+---------+-------------+-----------+----------+----------------------------+---
-------------------------+-----------
1 | 73f04ef6-19ba-41d3-b14d-f2f1eed94a4a | quayadmin |
$2b$12$zoilcTG6XQeAoVuDulZH0..UpvQEZcKh3V6puksQJaUQupHgJ4.4y |
quayadmin@example.com | t | | f | f | f | 0 | 2023-02-23 07:54:39.116485 | 1209600 | t | | | |
| | | 2023-02-23 07:54:39.116492
```

11. 导航到 OpenShift Container Platform 上的 Red Hat Quay UI，并使用新凭证登录。

6.2. RED HAT QUAY 身份验证故障排除

认证和授权对于安全访问 Red Hat Quay 至关重要。它们一起保护敏感的容器镜像、验证用户身份、强制访问控制、促进审计和责任，以及实现与外部身份提供程序的无缝集成。通过优先考虑身份验证，组织可以增强其容器 registry 环境的整体安全性和完整性。

Red Hat Quay 支持以下验证方法：

- 用户名和密码。用户可以通过提供其用户名和密码来进行身份验证，这些用户名和密码会根据 Red Hat Quay 中配置的用户数据库进行验证。这种传统方法要求用户输入其凭据来获取访问权限。
- OAuth.Red Hat Quay 支持 OAuth 身份验证，允许用户使用 Google、GitHub 或 Keycloak 等第三方服务的凭证进行身份验证。OAuth 启用无缝和联合登录体验，无需单独创建帐户并简化用户管理。
- OIDC.OpenID Connect 启用单点登录(SSO)功能，并与企业身份提供程序集成。通过 OpenID Connect，用户可以使用其现有的机构凭证进行身份验证，为各种系统和应用程序提供统一身份验证体验。
- 基于令牌的身份验证.用户可以获取为 Red Hat Quay 中特定资源授予访问权限的唯一令牌。令牌可以通过各种方法获取，如 OAuth 或在 Red Hat Quay 用户界面中生成 API 令牌。基于令牌的身份验证通常用于自动或编程对 registry 的访问。
- 外部身份提供程序.Red Hat Quay 可以与外部身份提供程序（如 LDAP 或 AzureAD）集成，以进行身份验证。这种集成允许组织使用其现有的身份管理基础架构，实现集中用户身份验证并减少对单独的用户数据库的需求。

6.2.1. 对特定用户的 Red Hat Quay 身份验证和授权问题进行故障排除

使用以下步骤对特定用户的身份验证和授权问题进行故障排除。

步骤

1. 执行到 Red Hat Quay pod 或容器。如需更多信息，请参阅"与 Red Hat Quay 数据库交互"。
2. 输入以下命令以显示用于外部身份验证的所有用户：

```
quay=# select * from federatedlogin;
```

输出示例

```
id | user_id | service_id |              service_ident              |              metadata_json
----+---------+------------+-----------------------------------------+-----------------------------------
------
1 |       1 |          3 | testuser0                               | {}
2 |       1 |          8 | PK7Zpg2Yu2AnfUKG15hKNXqOXirqUog6G-oE7OgzSWc |
{"service_username": "live.com#testuser0"}
3 |       2 |          3 | testuser1                               | {}
4 |       2 |          4 | 110875797246250333431                   | {"service_username":
"testuser1"}
5 |       3 |          3 | testuser2                               | {}
6 |       3 |          1 | 26310880                                | {"service_username": "testuser2"}
(6 rows)
```

3. 验证用户是否已插入到 user 表中：

```
quay=# select username, email from "user";
```

输出示例

```
username |       email
----------+---------------------
testuser0 | testuser0@outlook.com
testuser1  | testuser1@gmail.com
testuser2  | testuser2@redhat.com
(3 rows)
```

6.3. RED HAT QUAY 对象存储故障排除

对象存储是一种数据存储架构，将数据作为称为 对象的 离散单元进行管理。与将数据组织为分层目录和文件的传统文件系统不同，对象存储将数据视为具有唯一标识符的独立实体。每个对象都包含数据本身，以及描述对象并实现高效检索的元数据。

Red Hat Quay 使用对象存储作为存储和管理容器镜像的底层存储机制。它将容器镜像存储为各个对象。每个容器镜像都被视为对象，其唯一标识符和相关元数据。

6.3.1. Red Hat Quay 对象存储问题故障排除

使用以下选项对 Red Hat Quay 对象存储问题进行故障排除。

步骤

- 输入以下命令查看使用什么对象存储：

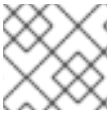
```
$ oc get quayregistry quay-registry-name -o yaml
```

- 通过检查 经过测试的集成 页面，确保 Red Hat Quay 正式支持您使用的对象存储。
- 启用调试模式。如需更多信息，请参阅"在调试模式中运行 Red Hat Quay"。
- 检查 config.yaml 文件中的对象存储配置。确保它准确，并与对象存储供应商提供的设置匹配。您可以检查访问凭证、端点 URL、存储桶和容器名称等信息，以及其他相关的配置参数。
- 确保 Red Hat Quay 具有与对象存储端点的网络连接。检查网络配置，以确保没有限制阻止 Red Hat Quay 和对象存储端点之间的通信。
- 如果在 config.yaml 文件中启用了 FEATURE_STORAGE_PROXY，请检查其下载 URL 是否可以访问。这可以在 Red Hat Quay 调试日志中找到。例如：

```
$ curl -vvv
"https://QUAY_HOSTNAME/_storage_proxy/dhaWZKRjlyO.....Kuhc=/https/quay.hostname.com/quay-
test/datastorage/registry/sha256/0e/0e1d17a1687fa270ba4f52a85c0f0e7958e13d3ded5123c3851a8031a9e55681?
AWSAccessKeyId=xxxx&Signature=xxxxxx4%3D&Expires=1676066703"
```

- 尝试访问 Red Hat Quay 之外的对象存储服务，以确定问题是否特定于您的部署或底层对象存储。您可以使用对象存储供应商提供的命令行工具（如 aws、gsutil、或 s3cmd ）来执行列出存储桶、容器或上传和下载对象等基本操作。这可帮助您隔离问题。

6.4. GEO-REPLICATION



注意

目前，IBM Power 不支持 geo-replication 功能。

地理复制允许多个地理分布的 Red Hat Quay 部署从客户端或用户的角度来看作为单个 registry 工作。它显著提高了在全局分布式 Red Hat Quay 设置中的推送和拉取性能。镜像数据在后台异步复制，并带有透明故障转移，并为客户端重定向。

在独立和 Operator 部署中支持部署带有 geo-replication 的 Red Hat Quay。

6.4.1. Red Hat Quay 的 geo-replication 故障排除

使用以下部分对 Red Hat Quay 的 geo-replication 进行故障排除。

6.4.1.1. 检查后端存储桶中的数据复制

使用以下步骤确保数据在所有后端存储桶中正确复制。

先决条件

- 已安装 aws CLI。

步骤

1. 输入以下命令以确保数据在所有后端存储桶中复制：

```
$ aws --profile quay_prod_s3 --endpoint=http://10.0.x.x:port s3 ls ocp-quay --recursive
--human-readable --summarize
```

输出示例

```
Total Objects: 17996
Total Size: 514.4 GiB
```

6.4.1.2. 检查后端存储的状态

使用以下资源检查后端存储的状态。

- Amazon Web Service Storage (AWS)在 AWS Service Health Dashboard上检查 AWS S3 服务健康状况。使用 aws CLI 或 SDK 列出已知存储桶中的对象，以验证您对 S3 的访问。
- Google Cloud Storage (GCS)检查 Google Cloud Status Dashboard以查看 GCS 服务的状态。使用 Google Cloud SDK 或 GCS 客户端库列出已知存储桶中的对象，以验证您对 GCS 的访问。
- NooBaa.检查 NooBaa 管理控制台或管理界面，以了解任何健康或状态指示符。确保 NooBaa 服务和相关组件正在运行并可访问。使用 NooBaa CLI 或 SDK 列出已知存储桶中的对象，以验证对 NooBaa 的访问。
- Red Hat OpenShift Data Foundation检查 OpenShift Container Platform 控制台或管理界面，以查看 Red Hat OpenShift Data Foundation 组件的状态。验证 Red Hat OpenShift Data Foundation S3 接口和服务的可用性。确保 Red Hat OpenShift Data Foundation 服务正在运行并可访问。使用适当的 S3 兼容 SDK 或 CLI 列出已知存储桶中的对象，以验证对 Red Hat OpenShift Data Foundation S3 的访问。

- Ceph.检查 Ceph 服务的状态，包括 Ceph 监视器、OSD 和 RGW。验证 Ceph 集群是否正常运行。使用适当的 Ceph 对象存储 API 或 CLI 列出已知 bucket 中的对象，以验证对 Ceph 对象存储的访问。
- Azure Blob 存储.检查 Azure Status Dashboard，以查看 Azure Blob Storage 服务的健康状态。使用 Azure CLI 或 Azure SDK 列出容器或对象，验证您对 Azure Blob Storage 的访问。
- OpenStack Swift.检查 OpenStack Status 页面，以验证 OpenStack Swift 服务的状态。确保 Swift 服务（如代理服务器、容器服务器、对象服务器）正在运行并可访问。使用适当的 Swift CLI 或 SDK 列出容器或对象，以验证您对 Swift 的访问。

检查后端存储的状态后，请确保所有 Red Hat Quay 实例都可以访问所有 s3 存储后端。

6.5. 存储库镜像

Red Hat Quay 存储库镜像可让您将外部容器 registry 或另一个本地 registry 的镜像 mirror 到 Red Hat Quay 集群。使用存储库镜像，您可以根据存储库名称和标签将镜像同步到 Red Hat Quay。

在启用了存储库镜像的 Red Hat Quay 集群中，您可以执行以下操作：

- 从外部 registry 中选择一个仓库(mirror)
- 添加用于访问外部 registry 的凭证
- 识别要同步的特定容器镜像存储库名称和标签
- 设置同步存储库的间隔
- 检查同步的当前状态

要使用镜像功能，您需要执行以下操作：

- 在 Red Hat Quay 配置文件中启用存储库镜像
- 运行存储库镜像 worker
- 创建镜像的存储库

所有存储库镜像配置都可以使用配置工具 UI 或 Red Hat Quay API 执行。

6.5.1. 存储库镜像故障排除

使用以下部分对 Red Hat Quay 的存储库镜像进行故障排除。

6.5.1.1. 验证身份验证和权限

确保用于镜像的身份验证凭证对源和目标 Red Hat Quay 实例都有必要的权限和访问权限。

在 Red Hat Quay UI 中，检查以下设置：

- 访问控制设置。确保执行镜像操作的用户或服务帐户具有所需的特权。
- Red Hat Quay registry 中的机器人帐户的权限。

6.6. CLAIR 安全扫描程序

6.6.1. Clair 故障排除问题

使用以下步骤对 Clair 进行故障排除。

6.6.1.1. 验证镜像兼容性

如果使用 Clair，请确保 Clair 支持您要扫描的镜像。Clair 具有特定要求，不支持所有镜像格式或配置。

如需更多信息，请参阅 Clair 漏洞数据库。

6.6.1.2. allowlisting Clair updaters

如果您在代理配置后面使用 Clair，则必须列出代理或防火墙配置中的 updaters。如需有关 updater URL 的更多信息，请参阅 Clair updater URL。

6.6.1.3. 更新 Clair 扫描程序及其依赖项

确保您使用最新版本的 Clair 安全扫描程序。过时的版本可能缺少对较新镜像格式的支持，或者可能存在一个已知问题。

使用以下步骤检查 Clair 的版本。

注意

检查 Clair 日志也可用于检查 Clair 日志中是否有 updaters 微服务的错误。默认情况下，Clair 每 30 分钟更新漏洞数据库。

步骤

1. 检查您的 Clair 版本。
   a. **如果您在 Red Hat Quay Operator 上运行 Clair，请输入以下命令：**

   ```
   $ oc logs clair-pod
   ```

   b. **如果您正在运行 Red Hat Quay 的独立部署并使用 Clair 容器，请输入以下命令：**

   ```
   $ podman logs clair-container
   ```

   输出示例

   ```
   "level":"info",
   "component":"main",
   "version":"v4.5.1",
   ```

6.6.1.4. 为 Clair 启用调试模式

默认情况下，Clair 的调试模式被启用。您可以通过更新 Clair config.yaml 文件为 Clair 启用调试模式。

使用以下步骤为 Clair 启用调试模式。

步骤

1. 为 Clair 启用调试模式

   a. 如果您在 Red Hat Quay Operator 上运行 Clair，请输入以下命令：

   ```
   $ oc exec -it clair-pod-name -- cat /clair/config.yaml
   ```

   b. 如果您正在运行 Red Hat Quay 的独立部署并使用 Clair 容器，请输入以下命令：

   ```
   $ podman exec -it clair-container-name cat /clair/config.yaml
   ```

2. 更新 Clair config.yaml 文件以启用调试：

```
http_listen_addr: :8081
introspection_addr: :8088
log_level: debug
```

6.6.1.5. 检查 Clair 配置

检查 Clair config.yaml 文件，以确保没有可能导致问题的错误配置或不一致。如需更多信息，请参阅 Clair 配置概述。

6.6.1.6. 检查镜像元数据

在某些情况下，您可能会收到 Unsupported 信息。这可能表示扫描程序无法从镜像中提取必要的元数据。检查镜像元数据是否已正确格式化并可访问。

其他资源

如需更多信息，请参阅 Clair 故障排除。