Red Hat OpenShift GitOps 1.12

访问控制和用户管理

为用户和命名空间配置用户身份验证和访问控制

Last Updated: 2024-05-15

Red Hat OpenShift GitOps 1.12 访问控制和用户管理

为用户和命名空间配置用户身份验证和访问控制

法律通告

摘要

本文档提供了有关更改和管理用户级别访问和资源请求的说明。它还讨论了如何配置基于角色的访问控制和单点登录身份验证供应商，以管理集群中的多个用户、权限、Argo CD 资源和实例。

目录

第 1 章 配置 ARGO CD RBAC

默认情况下，如果您使用 Red Hat SSO (RH SSO)登录到 Argo CD，则是一个只读用户。您可以更改并管理用户级别访问权限。

1.1. 配置用户级别访问权限

要管理和修改用户级别访问权限，请在 Argo CD 自定义资源(CR)中配置基于角色的访问控制(RBAC)部分。

流程

1. 编辑 **argocd** CR：

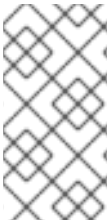
```
$ oc edit argocd [argocd-instance-name] -n [namespace]
```

输出

```
metadata
...
...
  rbac:
    policy: 'g, rbacsystem:cluster-admins, role:admin'
    scopes: '[groups]'
```

2. 将策略配置添加到 **rbac** 部分，并添加要应用到 **用户的名称** 和所需角色：

```
metadata
...
...
rbac:
   policy: g, <name>, role:<admin>
   scopes: '[groups]'
```



注意

目前，RHSSO 无法读取 Red Hat OpenShift GitOps 用户的组信息。因此，在用户级别配置 RBAC。

1.2. 修改 RHSSO 资源请求/限制

默认情况下，RHSSO 容器创建有资源请求和限值。您可以更改并管理资源请求。

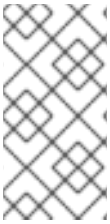
资源	Requests	Limits
CPU	500	1000m

资源	Requests	Limits
内存	512 Mi	1024 Mi

流程

- **修改默认资源要求，修补 Argo CD 自定义资源(CR)：**

```
$ oc -n openshift-gitops patch argocd openshift-gitops --type='json' -p='[{"op": "add", "path": "/spec/sso", "value": {"provider": "keycloak", "resources": {"requests": {"cpu": "512m", "memory": "512Mi"}, "limits": {"cpu": "1024m", "memory": "1024Mi"}} }}]'
```

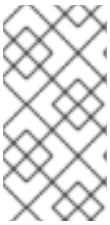


注意

Red Hat OpenShift GitOps 创建的 RHSSO 仅保留操作器所进行的更改。如果 RHSSO 重新启动，则 RHSSO 中的 Admin 创建的额外配置也会被删除。

第 2 章 使用 DEX 为 ARGO CD 配置 SSO

安装 Red Hat OpenShift GitOps Operator 后，Argo CD 会自动创建一个具有 admin 权限的用户。要管理多个用户，集群管理员可以使用 Argo CD 来配置 Single Sign-On(SSO)。



注意

ArgoCD CR 中的 spec.dex 参数不再支持 Red Hat OpenShift GitOps v1.10.0。考虑改用 .spec.sso 参数。

2.1. 配置以启用 DEX OPENSHIFT OAUTH CONNECTOR

对于 Operator 创建的所有 Argo CD 实例，默认安装 Dex。您可以通过设置 .spec.sso 参数，将 Red Hat OpenShift GitOps 配置为使用 Dex 作为 SSO 身份验证提供程序。

Dex 通过检查平台提供的 OAuth 服务器，使用 OpenShift Container Platform 中定义的用户和组。

流程

- 要启用 Dex，在 Operator 的 YAML 资源中将 .spec.sso.provider 参数设置为 dex ：

```
# ...
spec:
  sso:
    provider: dex
    dex:
      openShiftOAuth: true 1
# ...
```

1. openShiftOAuth 属性触发 Operator，以便在值设为 true 时自动配置内置 OpenShift Container Platform OAuth 服务器。

2.1.1. 将用户映射到特定的角色

如果有直接 ClusterRoleBinding 角色，Argo CD 无法将用户映射到特定角色。您可以通过 OpenShift，手动更改 SSO 上的 role:admin 角色。

流程

流程

1. 创建名为 cluster-admins 的组。

```
$ oc adm groups new cluster-admins
```

2. 将用户添加到组。

```
$ oc adm groups add-users cluster-admins USER
```

3. 将 cluster-admin ClusterRole 应用到组：

```
$ oc adm policy add-cluster-role-to-group cluster-admin cluster-admins
```

2.2. 通过替换 .SPEC.SSO 禁用 DEX

- 要禁用 dex，可以从 Argo CD 自定义资源中删除 spec.sso 元素，或指定不同的 SSO 供应商。

第 3 章 使用 KEYCLOAK 为 ARGO CD 配置 SSO

安装 Red Hat OpenShift GitOps Operator 后，Argo CD 会自动创建一个具有 admin 权限的用户。要管理多个用户，集群管理员可以使用 Argo CD 来配置 Single Sign-On(SSO)。

3.1. 先决条件

- 在集群中安装了 Red Hat SSO。
- Red Hat OpenShift GitOps Operator 已安装在 OpenShift Container Platform 集群中。
- 在集群中安装了 Argo CD。

3.2. 在 KEYCLOAK 中配置新客户端

对于 Operator 创建的所有 Argo CD 实例，默认安装 Dex。但是，您可以删除 Dex 配置并添加 Keycloak，以使用 OpenShift 凭证登录到 Argo CD。Keycloak 作为 Argo CD 和 OpenShift 之间的身份代理。

流程

要配置 Keycloak，请按照以下步骤执行：

1. 通过从 Argo CD 自定义资源 (CR) 中删除 .spec.sso.dex 参数来删除 Dex 配置，并保存 CR：

```
dex:
  openShiftOAuth: true
  resources:
    limits:
      cpu:
      memory:
    requests:
      cpu:
      memory:
```

2. 在 Argo CD CR 中将 provider 参数的值设置为 keycloak。

3. 通过执行以下步骤配置 Keycloak：

- 对于安全连接，设置 rootCA 参数的值，如下例所示：

```
apiVersion: argoproj.io/v1beta1
kind: ArgoCD
metadata:
  name: example-argocd
  labels:
    example: basic
spec:
  sso:
    provider: keycloak
    keycloak:
      rootCA: "<PEM-encoded-root-certificate>" 1
  server:
    route:
      enabled: true
```

1 用于验证 Keycloak 的 TLS 证书的自定义证书。

Operator 会协调 .spec.sso.keycloak.rootCA 参数中的更改，并使用 argocd-cm 配置映射中的 PEM 编码 root 证书更新 oidc.config 参数。

- 对于不安全连接，将 rootCA 参数的值留空，并使用 oidc.tls.insecure.skip.verify 参数，如下所示：

```
apiVersion: argoproj.io/v1beta1
kind: ArgoCD
metadata:
  name: example-argocd
  labels:
    example: basic
spec:
  extraConfig:
    oidc.tls.insecure.skip.verify: "true"
  sso:
    provider: keycloak
    keycloak:
      rootCA: ""
```

注意

Keycloak 实例需要 2-3 分钟来安装和运行。

3.3. 登录到 KEYCLOAK

登录到 Keycloak 控制台以管理身份或角色，并定义分配给不同角色的权限。

先决条件

- 删除 Dex 的默认配置。
- Argo CD CR 必须配置为使用 Keycloak SSO 供应商。

流程

1. 获取用于登录的 Keycloak 路由 URL：

```
$ oc -n argocd get route keycloak

NAME       HOST/PORT                                          PATH   SERVICES   PORT
TERMINATION   WILDCARD
keycloak   keycloak-default.apps.ci-ln-******.origin-ci-int-aws.dev.**.com        keycloak
<all>    reencrypt     None
```

2. 获取将用户名和密码存储为环境变量的 Keycloak pod 名称：

```
$ oc -n argocd get pods

NAME                 READY   STATUS        RESTARTS   AGE
keycloak-1-2sjcl         1/1    Running         0       45m
```

   a. 获取 Keycloak 用户名：

```
$ oc -n argocd exec keycloak-1-2sjcl -- "env" | grep SSO_ADMIN_USERNAME

SSO_ADMIN_USERNAME=Cqid54lh
```

b. 获取 Keycloak 密码：

```
$ oc -n argocd exec keycloak-1-2sjcl -- "env" | grep SSO_ADMIN_PASSWORD

SSO_ADMIN_PASSWORD=GVXxHifH
```

3. 在登录页面上，点 LOG IN VIA KEYCLOAK。

注意

您只能在 Keycloak 实例就绪后看到 LOGIN VIA KEYCLOAK 选项。

4. 点 Login with OpenShift。

注意

不支持使用 kubeadmin 登录。

5. 输入要登录的 OpenShift 凭据。

6. 可选：默认情况下，登录到 Argo CD 的任何用户都具有只读访问权限。您可以通过更新 argocd-rbac-cm 配置映射来管理用户级别访问权限：

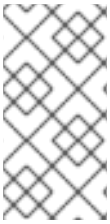
```
policy.csv:
<name>, <email>, role:admin
```

3.4. 卸载 KEYCLOAK

您可以通过从 Argo CD 自定义资源(CR)文件中删除 SSO 字段来删除 Keycloak 资源及其相关配置。删除 SSO 字段后，文件中的值类似如下：

```
apiVersion: argoproj.io/v1beta1
kind: ArgoCD
metadata:
  name: example-argocd
  labels:
    example: basic
```

```
spec:
  server:
    route:
     enabled: true
```



注意

使用此方法创建的 Keycloak 应用程序当前不是持久性。在服务器重启时，在 Argo CD Keycloak 域中创建的其他配置会被删除。