



OpenShift Container Platform 4.12

Console web

Démarrer avec la console web dans OpenShift Container Platform

OpenShift Container Platform 4.12 Console web

Démarrer avec la console web dans OpenShift Container Platform

Notice légale

Copyright © 2023 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

Résumé

Ce document fournit des instructions pour accéder à la console web d'OpenShift Container Platform et la personnaliser.

Table des matières

CHAPITRE 1. VUE D'ENSEMBLE DE LA CONSOLE WEB	4
1.1. À PROPOS DE LA PERSPECTIVE DE L'ADMINISTRATEUR DANS LA CONSOLE WEB	4
1.2. À PROPOS DE LA PERSPECTIVE DU DÉVELOPPEUR DANS LA CONSOLE WEB	5
1.3. ACCÉDER AUX PERSPECTIVES	6
CHAPITRE 2. ACCÉDER À LA CONSOLE WEB	7
2.1. CONDITIONS PRÉALABLES	7
2.2. COMPRENDRE ET ACCÉDER À LA CONSOLE WEB	7
CHAPITRE 3. UTILISER LE TABLEAU DE BORD D'OPENSIFT CONTAINER PLATFORM POUR OBTENIR DES INFORMATIONS SUR LES CLUSTERS	8
3.1. A PROPOS DE LA PAGE DES TABLEAUX DE BORD D'OPENSIFT CONTAINER PLATFORM	8
CHAPITRE 4. AJOUT DES PRÉFÉRENCES DE L'UTILISATEUR	10
4.1. DÉFINITION DES PRÉFÉRENCES DE L'UTILISATEUR	10
CHAPITRE 5. CONFIGURER LA CONSOLE WEB DANS OPENSIFT CONTAINER PLATFORM	11
5.1. CONDITIONS PRÉALABLES	11
5.2. CONFIGURATION DE LA CONSOLE WEB	11
CHAPITRE 6. PERSONNALISER LA CONSOLE WEB DANS OPENSIFT CONTAINER PLATFORM	12
6.1. AJOUT D'UN LOGO ET D'UN NOM DE PRODUIT PERSONNALISÉS	12
6.2. CRÉER DES LIENS PERSONNALISÉS DANS LA CONSOLE WEB	13
6.3. PERSONNALISATION DES ITINÉRAIRES DE LA CONSOLE	14
6.4. PERSONNALISER LA PAGE DE CONNEXION	16
6.5. DÉFINITION D'UN MODÈLE POUR UN LIEN DE JOURNAL EXTERNE	17
6.6. CRÉATION DE BANNIÈRES DE NOTIFICATION PERSONNALISÉES	18
6.7. PERSONNALISATION DES TÉLÉCHARGEMENTS CLI	18
6.8. AJOUTER DES EXEMPLES YAML AUX RESSOURCES KUBERNETES	19
6.9. PERSONNALISER LES PERSPECTIVES DES UTILISATEURS	20
6.10. PERSONNALISATION DU CATALOGUE ET DU SOUS-CATALOGUE DU DÉVELOPPEUR	23
CHAPITRE 7. PLUGINS DYNAMIQUES	27
7.1. APERÇU DES PLUGINS DYNAMIQUES	27
7.2. DÉMARRER AVEC DES PLUGINS DYNAMIQUES	28
7.3. DÉPLOYER VOTRE PLUGIN SUR UN CLUSTER	29
7.4. EXEMPLE DE PLUGIN DYNAMIQUE	31
7.5. RÉFÉRENCE DU PLUGIN DYNAMIQUE	33
CHAPITRE 8. A PROPOS DU TERMINAL WEB DANS LA CONSOLE WEB	98
8.1. INSTALLATION DU TERMINAL WEB	98
8.2. UTILISATION DU TERMINAL WEB	99
8.3. DÉSINSTALLATION DU TERMINAL WEB	100
CHAPITRE 9. DÉSACTIVER LA CONSOLE WEB DANS OPENSIFT CONTAINER PLATFORM	104
9.1. CONDITIONS PRÉALABLES	104
9.2. DÉSACTIVATION DE LA CONSOLE WEB	104
CHAPITRE 10. CRÉER DES TUTORIELS DE DÉMARRAGE RAPIDE DANS LA CONSOLE WEB	105
10.1. COMPRENDRE LES DÉMARRAGES RAPIDES	105
10.2. DÉMARRAGE RAPIDE DU FLUX DE TRAVAIL DE L'UTILISATEUR	105
10.3. COMPOSANTS DE DÉMARRAGE RAPIDE	106
10.4. CONTRIBUER AUX DÉMARRAGES RAPIDES	106
10.5. LIGNES DIRECTRICES SUR LE CONTENU POUR UN DÉMARRAGE RAPIDE	118

CHAPITRE 1. VUE D'ENSEMBLE DE LA CONSOLE WEB

La console web de Red Hat OpenShift Container Platform fournit une interface utilisateur graphique permettant de visualiser les données de votre projet et d'effectuer des tâches d'administration, de gestion et de dépannage. La console web s'exécute en tant que pods sur les nœuds du plan de contrôle dans le projet openshift-console. Elle est gérée par un pod **console-operator**. Les perspectives **Administrator** et **Developer** sont toutes deux prises en charge.

Les perspectives **Administrator** et **Developer** vous permettent de créer des tutoriels de démarrage rapide pour OpenShift Container Platform. Un démarrage rapide est un tutoriel guidé avec des tâches d'utilisateur et est utile pour s'orienter avec une application, un opérateur ou une autre offre de produit.

1.1. À PROPOS DE LA PERSPECTIVE DE L'ADMINISTRATEUR DANS LA CONSOLE WEB

La perspective **Administrator** vous permet de visualiser l'inventaire de la grappe, sa capacité, des informations générales et spécifiques sur son utilisation, ainsi que le flux d'événements importants, ce qui vous aide à simplifier les tâches de planification et de dépannage. Les administrateurs de projets et les administrateurs de clusters peuvent visualiser la perspective **Administrator**.

Les administrateurs de cluster peuvent également ouvrir une instance de terminal de ligne de commande intégrée avec l'opérateur de terminal web dans OpenShift Container Platform 4.7 et plus.



NOTE

La perspective de la console web affichée par défaut dépend du rôle de l'utilisateur. La perspective **Administrator** est affichée par défaut si l'utilisateur est reconnu comme administrateur.

La perspective **Administrator** fournit des flux de travail spécifiques aux cas d'utilisation de l'administrateur, tels que la possibilité de.. :

- Gérer la charge de travail, le stockage, la mise en réseau et les paramètres des clusters.
- Installer et gérer les opérateurs à l'aide du Hub des opérateurs.
- Ajouter des fournisseurs d'identité qui permettent aux utilisateurs de se connecter et de gérer l'accès des utilisateurs par le biais de rôles et de liaisons de rôles.
- Visualiser et gérer une variété de paramètres avancés tels que les mises à jour de clusters, les mises à jour partielles de clusters, les opérateurs de clusters, les définitions de ressources personnalisées (CRD), les liaisons de rôles et les quotas de ressources.
- Accéder et gérer les fonctions de surveillance telles que les mesures, les alertes et les tableaux de bord de surveillance.
- Visualiser et gérer la journalisation, les métriques et les informations sur l'état de la grappe.
- Interagir visuellement avec les applications, les composants et les services associés à la perspective **Administrator** dans OpenShift Container Platform.

Ressources supplémentaires

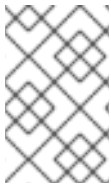
Voir [À propos du terminal web dans la console web](#) pour plus d'informations sur l'opérateur du terminal web.

1.2. À PROPOS DE LA PERSPECTIVE DU DÉVELOPPEUR DANS LA CONSOLE WEB

La perspective **Developer** offre plusieurs moyens intégrés de déployer des applications, des services et des bases de données. Dans la perspective **Developer**, vous pouvez

- Visualisation en temps réel du déroulement et de la recreation des déroulements sur le composant.
- Visualisez l'état de l'application, l'utilisation des ressources, le flux d'événements du projet et la consommation de quotas.
- Partagez votre projet avec d'autres.
- Dépannez les problèmes de vos applications en exécutant des requêtes Prometheus Query Language (PromQL) sur votre projet et en examinant les métriques visualisées sur un graphique. Les métriques fournissent des informations sur l'état d'un cluster et sur les charges de travail définies par l'utilisateur que vous surveillez.

Les administrateurs de cluster peuvent également ouvrir une instance de terminal de ligne de commande intégrée dans la console web dans OpenShift Container Platform 4.7 et plus.



NOTE

La perspective de la console web affichée par défaut dépend du rôle de l'utilisateur. La perspective **Developer** est affichée par défaut si l'utilisateur est reconnu comme développeur.

La perspective **Developer** fournit des flux de travail spécifiques aux cas d'utilisation des développeurs, tels que la possibilité de.. :

- Créer et déployer des applications sur OpenShift Container Platform en important des bases de code, des images et des fichiers de conteneurs existants.
- Interagir visuellement avec les applications, les composants et les services qui leur sont associés au sein d'un projet et surveiller leur déploiement et leur état de construction.
- Regrouper des composants au sein d'une application et relier les composants au sein d'une même application et d'une application à l'autre.
- Intégrer des capacités sans serveur (aperçu technologique).
- Créez des espaces de travail pour éditer le code de votre application à l'aide d'Eclipse Che.

Vous pouvez utiliser la vue **Topology** pour afficher les applications, les composants et les charges de travail de votre projet. Si vous n'avez pas de charges de travail dans le projet, la vue **Topology** affichera des liens pour les créer ou les importer. Vous pouvez également utiliser la vue **Quick Search** pour importer directement des composants.

Ressources complémentaires

Voir [Visualisation de la composition des applications à l'aide de la](#) vue **Topology** pour plus d'informations sur l'utilisation de la vue **Topology** dans la perspective **Developer**.

1.3. ACCÉDER AUX PERSPECTIVES

Vous pouvez accéder aux perspectives **Administrator** et **Developer** à partir de la console web de la manière suivante :

Conditions préalables

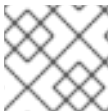
Pour accéder à une perspective, assurez-vous d'être connecté à la console web. Votre perspective par défaut est automatiquement déterminée par les droits des utilisateurs. La perspective **Administrator** est sélectionnée pour les utilisateurs ayant accès à tous les projets, tandis que la perspective **Developer** est sélectionnée pour les utilisateurs ayant un accès limité à leurs propres projets

Ressources complémentaires

Voir [Ajouter des préférences utilisateur](#) pour plus d'informations sur la modification des perspectives.

Procédure

1. Utilisez le sélecteur de perspective pour passer à la perspective **Administrator** ou **Developer**.
2. Sélectionnez un projet existant dans la liste déroulante **Project**. Vous pouvez également créer un nouveau projet à partir de cette liste déroulante.



NOTE

Vous ne pouvez utiliser le commutateur de perspective qu'à l'adresse **cluster-admin**.

Ressources supplémentaires

- [En savoir plus sur l'administrateur de cluster](#)
- [Aperçu de la perspective **Administrator**](#)
- [Créer et déployer des applications sur OpenShift Container Platform en utilisant la perspective **Developer**](#)
- [Visualiser les applications de votre projet, vérifier leur état de déploiement et interagir avec elles dans la vue **Topology**](#)
- [Visualisation des informations sur les clusters](#)
- [Configuration de la console web](#)
- [Personnaliser la console web](#)
- [Lancement d'une instance de terminal de ligne de commande intégrée dans la console web](#)
- [Créer des tutoriels de démarrage rapide](#)
- [Désactivation de la console web](#)

CHAPITRE 2. ACCÉDER À LA CONSOLE WEB

La console web d'OpenShift Container Platform est une interface utilisateur accessible depuis un navigateur web. Les développeurs peuvent utiliser la console web pour visualiser, parcourir et gérer le contenu des projets.

2.1. CONDITIONS PRÉALABLES

- JavaScript doit être activé pour utiliser la console web. Pour une expérience optimale, utilisez un navigateur web qui prend en charge les [WebSockets](#).
- Consultez la page [OpenShift Container Platform 4.x Tested Integrations](#) avant de créer l'infrastructure de support de votre cluster.

2.2. COMPRENDRE ET ACCÉDER À LA CONSOLE WEB

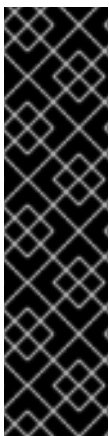
La console web fonctionne comme un pod sur le maître. Les ressources statiques nécessaires à l'exécution de la console web sont servies par le pod. Après l'installation réussie d'OpenShift Container Platform à l'aide de **openshift-install create cluster**, trouvez l'URL de la console web et les identifiants de connexion pour votre cluster installé dans la sortie CLI du programme d'installation. Par exemple :

Exemple de sortie

```
INFO Install complete!
INFO Run 'export KUBECONFIG=<your working directory>/auth/kubeconfig' to manage the cluster
with 'oc', the OpenShift CLI.
INFO The cluster is ready when 'oc login -u kubeadmin -p <provided>' succeeds (wait a few minutes).
INFO Access the OpenShift web-console here: https://console-openshift-
console.apps.demo1.openshift4-beta-abc.com
INFO Login to the console with user: kubeadmin, password: <provided>
```

Utilisez ces informations pour vous connecter et accéder à la console web.

Pour les clusters existants que vous n'avez pas installés, vous pouvez utiliser **oc whoami --show-console** pour voir l'URL de la console web.



IMPORTANT

Le paramètre **dir** spécifie le répertoire **assets**, qui contient les fichiers manifestes, l'image ISO et le répertoire **auth**. Le répertoire **auth** contient les fichiers **kubeadmin-password** et **kubeconfig**. En tant qu'utilisateur de **kubeadmin**, vous pouvez utiliser le fichier **kubeconfig** pour accéder au cluster avec le paramètre suivant : **export KUBECONFIG=<install_directory>/auth/kubeconfig**. Le paramètre **kubeconfig** est spécifique à l'image ISO générée. Par conséquent, si le paramètre **kubeconfig** est défini et que la commande **oc** échoue, il est possible que le système n'ait pas démarré avec l'image ISO générée. Pour effectuer un débogage, pendant le processus de démarrage, vous pouvez vous connecter à la console en tant qu'utilisateur **core** en utilisant le contenu du fichier **kubeadmin-password**.

Ressources supplémentaires

- [Activation des jeux de fonctionnalités à l'aide de la console web](#)

CHAPITRE 3. UTILISER LE TABLEAU DE BORD D'OPENSIFT CONTAINER PLATFORM POUR OBTENIR DES INFORMATIONS SUR LES CLUSTERS

Accédez au tableau de bord d'OpenShift Container Platform, qui capture des informations de haut niveau sur le cluster, en naviguant vers **Home** → **Dashboards** → **Overview** depuis la console web d'OpenShift Container Platform.

Le tableau de bord d'OpenShift Container Platform fournit diverses informations sur les clusters, capturées dans des cartes de tableau de bord individuelles.

3.1. A PROPOS DE LA PAGE DES TABLEAUX DE BORD D'OPENSIFT CONTAINER PLATFORM

Le tableau de bord d'OpenShift Container Platform se compose des cartes suivantes :

- **Details** fournit un bref aperçu des détails de la grappe d'information. Les statuts comprennent **ok**, **error**, **warning**, **in progress**, et **unknown**. Les ressources peuvent ajouter des noms d'état personnalisés.
 - ID du groupe
 - Fournisseur
 - Version
- **Cluster Inventory** détaille le nombre de ressources et les statuts associés. Il est utile lorsqu'une intervention est nécessaire pour résoudre des problèmes, y compris des informations sur :
 - Nombre de nœuds
 - Nombre de gousses
 - Demandes de volumes de stockage persistants
 - Hôtes en métal nu dans le cluster, listés en fonction de leur état (disponible uniquement dans l'environnement **metal3**).
- **Cluster Capacity** aident les administrateurs à comprendre quand des ressources supplémentaires sont nécessaires dans le cluster. Les graphiques contiennent un anneau intérieur qui affiche la consommation actuelle, tandis qu'un anneau extérieur affiche les seuils configurés pour la ressource, y compris des informations sur :
 - Temps CPU
 - Allocation de mémoire
 - Stockage consommé
 - Ressources réseau consommées
- **Cluster Utilization** montre la capacité de diverses ressources sur une période donnée, afin d'aider les administrateurs à comprendre l'ampleur et la fréquence d'une forte consommation de ressources.

- **Events** répertorie les messages liés à l'activité récente du cluster, comme la création d'un pod ou la migration d'une machine virtuelle vers un autre hôte.
- **Top Consumers** aide les administrateurs à comprendre comment les ressources de la grappe sont consommées. Cliquez sur une ressource pour accéder à une page détaillée répertoriant les pods et les nœuds qui consomment la plus grande quantité de la ressource de cluster spécifiée (CPU, mémoire ou stockage).

CHAPITRE 4. AJOUT DES PRÉFÉRENCES DE L'UTILISATEUR

Vous pouvez modifier les préférences par défaut de votre profil pour répondre à vos besoins. Vous pouvez définir votre projet par défaut, la vue de la topologie (graphique ou liste), le support d'édition (formulaire ou YAML), les préférences linguistiques et le type de ressource.

Les modifications apportées aux préférences de l'utilisateur sont automatiquement enregistrées.

4.1. DÉFINITION DES PRÉFÉRENCES DE L'UTILISATEUR

Vous pouvez définir les préférences utilisateur par défaut pour votre cluster.

Procédure

1. Connectez-vous à la console web d'OpenShift Container Platform en utilisant vos identifiants de connexion.
2. Utilisez l'en-tête pour accéder aux préférences de l'utilisateur sous le profil de l'utilisateur.
3. Dans la section **General**:
 - a. Dans le champ **Perspective**, vous pouvez définir la perspective par défaut dans laquelle vous souhaitez être connecté. Vous pouvez sélectionner la perspective **Administrator** ou **Developer** selon vos besoins. Si aucune perspective n'est sélectionnée, vous êtes connecté à la perspective que vous avez visitée en dernier.
 - b. Dans le champ **Project**, sélectionnez un projet dans lequel vous souhaitez travailler. La console affichera par défaut le projet à chaque fois que vous vous connecterez.
 - c. Dans le champ **Topology**, vous pouvez définir l'affichage de la topologie par défaut en vue graphique ou en vue liste. Si cette option n'est pas sélectionnée, la console reprend par défaut la dernière vue utilisée.
 - d. Dans le champ **Create/Edit resource method**, vous pouvez définir une préférence pour la création ou la modification d'une ressource. Si les options formulaire et YAML sont toutes deux disponibles, la console prend par défaut votre sélection.
4. Dans la section **Language**, sélectionnez **Default browser language** pour utiliser les paramètres linguistiques par défaut du navigateur. Sinon, sélectionnez la langue que vous souhaitez utiliser pour la console.
5. Dans la section **Applications**:
 - a. Vous pouvez voir le type de ressource par défaut **Resource type**. Par exemple, si OpenShift Serverless Operator est installé, le type de ressource par défaut est **Serverless Deployment**. Dans le cas contraire, le type de ressource par défaut est **Deployment**.
 - b. Vous pouvez sélectionner un autre type de ressource comme type de ressource par défaut dans le champ **Resource Type**.

CHAPITRE 5. CONFIGURER LA CONSOLE WEB DANS OPENSIFT CONTAINER PLATFORM

Vous pouvez modifier la console web de OpenShift Container Platform pour définir une URL de redirection de déconnexion ou désactiver la console.

5.1. CONDITIONS PRÉALABLES

- Déployer un cluster OpenShift Container Platform.

5.2. CONFIGURATION DE LA CONSOLE WEB

Vous pouvez configurer les paramètres de la console web en modifiant la ressource **console.config.openshift.io**.

- Modifier la ressource **console.config.openshift.io**:

```
$ oc edit console.config.openshift.io cluster
```

L'exemple suivant présente un exemple de définition de ressource pour la console :

```
apiVersion: config.openshift.io/v1
kind: Console
metadata:
  name: cluster
spec:
  authentication:
    logoutRedirect: "" 1
status:
  consoleURL: "" 2
```

1 Spécifiez l'URL de la page à charger lorsqu'un utilisateur se déconnecte de la console web. Si vous ne spécifiez pas de valeur, l'utilisateur retourne à la page de connexion de la console web. La spécification d'une URL **logoutRedirect** permet à vos utilisateurs d'effectuer une déconnexion unique (SLO) via le fournisseur d'identité pour détruire leur session d'authentification unique.

2 L'URL de la console web. Pour mettre à jour cette URL avec une valeur personnalisée, voir **Customizing the web console URL**

CHAPITRE 6. PERSONNALISER LA CONSOLE WEB DANS OPENSIFT CONTAINER PLATFORM

Vous pouvez personnaliser la console web d'OpenShift Container Platform pour définir un logo, un nom de produit, des liens, des notifications et des téléchargements de ligne de commande personnalisés. Ceci est particulièrement utile si vous avez besoin d'adapter la console web pour répondre aux exigences spécifiques d'une entreprise ou d'un gouvernement.

6.1. AJOUT D'UN LOGO ET D'UN NOM DE PRODUIT PERSONNALISÉS

Vous pouvez créer une marque personnalisée en ajoutant un logo ou un nom de produit personnalisé. Vous pouvez définir les deux ou l'un sans l'autre, car ces paramètres sont indépendants l'un de l'autre.

Conditions préalables

- Vous devez avoir des privilèges d'administrateur.
- Créez un fichier du logo que vous souhaitez utiliser. Le logo peut être un fichier dans n'importe quel format d'image courant, y compris GIF, JPG, PNG ou SVG, et est limité à une adresse **max-height** de **60px**.

Procédure

1. Importez votre fichier de logo dans une carte de configuration dans l'espace de noms **openshift-config**:

```
$ oc create configmap console-custom-logo --from-file /path/to/console-custom-logo.png -n openshift-config
```

ASTUCE

Vous pouvez également appliquer le YAML suivant pour créer la carte de configuration :

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: console-custom-logo
  namespace: openshift-config
binaryData:
  console-custom-logo.png: <base64-encoded_logo> ... 1
```

- 1 Fournir un logo valide codé en base64.

2. Modifiez la configuration de l'opérateur de la console web pour inclure **customLogoFile** et **customProductName**:

```
$ oc edit consoles.operator.openshift.io cluster
```

```
apiVersion: operator.openshift.io/v1
kind: Console
metadata:
  name: cluster
```



```
spec:
  customization:
    customLogoFile:
      key: console-custom-logo.png
      name: console-custom-logo
    customProductName: My Console
```

Une fois la configuration de l'opérateur mise à jour, il synchronisera la carte de configuration du logo personnalisé dans l'espace de noms de la console, la montera sur le pod de la console et la redéploiera.

3. Vérifier le succès. En cas de problème, l'opérateur de cluster de la console signale un état **Degraded** et la configuration de l'opérateur de la console signale également un état **CustomLogoDegraded**, mais avec des raisons telles que **KeyOrFilenameInvalid** ou **NoImageProvided**.

Pour vérifier le site **clusteroperator**, exécutez le programme :

```
$ oc get clusteroperator console -o yaml
```

Pour vérifier la configuration de l'opérateur de console, exécutez :

```
$ oc get consoles.operator.openshift.io -o yaml
```

6.2. CRÉER DES LIENS PERSONNALISÉS DANS LA CONSOLE WEB

Conditions préalables

- Vous devez avoir des privilèges d'administrateur.

Procédure

1. À partir de **Administration** → **Custom Resource Definitions**, cliquez sur **ConsoleLink**.
2. Sélectionnez l'onglet **Instances**
3. Cliquez sur **Create Console Link** et modifiez le fichier :

```
apiVersion: console.openshift.io/v1
kind: ConsoleLink
metadata:
  name: example
spec:
  href: 'https://www.example.com'
  location: HelpMenu ❶
  text: Link ❶
```

- ❶ Les paramètres de localisation valides sont **HelpMenu**, **UserMenu**, **ApplicationMenu** et **NamespaceDashboard**.

Pour que le lien personnalisé apparaisse dans tous les espaces de noms, suivez cet exemple :

```
apiVersion: console.openshift.io/v1
```

```
kind: ConsoleLink
metadata:
  name: namespaced-dashboard-link-for-all-namespaces
spec:
  href: 'https://www.example.com'
  location: NamespaceDashboard
  text: This appears in all namespaces
```

Pour que le lien personnalisé n'apparaisse que dans certains espaces de noms, suivez cet exemple :

```
apiVersion: console.openshift.io/v1
kind: ConsoleLink
metadata:
  name: namespaced-dashboard-for-some-namespaces
spec:
  href: 'https://www.example.com'
  location: NamespaceDashboard
  # This text will appear in a box called "Launcher" under "namespace" or "project" in the web console
  text: Custom Link Text
  namespaceDashboard:
    namespaces:
      # for these specific namespaces
      - my-namespace
      - your-namespace
      - other-namespace
```

Pour que le lien personnalisé apparaisse dans le menu de l'application, suivez cet exemple :

```
apiVersion: console.openshift.io/v1
kind: ConsoleLink
metadata:
  name: application-menu-link-1
spec:
  href: 'https://www.example.com'
  location: ApplicationMenu
  text: Link 1
  applicationMenu:
    section: My New Section
    # image that is 24x24 in size
    imageURL: https://via.placeholder.com/24
```

4. Cliquez sur **Save** pour appliquer vos modifications.

6.3. PERSONNALISATION DES ITINÉRAIRES DE LA CONSOLE

Pour les itinéraires **console** et **downloads**, la fonctionnalité d'itinéraires personnalisés utilise l'API de configuration d'itinéraires **ingress** config. Si l'itinéraire personnalisé **console** est configuré à la fois dans **ingress** config et **console-operator** config, la nouvelle configuration de l'itinéraire personnalisé **ingress** config a la priorité. La configuration de l'itinéraire avec **console-operator** config est obsolète.

6.3.1. Personnalisation de l'itinéraire de la console

Vous pouvez personnaliser l'itinéraire de la console en définissant le nom d'hôte personnalisé et le certificat TLS dans le champ **spec.componentRoutes** de la configuration du cluster **Ingress**.

Conditions préalables

- Vous vous êtes connecté au cluster en tant qu'utilisateur disposant de privilèges administratifs.
- Vous avez créé un secret dans l'espace de noms **openshift-config** contenant le certificat et la clé TLS. Cela est nécessaire si le domaine du suffixe du nom d'hôte personnalisé ne correspond pas au suffixe du domaine du cluster. Le secret est facultatif si le suffixe correspond.

ASTUCE

Vous pouvez créer un secret TLS à l'aide de la commande **oc create secret tls**.

Procédure

1. Modifiez la configuration du cluster **Ingress**:

```
$ oc edit ingress.config.openshift.io cluster
```

2. Définir le nom d'hôte personnalisé et, éventuellement, le certificat et la clé de service :

```
apiVersion: config.openshift.io/v1
kind: Ingress
metadata:
  name: cluster
spec:
  componentRoutes:
    - name: console
      namespace: openshift-console
      hostname: <custom_hostname> 1
      servingCertKeyPairSecret:
        name: <secret_name> 2
```

1 Le nom d'hôte personnalisé.

2 Référence à un secret dans l'espace de noms **openshift-config** qui contient un certificat TLS (**tls.crt**) et une clé (**tls.key**). Cette référence est obligatoire si le domaine du suffixe du nom d'hôte personnalisé ne correspond pas au suffixe du domaine du cluster. Le secret est facultatif si le suffixe correspond.

3. Enregistrez le fichier pour appliquer les modifications.

6.3.2. Personnalisation de l'itinéraire de téléchargement

Vous pouvez personnaliser la route de téléchargement en définissant le nom d'hôte personnalisé et le certificat TLS dans le champ **spec.componentRoutes** de la configuration du cluster **Ingress**.

Conditions préalables

- Vous vous êtes connecté au cluster en tant qu'utilisateur disposant de privilèges administratifs.

- Vous avez créé un secret dans l'espace de noms **openshift-config** contenant le certificat et la clé TLS. Cela est nécessaire si le domaine du suffixe du nom d'hôte personnalisé ne correspond pas au suffixe du domaine du cluster. Le secret est facultatif si le suffixe correspond.

ASTUCE

Vous pouvez créer un secret TLS à l'aide de la commande **oc create secret tls**.

Procédure

1. Modifiez la configuration du cluster **Ingress**:

```
$ oc edit ingress.config.openshift.io cluster
```

2. Définir le nom d'hôte personnalisé et, éventuellement, le certificat et la clé de service :

```
apiVersion: config.openshift.io/v1
kind: Ingress
metadata:
  name: cluster
spec:
  componentRoutes:
    - name: downloads
  namespace: openshift-console
  hostname: <custom_hostname> 1
  servingCertKeyPairSecret:
    name: <secret_name> 2
```

1

Le nom d'hôte personnalisé.

2

Référence à un secret dans l'espace de noms **openshift-config** qui contient un certificat TLS (**tls.crt**) et une clé (**tls.key**). Cette référence est obligatoire si le domaine du suffixe du nom d'hôte personnalisé ne correspond pas au suffixe du domaine du cluster. Le secret est facultatif si le suffixe correspond.

3. Enregistrez le fichier pour appliquer les modifications.

6.4. PERSONNALISER LA PAGE DE CONNEXION

Créez des informations sur les conditions d'utilisation à l'aide de pages de connexion personnalisées. Les pages de connexion personnalisées peuvent également être utiles si vous utilisez un fournisseur de connexion tiers, tel que GitHub ou Google, pour montrer aux utilisateurs une page de marque à laquelle ils font confiance et qu'ils attendent avant d'être redirigés vers le fournisseur d'authentification. Vous pouvez également créer des pages d'erreur personnalisées pendant le processus d'authentification.



NOTE

La personnalisation du modèle d'erreur est limitée aux fournisseurs d'identité (IDP) qui utilisent des redirections, tels que l'en-tête de requête et les IDP basés sur OIDC. Elle n'a pas d'effet sur les IDP qui utilisent l'authentification directe par mot de passe, tels que LDAP et httpasswd.

Conditions préalables

- Vous devez avoir des privilèges d'administrateur.

Procédure

1. Exécutez les commandes suivantes pour créer des modèles que vous pouvez modifier :

```
$ oc adm create-login-template > login.html
```

```
oc adm create-provider-selection-template > providers.html
```

```
oc adm create-error-template > errors.html
```

2. Créer les secrets :

```
$ oc create secret generic login-template --from-file=login.html -n openshift-config
```

```
$ oc create secret generic providers-template --from-file=providers.html -n openshift-config
```

```
$ oc create secret generic error-template --from-file=errors.html -n openshift-config
```

3. Exécutez :

```
$ oc edit oauths cluster
```

4. Mettre à jour le cahier des charges :

```
spec:
  templates:
    error:
      name: error-template
    login:
      name: login-template
    providerSelection:
      name: providers-template
```

Lancez **oc explain oauths.spec.templates** pour comprendre les options.

6.5. DÉFINITION D'UN MODÈLE POUR UN LIEN DE JOURNAL EXTERNE

Si vous êtes connecté à un service qui vous aide à parcourir vos journaux, mais que vous devez générer des URL d'une manière particulière, vous pouvez définir un modèle pour votre lien.

Conditions préalables

- Vous devez avoir des privilèges d'administrateur.

Procédure

1. À partir de **Administration** → **Custom Resource Definitions**, cliquez sur **ConsoleExternalLogLink**.

2. Sélectionnez l'onglet **Instances**
3. Cliquez sur **Create Console External Log Link** et modifiez le fichier :

```
apiVersion: console.openshift.io/v1
kind: ConsoleExternalLogLink
metadata:
  name: example
spec:
  hrefTemplate: >-
    https://example.com/logs?
    resourceName=${resourceName}&containerName=${containerName}&resourceNamespace=${
    {resourceNamespace}&podLabels=${podLabels}
  text: Example Logs
```

6.6. CRÉATION DE BANNIÈRES DE NOTIFICATION PERSONNALISÉES

Conditions préalables

- Vous devez avoir des privilèges d'administrateur.

Procédure

1. À partir de **Administration** → **Custom Resource Definitions**, cliquez sur **ConsoleNotification**.
2. Sélectionnez l'onglet **Instances**
3. Cliquez sur **Create Console Notification** et modifiez le fichier :

```
apiVersion: console.openshift.io/v1
kind: ConsoleNotification
metadata:
  name: example
spec:
  text: This is an example notification message with an optional link.
  location: BannerTop 1
  link:
    href: 'https://www.example.com'
    text: Optional link text
  color: '#fff'
  backgroundColor: '#0088ce'
```

- 1** Les paramètres de localisation valides sont **BannerTop**, **BannerBottom**, et **BannerTopBottom**.

4. Cliquez sur **Create** pour appliquer vos modifications.

6.7. PERSONNALISATION DES TÉLÉCHARGEMENTS CLI

Vous pouvez configurer les liens de téléchargement de l'interface de programmation avec un texte de lien et des URL personnalisés, qui peuvent pointer directement vers les paquets de fichiers ou vers une page externe qui fournit les paquets.

Conditions préalables

- Vous devez avoir des privilèges d'administrateur.

Procédure

1. Naviguez jusqu'à **Administration** → **Custom Resource Definitions**
2. Sélectionnez **ConsoleCLIDownload** dans la liste des Custom Resource Definitions (CRD).
3. Cliquez sur l'onglet **YAML**, puis effectuez vos modifications :

```
apiVersion: console.openshift.io/v1
kind: ConsoleCLIDownload
metadata:
  name: example-cli-download-links-for-foo
spec:
  description: |
    This is an example of download links for foo
  displayName: example-foo
  links:
    - href: 'https://www.example.com/public/foo.tar'
      text: foo for linux
    - href: 'https://www.example.com/public/foo.mac.zip'
      text: foo for mac
    - href: 'https://www.example.com/public/foo.win.zip'
      text: foo for windows
```

4. Cliquez sur le bouton **Save**.

6.8. AJOUTER DES EXEMPLES YAML AUX RESSOURCES KUBERNETES

Vous pouvez ajouter dynamiquement et à tout moment des exemples YAML à n'importe quelle ressource Kubernetes.

Conditions préalables

- Vous devez disposer des privilèges d'administrateur de cluster.

Procédure

1. À partir de **Administration** → **Custom Resource Definitions**, cliquez sur **ConsoleYAMLSample**.
2. Cliquez sur **YAML** et modifiez le fichier :

```
apiVersion: console.openshift.io/v1
kind: ConsoleYAMLSample
metadata:
  name: example
spec:
  targetResource:
    apiVersion: batch/v1
    kind: Job
```

```

title: Example Job
description: An example Job YAML sample
yaml: |
  apiVersion: batch/v1
  kind: Job
  metadata:
    name: countdown
  spec:
    template:
      metadata:
        name: countdown
      spec:
        containers:
        - name: counter
          image: centos:7
          command:
            - "bin/bash"
            - "-c"
            - "for i in 9 8 7 6 5 4 3 2 1 ; do echo $i ; done"
        restartPolicy: Never

```

Utilisez **spec.snippet** pour indiquer que l'échantillon YAML n'est pas la définition complète de la ressource YAML, mais un fragment qui peut être inséré dans le document YAML existant au niveau du curseur de l'utilisateur.

3. Cliquez sur **Save**.

6.9. PERSONNALISER LES PERSPECTIVES DES UTILISATEURS

La console web d'OpenShift Container Platform fournit deux perspectives par défaut, **Administrator** et **Developer**. Vous pouvez avoir plus de perspectives disponibles en fonction des plugins de console installés. En tant qu'administrateur de cluster, vous pouvez afficher ou masquer une perspective pour tous les utilisateurs ou pour un rôle d'utilisateur spécifique. La personnalisation des perspectives permet de s'assurer que les utilisateurs ne peuvent voir que les perspectives applicables à leur rôle et à leurs tâches. Par exemple, vous pouvez masquer la perspective **Administrator** aux utilisateurs non privilégiés afin qu'ils ne puissent pas gérer les ressources, les utilisateurs et les projets de la grappe. De même, vous pouvez afficher la perspective **Developer** aux utilisateurs ayant le rôle de développeur afin qu'ils puissent créer, déployer et surveiller des applications.

Vous pouvez également personnaliser la visibilité de la perspective pour les utilisateurs en fonction du contrôle d'accès basé sur les rôles (RBAC). Par exemple, si vous personnalisez une perspective à des fins de surveillance, ce qui nécessite des autorisations spécifiques, vous pouvez définir que la perspective n'est visible que par les utilisateurs disposant des autorisations requises.

Chaque perspective comprend les paramètres obligatoires suivants, que vous pouvez modifier dans la vue YAML :

- **id**: Définit l'ID de la perspective à afficher ou à masquer
- **visibility**: Définit l'état de la perspective ainsi que les contrôles d'accès, si nécessaire
- **state**: Définit si la perspective est activée, désactivée ou si elle doit faire l'objet d'un contrôle d'accès



NOTE

Par défaut, toutes les perspectives sont activées. Lorsque vous personnalisez la perspective de l'utilisateur, vos modifications s'appliquent à l'ensemble du cluster.

6.9.1. Personnalisation d'une perspective à l'aide d'une vue YAML

Conditions préalables

- Vous devez avoir des privilèges d'administrateur.

Procédure

1. Dans la perspective **Administrator**, naviguez vers **Administration** → **Cluster Settings**.
2. Sélectionnez l'onglet **Configuration** et cliquez sur la ressource **Console** (**operator.openshift.io**).
3. Cliquez sur l'onglet **YAML** et effectuez votre personnalisation :
 - a. Pour activer ou désactiver une perspective, insérez l'extrait pour **Add user perspectives** et modifiez le code YAML si nécessaire :

```
apiVersion: operator.openshift.io/v1
kind: Console
metadata:
  name: cluster
spec:
  customization:
    perspectives:
      - id: admin
        visibility:
          state: Enabled
      - id: dev
        visibility:
          state: Enabled
```

- b. Pour masquer une perspective en fonction des autorisations RBAC, insérez l'extrait pour **Hide user perspectives** et modifiez le code YAML si nécessaire :

```
apiVersion: operator.openshift.io/v1
kind: Console
metadata:
  name: cluster
spec:
  customization:
    perspectives:
      - id: admin
        requiresAccessReview:
          - group: rbac.authorization.k8s.io
            resource: clusterroles
            verb: list
      - id: dev
        state: Enabled
```

- c. Pour personnaliser une perspective en fonction de vos besoins, créez votre propre extrait YAML :

```
apiVersion: operator.openshift.io/v1
kind: Console
metadata:
  name: cluster
spec:
  customization:
    perspectives:
      - id: admin
        visibility:
          state: AccessReview
          accessReview:
            missing:
              - resource: deployment
                verb: list
            required:
              - resource: namespaces
                verb: list
      - id: dev
        visibility:
          state: Enabled
```

4. Cliquez sur **Save**.

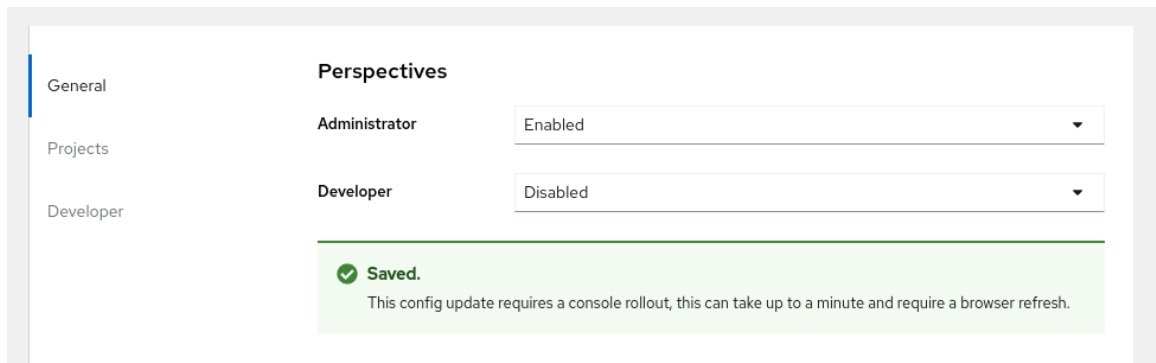
6.9.2. Personnalisation d'une perspective à l'aide de la vue de formulaire

Conditions préalables

- Vous devez avoir des privilèges d'administrateur.

Procédure

1. Dans la perspective **Administrator**, naviguez vers **Administration → Cluster Settings**.
2. Sélectionnez l'onglet **Configuration** et cliquez sur la ressource **Console (operator.openshift.io)**.
3. Cliquez sur **Actions → Customize** à droite de la page.
4. Dans les paramètres de **General**, personnalisez la perspective en sélectionnant l'une des options suivantes dans la liste déroulante :
 - **Enabled**: Active la perspective pour tous les utilisateurs
 - **Only visible for privileged users**: Active la perspective pour les utilisateurs qui peuvent lister tous les espaces de noms
 - **Only visible for unprivileged users**: Active la perspective pour les utilisateurs qui ne peuvent pas lister tous les espaces de noms
 - **Disabled**: Désactive la perspective pour tous les utilisateurs
 Une notification s'ouvre pour confirmer l'enregistrement des modifications.



NOTE

Lorsque vous personnalisez le point de vue de l'utilisateur, vos modifications sont automatiquement enregistrées et prennent effet après une actualisation du navigateur.

6.10. PERSONNALISATION DU CATALOGUE ET DU SOUS-CATALOGUE DU DÉVELOPPEUR

En tant qu'administrateur de cluster, vous avez la possibilité d'organiser et de gérer le catalogue des développeurs ou ses sous-catalogues. Vous pouvez activer ou désactiver les types de sous-catalogues ou désactiver l'ensemble du catalogue de développeurs.

L'objet **developerCatalog.types** comprend les paramètres suivants que vous devez définir dans un extrait pour les utiliser dans la vue YAML :

- **state**: Définit si une liste de types de catalogues de développeurs doit être activée ou désactivée.
- **enabled**: Définit une liste de types de catalogues de développeurs (sous-catalogues) visibles par les utilisateurs.
- **disabled**: Définit une liste de types de catalogues de développeurs (sous-catalogues) qui ne sont pas visibles pour les utilisateurs.

Vous pouvez activer ou désactiver les types de catalogues de développeurs suivants (sous-catalogues) à l'aide de la vue YAML ou de la vue formulaire.

- **Builder Images**
- **Templates**
- **Devfiles**
- **Samples**
- **Helm Charts**
- **Event Sources**
- **Event Sinks**
- **Operator Backed**

6.10.1. Personnaliser un catalogue de développeurs ou ses sous-catalogues à l'aide de la vue YAML

Vous pouvez personnaliser un catalogue de développeurs en modifiant le contenu YAML dans la vue YAML.

Conditions préalables

- Une session de console web OpenShift avec des privilèges d'administrateur de cluster.

Procédure

1. Dans la perspective **Administrator** de la console web, naviguez vers **Administration → Cluster Settings**.
2. Sélectionnez l'onglet **Configuration**, cliquez sur la ressource **Console (operator.openshift.io)** et consultez la page **Details**.
3. Cliquez sur l'onglet **YAML** pour ouvrir l'éditeur et modifier le contenu YAML si nécessaire.
Par exemple, pour désactiver un type de catalogue de développeurs, insérez l'extrait suivant qui définit une liste de ressources de catalogue de développeurs désactivées :

```
apiVersion: operator.openshift.io/v1
kind: Console
metadata:
  name: cluster
...
spec:
  customization:
    developerCatalog:
      categories:
        types:
          state: Disabled
          disabled:
            - BuilderImage
            - Devfile
            - HelmChart
...
```

4. Cliquez sur **Save**.



NOTE

Par défaut, les types de catalogues de développeurs sont activés dans la vue Administrateur de la console Web.

6.10.2. Personnaliser un catalogue de développeurs ou ses sous-catalogues à l'aide de la vue formulaire

Vous pouvez personnaliser un catalogue de développeurs en utilisant la vue formulaire de la Web Console.

Conditions préalables

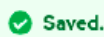
- Une session de console web OpenShift avec des privilèges d'administrateur de cluster.

Procédure

1. Dans la perspective **Administrator**, naviguez vers **Administration** → **Cluster Settings**.
2. Sélectionnez l'onglet **Configuration** et cliquez sur la ressource **Console (operator.openshift.io)**.
3. Cliquez sur **Actions** → **Customize**.
4. Dans les sections correspondantes, activez ou désactivez les éléments de la liste.

Vérification

Après avoir personnalisé le catalogue du développeur, vos modifications sont automatiquement enregistrées dans le système et prennent effet dans le navigateur après une actualisation.



Saved.

This config update requires a console rollout, this can take up to a minute and require a browser refresh.



NOTE

Vous pouvez utiliser une procédure similaire pour personnaliser les éléments de l'interface utilisateur Web tels que les démarrages rapides, les rôles de cluster et les actions.

6.10.2.1. Exemple de modifications du fichier YAML

Vous pouvez ajouter dynamiquement les extraits suivants dans l'éditeur YAML pour personnaliser un catalogue de développeurs.

Utilisez l'extrait suivant pour afficher tous les sous-catalogues en définissant le type *state* sur **Enabled**.

```
apiVersion: operator.openshift.io/v1
kind: Console
metadata:
  name: cluster
...
spec:
  customization:
    developerCatalog:
      categories:
        types:
          state: Enabled
```

Utilisez l'extrait suivant pour désactiver tous les sous-catalogues en définissant le type *state* sur **Disabled**:

```
apiVersion: operator.openshift.io/v1
kind: Console
metadata:
  name: cluster
...
```

```
spec:
  customization:
    developerCatalog:
      categories:
      types:
      state: Disabled
```

Utilisez l'extrait suivant lorsqu'un administrateur de cluster définit une liste de sous-catalogues, qui sont activés dans la console Web.

```
apiVersion: operator.openshift.io/v1
kind: Console
metadata:
  name: cluster
...
spec:
  customization:
    developerCatalog:
      categories:
      types:
      state: Enabled
      enabled:
        - BuilderImage
        - Devfile
        - HelmChart
        - ...
```

CHAPITRE 7. PLUGINS DYNAMIQUES

7.1. APERÇU DES PLUGINS DYNAMIQUES

7.1.1. A propos des plugins dynamiques

Un plugin dynamique vous permet d'ajouter des pages personnalisées et d'autres extensions à votre interface au moment de l'exécution. La ressource personnalisée **ConsolePlugin** enregistre les plugins avec la console, et un administrateur de cluster active les plugins dans la configuration **console-operator**.

7.1.2. Caractéristiques principales

Un plugin dynamique vous permet d'apporter les personnalisations suivantes à l'expérience OpenShift Container Platform :

- Ajouter des pages personnalisées.
- Ajouter des perspectives au-delà de l'administrateur et du développeur.
- Ajouter des éléments de navigation.
- Ajouter des onglets et des actions aux pages de ressources.

7.1.3. Lignes directrices générales

Lors de la création de votre plugin, suivez ces lignes directrices générales :

- **Node.js** et **yarn** sont nécessaires à la construction et à l'exécution de votre plugin.
- Préfixez les noms de vos classes CSS par le nom de votre plugin pour éviter les collisions. Par exemple, **my-plugin__heading** et **my-plugin__icon**.
- Veiller à la cohérence de l'aspect, de la convivialité et du comportement avec les autres pages de la console.
- Suivez les directives de localisation de [react-i18next](#) lors de la création de votre plugin. Vous pouvez utiliser le crochet **useTranslation** comme dans l'exemple suivant :

```
const Header: React.FC = () => {
  const { t } = useTranslation('plugin__console-demo-plugin');
  return <h1>{t('Hello, World!')}</h1>;
};
```

- Évitez les sélecteurs qui pourraient affecter le balisage en dehors des composants de votre plugin, tels que les sélecteurs d'éléments. Ce ne sont pas des API et ils sont susceptibles d'être modifiés. Leur utilisation peut endommager votre plugin. Évitez les sélecteurs comme les sélecteurs d'éléments qui pourraient affecter le balisage en dehors des composants de votre plugin.

Lignes directrices de PatternFly 4

Lors de la création de votre plugin, suivez les instructions suivantes pour l'utilisation de PatternFly :

- Utilisez les composants [PatternFly4](#) et les variables CSS PatternFly. Les composants de base de PatternFly sont disponibles via le SDK. L'utilisation des composants et des variables PatternFly permet à votre plugin d'être cohérent dans les futures versions de la console.
- Rendez votre plugin accessible en suivant les [principes fondamentaux d'accessibilité de PatternFly](#).
- Évitez d'utiliser d'autres bibliothèques CSS telles que Bootstrap ou Tailwind. Elles peuvent entrer en conflit avec PatternFly et ne correspondront pas à l'apparence de la console.

7.2. DÉMARRER AVEC DES PLUGINS DYNAMIQUES

Pour commencer à utiliser le plugin dynamique, vous devez configurer votre environnement pour écrire un nouveau plugin dynamique OpenShift Container Platform. Pour un exemple d'écriture d'un nouveau plugin, voir [Adding a tab to the pods page](#).

7.2.1. Développement de plugins dynamiques

Vous pouvez exécuter le plugin en utilisant un environnement de développement local. La console web d'OpenShift Container Platform s'exécute dans un conteneur connecté au cluster auquel vous vous êtes connecté.

Conditions préalables

- Vous devez avoir un cluster OpenShift en cours d'exécution.
- Il faut que la CLI OpenShift (**oc**) soit installée.
- Vous devez avoir [yarn](#) installé.
- Vous devez avoir [Docker](#) v3.2.0 ou plus récent ou [Podman](#) installé et en cours d'exécution.

Procédure

1. Dans votre terminal, exécutez la commande suivante pour installer les dépendances de votre plugin en utilisant yarn.

```
$ yarn install
```

2. Après l'installation, lancez la commande suivante pour démarrer Yarn.

```
$ yarn run start
```

3. Dans une autre fenêtre de terminal, connectez-vous à OpenShift Container Platform via le CLI.

```
$ oc login
```

4. Exécutez la console web de OpenShift Container Platform dans un conteneur connecté au cluster dans lequel vous vous êtes connecté en exécutant la commande suivante :

```
$ yarn run start-console
```

Vérification

- Visitez localhost:9000 pour voir le plugin en cours d'exécution. Inspectez la valeur de **window.SERVER_FLAGS.consolePlugins** pour voir la liste des plugins qui se chargent au moment de l'exécution.

7.3. DÉPLOYER VOTRE PLUGIN SUR UN CLUSTER

Vous pouvez déployer le plugin sur un cluster OpenShift Container Platform.

7.3.1. Construire une image avec Docker

Pour déployer votre plugin sur un cluster, vous devez construire une image et la pousser vers un registre d'images.

Procédure

1. Créez l'image à l'aide de la commande suivante :

```
$ docker build -t quay.io/my-repository/my-plugin:latest .
```

2. Facultatif : si vous souhaitez tester votre image, exécutez la commande suivante :

```
$ docker run -it --rm -d -p 9001:80 quay.io/my-repository/my-plugin:latest
```

3. Poussez l'image en exécutant la commande suivante :

```
$ docker push quay.io/my-repository/my-plugin:latest
```

7.3.2. Déployer votre plugin sur un cluster

Après avoir transféré une image contenant vos modifications dans un registre, vous pouvez déployer le plugin dans un cluster.

Procédure

1. Pour déployer votre plugin sur un cluster, installez un graphique Helm avec le nom du plugin comme nom de version Helm dans un nouvel espace de noms ou un espace de noms existant comme spécifié par l'option de ligne de commande **-n**. Indiquez l'emplacement de l'image dans le paramètre **plugin.image** en utilisant la commande suivante :

```
$ helm upgrade -i my-plugin charts/openshift-console-plugin -n my-plugin-namespace --create-namespace --set plugin.image=my-plugin-image-location
```

Où ?

n <my-plugin-namespace>

Spécifie un espace de noms existant dans lequel déployer votre plugin.

--create-namespace

Facultatif : si le déploiement se fait dans un nouvel espace de noms, utilisez ce paramètre.

--set plugin.image=my-plugin-image-location

Spécifie l'emplacement de l'image dans le paramètre **plugin.image**.

2. Facultatif : vous pouvez spécifier des paramètres supplémentaires en utilisant l'ensemble des paramètres pris en charge dans le fichier **charts/openshift-console-plugin/values.yaml**.

```

plugin:
  name: ""
  description: ""
  image: ""
  imagePullPolicy: IfNotPresent
  replicas: 2
  port: 9443
  securityContext:
    enabled: true
  podSecurityContext:
    enabled: true
    runAsNonRoot: true
    seccompProfile:
      type: RuntimeDefault
  containerSecurityContext:
    enabled: true
    allowPrivilegeEscalation: false
    capabilities:
      drop:
        - ALL
  resources:
    requests:
      cpu: 10m
      memory: 50Mi
  basePath: /
  certificateSecretName: ""
  serviceAccount:
    create: true
    annotations: {}
    name: ""
  patcherServiceAccount:
    create: true
    annotations: {}
    name: ""
  jobs:
    patchConsoles:
      enabled: true
      image: "registry.redhat.io/openshift4/ose-tools-
rhel8@sha256:e44074f21e0cca6464e50cb6ff934747e0bd11162ea01d522433a1a1ae116103"
    podSecurityContext:
      enabled: true
      runAsNonRoot: true
      seccompProfile:
        type: RuntimeDefault
    containerSecurityContext:
      enabled: true
      allowPrivilegeEscalation: false
      capabilities:
        drop:
          - ALL
    resources:

```

```
requests:
  cpu: 10m
  memory: 50Mi
```

Vérification

Vous pouvez consulter la liste des plugins activés sur la page **Overview** ou en naviguant à partir de **Administration** → **Cluster Settings** → **Configuration** → **Console operator.openshift.io** → **Console plugins**.



NOTE

Il peut s'écouler quelques minutes avant que la nouvelle configuration du plugin n'apparaisse. Si vous ne voyez pas votre plugin, il se peut que vous deviez rafraîchir votre navigateur si le plugin a été récemment activé. Si vous recevez des erreurs lors de l'exécution, vérifiez la console JS dans les outils de développement du navigateur pour rechercher d'éventuelles erreurs dans le code de votre plugin.

7.3.3. Désactiver votre plugin dans le navigateur

Les utilisateurs de la console peuvent utiliser le paramètre de requête **disable-plugins** pour désactiver certains ou tous les plugins dynamiques qui seraient normalement chargés au moment de l'exécution.

Procédure

- Pour désactiver un ou plusieurs plugins spécifiques, supprimez le plugin que vous souhaitez désactiver de la liste des noms de plugins séparés par des virgules.
- Pour désactiver tous les plugins, laissez une chaîne vide dans le paramètre de requête **disable-plugins**.



NOTE

Les administrateurs de clusters peuvent désactiver les plugins dans la page **Cluster Settings** de la console web

7.4. EXEMPLE DE PLUGIN DYNAMIQUE

Avant de suivre l'exemple, vérifiez que le plugin fonctionne en suivant les étapes du [développement dynamique du plugin](#)

7.4.1. Ajout d'un onglet à la page des pods

Il existe différentes personnalisations que vous pouvez apporter à la console web d'OpenShift Container Platform. La procédure suivante ajoute un onglet à la page **Pod details** en tant qu'exemple d'extension de votre plugin.



NOTE

La console web d'OpenShift Container Platform s'exécute dans un conteneur connecté au cluster auquel vous vous êtes connecté. Voir "Dynamic plugin development" pour des informations permettant de tester le plugin avant de créer le vôtre.

Procédure

1. Visitez le dépôt [console-plugin-template](#) qui contient un modèle pour créer des plugins dans un nouvel onglet.



IMPORTANT

Le code des plugins personnalisés n'est pas pris en charge par Red Hat. Seul le [support de la communauté Cooperative](#) est disponible pour votre plugin.

2. Sélectionnez le bouton déroulant **Use this template** et sélectionnez **Create new repository** dans la liste déroulante pour créer un dépôt GitHub.
3. Renommez le nouveau dépôt avec le nom de votre plugin.
4. Clonez votre dépôt copié sur votre machine locale afin de pouvoir éditer le code.
5. Modifiez les métadonnées du plugin dans la déclaration **consolePlugin** de **package.json**.

```
"consolePlugin": {
  "name": "my-plugin", 1
  "version": "0.0.1", 2
  "displayName": "My Plugin", 3
  "description": "Enjoy this shiny, new console plugin!", 4
  "exposedModules": {
    "ExamplePage": "./components/ExamplePage"
  },
  "dependencies": {
    "@console/pluginAPI": "/"
  }
}
```

- 1 Mettez à jour le nom de votre plugin.
- 2 Mettre à jour la version.
- 3 Mettez à jour le nom d'affichage de votre plugin.
- 4 Mettez à jour la description avec un résumé de votre plugin.

6. Ajoutez ce qui suit au fichier **console-extensions.json**:

```
{
  "type": "console.tab/horizontalNav",
  "properties": {
    "page": {
      "name": "Example Tab",
      "href": "example"
    },
    "model": {
      "group": "core",
      "version": "v1",
      "kind": "Pod"
    }
  },
}
```

```
"component": { "$codeRef": "ExampleTab" }
}
```

7. Modifiez le fichier **package.json** pour y inclure les changements suivants :

```
"exposedModules": {
  "ExamplePage": "./components/ExamplePage",
  "ExampleTab": "./components/ExampleTab"
}
```

8. Écrivez un message à afficher dans un nouvel onglet personnalisé sur la page **Pods** en créant un nouveau fichier **src/components/ExampleTab.tsx** et en y ajoutant le script suivant :

```
import * as React from 'react';

export default function ExampleTab() {
  return (
    <p>This is a custom tab added to a resource using a dynamic plugin.</p>
  );
}
```

Vérification

- Visitez une page **Pod** pour voir l'onglet ajouté.

7.5. RÉFÉRENCE DU PLUGIN DYNAMIQUE

Vous pouvez ajouter des extensions qui vous permettent de personnaliser votre plugin. Ces extensions sont ensuite chargées dans la console au moment de l'exécution.

7.5.1. Types d'extensions dynamiques de plugins

console.action/filter

ActionFilter peut être utilisé pour filtrer une action.

Nom	Type de valeur	En option	Description
contextId	string	non	L'identifiant de contexte permet de limiter la portée des actions de contribution à un domaine particulier de l'application. Les exemples incluent topology et helm .

Nom	Type de valeur	En option	Description
filter	CodeRef<(scope: any, action: Action) ⇒ boolean>	non	Une fonction qui filtrera les actions sur la base de certaines conditions. scope : Le champ d'application dans lequel les actions doivent être prévues. Un crochet peut être nécessaire si vous souhaitez supprimer l'action ModifyCount d'un déploiement avec un pod autoscaler horizontal (HPA).

console.action/group

ActionGroup contribue à un groupe d'actions qui peut également être un sous-menu.

Nom	Type de valeur	En option	Description
id	string	non	ID utilisé pour identifier la section d'action.
label	string	yes	L'étiquette à afficher dans l'interface utilisateur. Nécessaire pour les sous-menus.
submenu	boolean	yes	Indique si ce groupe doit être affiché en tant que sous-menu.
insertBefore	string string[]	yes	Insérer cet élément avant l'élément référencé ici. Pour les tableaux, le premier élément trouvé dans l'ordre est utilisé.
insertAfter	string string[]	yes	Insérer cet élément après l'élément référencé ici. Pour les tableaux, le premier élément trouvé dans l'ordre est utilisé. La valeur insertBefore est prioritaire.

console.action/provider

ActionProvider apporte un crochet qui renvoie une liste d'actions pour un contexte spécifique.

Nom	Type de valeur	En option	Description
contextId	string	non	L'identifiant de contexte permet de limiter la portée des actions de contribution à un domaine particulier de l'application. Les exemples incluent topology et helm .
provider	CodeRef<Extension Hook<Action[], any>>	non	Un crochet React qui renvoie des actions pour le champ d'application donné. Si contextId = resource , le champ d'application sera toujours un objet ressource Kubernetes.

console.action/resource-provider

ResourceActionProvider apporte un crochet qui renvoie une liste d'actions pour un modèle de ressource spécifique.

Nom	Type de valeur	En option	Description
model	ExtensionK8sKindVersionModel	non	Le modèle pour lequel ce prestataire fournit des actions.
provider	CodeRef<Extension Hook<Action[], any>>	non	Un hook react qui renvoie des actions pour le modèle de ressource donné

console.alert-action

Nom	Type de valeur	En option	Description
alert	string	non	
text	string	non	
action	CodeRef<(alert: any) ⇒ void>	non	

console.catalog/item-filter

Nom	Type de valeur	En option	Description
catalogId	string string[]	non	L'identifiant unique du catalogue auquel ce fournisseur contribue.
type	string	non	ID de type pour le type d'article du catalogue.
filter	CodeRef<(item: CatalogItem) ⇒ boolean>	non	Filtre les éléments d'un type spécifique. Valeur est une fonction qui prend CatalogItem[] et renvoie un sous-ensemble basé sur les critères de filtrage.

console.catalog/item-metadata

Nom	Type de valeur	En option	Description
catalogId	string string[]	non	L'identifiant unique du catalogue auquel ce fournisseur contribue.
type	string	non	ID de type pour le type d'article du catalogue.
provider	CodeRef<Extension Hook<CatalogItemMetadataProviderFunction, CatalogExtensionHookOptions>>	non	Un crochet qui renvoie une fonction qui sera utilisée pour fournir des métadonnées aux éléments de catalogue d'un type spécifique.

console.catalog/item-provider

Nom	Type de valeur	En option	Description
catalogId	string string[]	non	L'identifiant unique du catalogue auquel ce fournisseur contribue.
type	string	non	ID de type pour le type d'article du catalogue.
title	string	non	Titre du fournisseur de l'article de catalogue

Nom	Type de valeur	En option	Description
provider	CodeRef<ExtensionHook<CatalogItem<any>[], CatalogExtensionHookOptions>>	non	Récupère les éléments et les normalise pour le catalogue. La valeur est un crochet d'effet react.
priority	number	yes	Priorité pour ce fournisseur. La valeur par défaut est 0 . Les fournisseurs ayant une priorité plus élevée peuvent remplacer les éléments de catalogue fournis par d'autres fournisseurs.

console.catalog/item-type

Nom	Type de valeur	En option	Description
type	string	non	Type d'élément du catalogue.
title	string	non	Titre de l'article du catalogue.
catalogDescription	string CodeRef<React.ReactNode>	yes	Description du catalogue spécifique au type.
typeDescription	string	yes	Description du type d'article du catalogue.
filters	CatalogItemAttribute []	yes	Filtres personnalisés spécifiques à l'article du catalogue.
groupings	CatalogItemAttribute []	yes	Regroupements personnalisés spécifiques à l'article du catalogue.

console.catalog/item-type-metadata

Nom	Type de valeur	En option	Description
type	string	non	Type d'élément du catalogue.
filters	CatalogItemAttribute []	yes	Filtres personnalisés spécifiques à l'article du catalogue.
groupings	CatalogItemAttribute []	yes	Regroupements personnalisés spécifiques à l'article du catalogue.

console.cluster-overview/inventory-item

Ajoute un nouvel élément d'inventaire à la page d'aperçu du cluster.

Nom	Type de valeur	En option	Description
component	CodeRef<React.ComponentType<{}>>	non	Le composant à rendre.

console.cluster-overview/multiline-utilization-item

Ajoute un nouvel élément d'utilisation multi-lignes de la vue d'ensemble du cluster.

Nom	Type de valeur	En option	Description
title	string	non	Titre de l'élément d'utilisation.
getUtilizationQueries	CodeRef<GetMultilineQueries>	non	Demande d'utilisation du Prométhée.
humanize	CodeRef<Humanize>	non	Convertir les données Prometheus en format lisible par l'homme.
TopConsumerPopovers	CodeRef<React.ComponentType<TopConsumerPopoverProps>[]>	yes	Affiche le popover Top consumer au lieu d'une simple valeur

console.cluster-overview/utilization-item

Ajoute un nouvel élément d'utilisation de la vue d'ensemble du cluster.

Nom	Type de valeur	En option	Description
title	string	non	Titre de l'élément d'utilisation.
getUtilizationQuery	CodeRef<GetQuery>	non	Demande d'utilisation du Prométhée.
humanize	CodeRef<Humanize>	non	Convertir les données Prometheus en format lisible par l'homme.
getTotalQuery	CodeRef<GetQuery>	yes	Requête totale de Prometheus.
getRequestQuery	CodeRef<GetQuery>	yes	Requête de demande de Prométhée.
getLimitQuery	CodeRef<GetQuery>	yes	Demande de limite de Prométhée.
TopConsumerPopover	CodeRef<React.ComponentType<TopConsumerPopoverProps>>	yes	Affiche le popover Top consumer au lieu d'une simple valeur

console.context-provider

Ajoute un nouveau fournisseur de contexte React à la racine de l'application de la console web.

Nom	Type de valeur	En option	Description
provider	CodeRef<Provider<T>>	non	Composant du fournisseur de contexte.
useValueHook	CodeRef<() ⇒ T>	non	Crochet pour la valeur du contexte.

console.dashboards/card

Ajoute une nouvelle carte de tableau de bord.

Nom	Type de valeur	En option	Description
tab	string	non	L'ID de l'onglet du tableau de bord auquel la carte sera ajoutée.
position	'LEFT' 'RIGHT' 'MAIN'	non	La position de la carte sur le tableau de bord.

Nom	Type de valeur	En option	Description
component	CodeRef<React.ComponentType<{}>>	non	Composant de la carte du tableau de bord.
span	OverviewCardSpan	yes	Espace vertical de la carte dans la colonne. Ignoré pour les petits écrans ; la valeur par défaut est 12 .

console.dashboards/overview/activity/resource

Ajoute une activité à la carte d'activité du tableau de bord de la vue d'ensemble où le déclenchement de l'activité est basé sur l'observation d'une ressource Kubernetes.

Nom	Type de valeur	En option	Description
k8sResource	CodeRef<FirehoseResource & { isList: true; }>	non	Le poste d'utilisation à remplacer.
component	CodeRef<React.ComponentType<K8sActivityProps<T>>>	non	La composante action.
isActivity	CodeRef<(resource: T) ⇒ boolean>	yes	Fonction qui détermine si la ressource donnée représente l'action. Si elle n'est pas définie, chaque ressource représente l'activité.
getTimestamp	CodeRef<(resource: T) ⇒ Date>	yes	Horodatage de l'action donnée, qui sera utilisé pour la commande.

console.dashboards/overview/detail/item

Ajoute un élément à la carte **Details** du tableau de bord **Overview**.

Nom	Type de valeur	En option	Description
component	CodeRef<React.ComponentType<{}>>	non	La valeur, basée sur la composante DetaillItem

console.dashboards/overview/health/operator

Ajoute un sous-système de santé à la carte d'état du tableau de bord **Overview**, où la source d'état est une API REST Kubernetes.

Nom	Type de valeur	En option	Description
title	string	non	Titre de la section Opérateurs dans le menu contextuel.
resources	CodeRef<FirehoseResource[]>	non	Ressources Kubernetes qui seront récupérées et transmises à healthHandler .
getOperatorsWithStatuses	CodeRef<GetOperatorsWithStatuses<T>>	yes	Résout les problèmes de statut des opérateurs.
operatorRowLoader	CodeRef<React.ComponentType<OperatorRowProps<T>>>	yes	Chargeur pour le composant "pop-up row".
viewAllLink	string	yes	Liens vers la page de toutes les ressources. Si elle n'est pas fournie, la page de liste de la première ressource de la liste des ressources est utilisée.

console.dashboards/overview/health/prometheus

Ajoute un sous-système de santé à la carte d'état d'un tableau de bord de synthèse dont la source d'état est Prometheus.

Nom	Type de valeur	En option	Description
title	string	non	Nom d'affichage du sous-système.
queries	string[]	non	Les requêtes de Prométhée
healthHandler	CodeRef<PrometheusHealthHandler>	non	Résoudre le problème de santé du sous-système.
additionalResource	CodeRef<FirehoseResource>	yes	Ressource supplémentaire qui sera recherchée et transmise à healthHandler .

Nom	Type de valeur	En option	Description
popupComponent	CodeRef<React.ComponentType<PrometheusHealthPopupProps>>	yes	Chargeur pour le contenu du menu contextuel. S'il est défini, un élément de santé est représenté par un lien qui ouvre un menu contextuel avec le contenu donné.
popupTitle	string	yes	Le titre du popover.
disallowedControlPlaneTopology	string[]	yes	Topologie du plan de contrôle pour laquelle le sous-système doit être masqué.

console.dashboards/overview/health/resource

Ajoute un sous-système de santé à la carte d'état du tableau de bord Overview lorsque la source d'état est une ressource Kubernetes.

Nom	Type de valeur	En option	Description
title	string	non	Nom d'affichage du sous-système.
resources	CodeRef<WatchK8sResources<T>>	non	Ressources Kubernetes qui seront récupérées et transmises à healthHandler .
healthHandler	CodeRef<ResourceHealthHandler<T>>	non	Résoudre le problème de santé du sous-système.
popupComponent	CodeRef<WatchK8sResults<T>>	yes	Chargeur pour le contenu du menu contextuel. S'il est défini, un élément de santé est représenté par un lien qui ouvre un menu contextuel avec le contenu donné.
popupTitle	string	yes	Le titre du popover.

console.dashboards/overview/health/url

Ajoute un sous-système de santé à la carte d'état du tableau de bord Overview lorsque la source d'état est une API REST Kubernetes.

Nom	Type de valeur	En option	Description
title	string	non	Nom d'affichage du sous-système.
url	string	non	L'URL à partir de laquelle les données doivent être récupérées. Elle sera préfixée par l'URL de base de Kubernetes.
healthHandler	<code>`CodeRef<URLHealthHandler<T, K8sResourceCommon>`</code>	<code>K8sResourceCommon[]>>`</code>	non
Résoudre le problème de santé du sous-système.	additionalResource	CodeRef<FirehoseResource>	yes
Ressource supplémentaire qui sera recherchée et transmise à healthHandler .	popupComponent	CodeRef<React.ComponentType<{ healthResult?: T; healthResultError?: any; k8sResult?: FirehoseResult<R>; }>>	yes
Chargeur pour le contenu de la fenêtre contextuelle. S'il est défini, un élément de santé sera représenté par un lien qui ouvrira une fenêtre contextuelle avec le contenu donné.	popupTitle	string	yes

console.dashboards/overview/inventory/item

Ajoute une tuile de ressource à la carte d'inventaire de la vue d'ensemble.

Nom	Type de valeur	En option	Description
model	CodeRef<T>	non	Le modèle de resource qui sera récupéré. Utilisé pour obtenir le modèle label ou abbr .

Nom	Type de valeur	En option	Description
mapper	CodeRef<StatusGroupMapper<T, R>>	yes	Fonction qui associe différents statuts à des groupes.
additionalResources	CodeRef<WatchK8sResources<R>>	yes	Ressources supplémentaires qui seront récupérées et transmises à la fonction mapper .

console.dashboards/overview/inventory/item/group

Ajoute un groupe d'état d'inventaire.

Nom	Type de valeur	En option	Description
id	string	non	L'identifiant du groupe d'état.
icon	CodeRef<React.ReactElement<any, string React.JSXElementConstructor<any>>>	non	Composant React représentant l'icône du groupe d'état.

console.dashboards/overview/inventory/item/replacement

Remplace une carte d'inventaire.

Nom	Type de valeur	En option	Description
model	CodeRef<T>	non	Le modèle de resource qui sera récupéré. Utilisé pour obtenir le modèle label ou abbr .
mapper	CodeRef<StatusGroupMapper<T, R>>	yes	Fonction qui associe différents statuts à des groupes.
additionalResources	CodeRef<WatchK8sResources<R>>	yes	Ressources supplémentaires qui seront récupérées et transmises à la fonction mapper .

console.dashboards/overview/prometheus/activity/resource

Ajoute une activité à la carte d'activité du tableau de bord Prometheus Overview où le déclenchement de l'activité est basé sur l'observation d'une ressource Kubernetes.

Nom	Type de valeur	En option	Description
queries	string[]	non	Requêtes à surveiller
component	CodeRef<React.ComponentType<PrometheusActivityProps>>	non	La composante action.
isActivity	CodeRef<(results: PrometheusResponse[]) ⇒ boolean>	yes	Fonction qui détermine si la ressource donnée représente l'action. Si elle n'est pas définie, chaque ressource représente l'activité.

console.dashboards/project/overview/item

Ajoute une tuile de ressource à la carte d'inventaire de la vue d'ensemble du projet.

Nom	Type de valeur	En option	Description
model	CodeRef<T>	non	Le modèle de resource qui sera récupéré. Utilisé pour obtenir le modèle label ou abbr .
mapper	CodeRef<StatusGroupMapper<T, R>>	yes	Fonction qui associe différents statuts à des groupes.
additionalResources	CodeRef<WatchK8sResources<R>>	yes	Ressources supplémentaires qui seront récupérées et transmises à la fonction mapper .

console.dashboards/tab

Ajoute un nouvel onglet au tableau de bord, placé après l'onglet **Overview**.

Nom	Type de valeur	En option	Description
id	string	non	Identifiant unique de l'onglet, utilisé comme lien vers l'onglet href et lors de l'ajout de cartes à cet onglet.
navSection	'home' 'storage'	non	Section de navigation à laquelle appartient l'onglet.

Nom	Type de valeur	En option	Description
title	string	non	Le titre de l'onglet.

console.file-upload

Nom	Type de valeur	En option	Description
fileExtensions	string[]	non	Extensions de fichiers prises en charge.
handler	CodeRef<FileUpload Handler>	non	Fonction qui gère l'action de dépôt de fichier.

console.flag

Permet de contrôler entièrement les drapeaux des fonctions de la console web.

Nom	Type de valeur	En option	Description
handler	CodeRef<FeatureFlagHandler>	non	Permet d'activer ou de désactiver des drapeaux de caractéristiques arbitraires.

console.flag/hookProvider

Permet de contrôler entièrement les indicateurs de fonctionnalité de la console web avec des gestionnaires de crochets.

Nom	Type de valeur	En option	Description
handler	CodeRef<FeatureFlagHandler>	non	Permet d'activer ou de désactiver des drapeaux de caractéristiques arbitraires.

console.flag/model

Ajout d'une nouvelle fonctionnalité de la console web, pilotée par la présence d'un CRD sur le cluster.

Nom	Type de valeur	En option	Description
flag	string	non	Nom de l'indicateur à activer lorsque le CRD est détecté.

Nom	Type de valeur	En option	Description
model	ExtensionK8sModel	non	Le modèle qui se réfère à une CustomResourceDefinition .

console.global-config

Nom	Type de valeur	En option	Description
id	string	non	Identifiant unique de l'instance de ressource de configuration de cluster.
name	string	non	Le nom de l'instance de ressource de configuration de cluster.
model	ExtensionK8sModel	non	Le modèle qui se réfère à une ressource de configuration de cluster.
namespace	string	non	L'espace de noms de l'instance de ressource de configuration de cluster.

console.model-metadata

Personnaliser l'affichage des modèles en remplaçant les valeurs récupérées et générées par la découverte de l'API.

Nom	Type de valeur	En option	Description
model	ExtensionK8sGroup Model	non	Le modèle à personnaliser. Peut ne spécifier qu'un groupe, ou une version et un type optionnels.
badge	ModelBadge	yes	Considérer ou non ce modèle de référence comme un Technology Preview ou un Developer Preview.
color	string	yes	La couleur à associer à ce modèle.

Nom	Type de valeur	En option	Description
label	string	yes	Remplacer l'étiquette. Nécessite que kind soit fourni.
labelPlural	string	yes	Remplacer l'étiquette plurielle. Exige que kind soit fourni.
abbr	string	yes	Personnalisez l'abréviation. Par défaut, tous les caractères majuscules de kind sont utilisés, dans la limite de 4 caractères. Nécessite que kind soit fourni.

console.navigation/href

Nom	Type de valeur	En option	Description
id	string	non	Un identifiant unique pour cet élément.
name	string	non	Le nom de cet élément.
href	string	non	La valeur de la référence du lien.
perspective	string	yes	L'identifiant de la perspective à laquelle cet élément appartient. Si elle n'est pas spécifiée, elle contribue à la perspective par défaut.
section	string	yes	Section de navigation à laquelle cet élément appartient. S'il n'est pas spécifié, cet élément est rendu comme un lien de premier niveau.
dataAttributes	{ [key: string]: string; }	yes	Ajoute des attributs de données au DOM.

Nom	Type de valeur	En option	Description
startsWith	string[]	yes	Marquer cet élément comme actif lorsque l'URL commence par l'un de ces chemins.
insertBefore	string string[]	yes	Insérer cet élément avant l'élément référencé ici. Pour les tableaux, le premier élément trouvé dans l'ordre est utilisé.
insertAfter	string string[]	yes	Insérer cet élément après l'élément référencé ici. Pour les tableaux, le premier trouvé dans l'ordre est utilisé. insertBefore est prioritaire.
namespaced	boolean	yes	Si true , ajouter /ns/active-namespace à la fin.
prefixNamespaced	boolean	yes	Si true , ajoute /k8s/ns/active-namespace au début de l'article

console.navigation/resource-cluster

Nom	Type de valeur	En option	Description
id	string	non	Un identifiant unique pour cet élément.
model	ExtensionK8sModel	non	Le modèle auquel cet élément de navigation renvoie.
perspective	string	yes	L'identifiant de la perspective à laquelle cet élément appartient. Si elle n'est pas spécifiée, elle contribue à la perspective par défaut.

Nom	Type de valeur	En option	Description
section	string	yes	Section de navigation à laquelle cet élément appartient. S'il n'est pas spécifié, cet élément est rendu comme un lien de premier niveau.
dataAttributes	{ [key: string]: string; }	yes	Ajoute des attributs de données au DOM.
startsWith	string[]	yes	Marquer cet élément comme actif lorsque l'URL commence par l'un de ces chemins.
insertBefore	string string[]	yes	Insérer cet élément avant l'élément référencé ici. Pour les tableaux, le premier élément trouvé dans l'ordre est utilisé.
insertAfter	string string[]	yes	Insérer cet élément après l'élément référencé ici. Pour les tableaux, le premier trouvé dans l'ordre est utilisé. insertBefore est prioritaire.
name	string	yes	Remplace le nom par défaut. S'il n'est pas fourni, le nom du lien sera égal à la valeur plurielle du modèle.

console.navigation/resource-ns

Nom	Type de valeur	En option	Description
id	string	non	Un identifiant unique pour cet élément.
model	ExtensionK8sModel	non	Le modèle auquel cet élément de navigation renvoie.

Nom	Type de valeur	En option	Description
perspective	string	yes	L'identifiant de la perspective à laquelle cet élément appartient. Si elle n'est pas spécifiée, elle contribue à la perspective par défaut.
section	string	yes	Section de navigation à laquelle cet élément appartient. S'il n'est pas spécifié, cet élément est rendu comme un lien de premier niveau.
dataAttributes	{ [key: string]: string; }	yes	Ajoute des attributs de données au DOM.
startsWith	string[]	yes	Marquer cet élément comme actif lorsque l'URL commence par l'un de ces chemins.
insertBefore	string string[]	yes	Insérer cet élément avant l'élément référencé ici. Pour les tableaux, le premier élément trouvé dans l'ordre est utilisé.
insertAfter	string string[]	yes	Insérer cet élément après l'élément référencé ici. Pour les tableaux, le premier trouvé dans l'ordre est utilisé. insertBefore est prioritaire.
name	string	yes	Remplace le nom par défaut. S'il n'est pas fourni, le nom du lien sera égal à la valeur plurielle du modèle.

console.navigation/section

Nom	Type de valeur	En option	Description
id	string	non	Un identifiant unique pour cet élément.
perspective	string	yes	L'identifiant de la perspective à laquelle cet élément appartient. Si elle n'est pas spécifiée, elle contribue à la perspective par défaut.
dataAttributes	{ [key: string]: string; }	yes	Ajoute des attributs de données au DOM.
insertBefore	string string[]	yes	Insérer cet élément avant l'élément référencé ici. Pour les tableaux, le premier élément trouvé dans l'ordre est utilisé.
insertAfter	string string[]	yes	Insérer cet élément après l'élément référencé ici. Pour les tableaux, le premier trouvé dans l'ordre est utilisé. insertBefore est prioritaire.
name	string	yes	Nom de cette section. S'il n'est pas fourni, seul un séparateur sera affiché au-dessus de la section.

console.navigation/separator

Nom	Type de valeur	En option	Description
id	string	non	Un identifiant unique pour cet élément.

Nom	Type de valeur	En option	Description
perspective	string	yes	L'identifiant de la perspective à laquelle cet élément appartient. Si elle n'est pas spécifiée, elle contribue à la perspective par défaut.
section	string	yes	Section de navigation à laquelle cet élément appartient. S'il n'est pas spécifié, cet élément est rendu comme un lien de premier niveau.
dataAttributes	{ [key: string]: string; }	yes	Ajoute des attributs de données au DOM.
insertBefore	string string[]	yes	Insérer cet élément avant l'élément référencé ici. Pour les tableaux, le premier élément trouvé dans l'ordre est utilisé.
insertAfter	string string[]	yes	Insérer cet élément après l'élément référencé ici. Pour les tableaux, le premier trouvé dans l'ordre est utilisé. insertBefore est prioritaire.

console.page/resource/details

Nom	Type de valeur	En option	Description
model	ExtensionK8sGroupKindModel	non	Le modèle pour lequel cette page de ressources renvoie.
component	CodeRef<React.ComponentType<{ match: match<{}>; namespace: string; model: ExtensionK8sModel; }>>	non	Le composant à afficher lorsque l'itinéraire correspond.

console.page/resource/list

Ajoute une nouvelle page de liste de ressources au routeur de la console.

Nom	Type de valeur	En option	Description
model	ExtensionK8sGroup KindModel	non	Le modèle pour lequel cette page de ressources renvoie.
component	CodeRef<React.ComponentType<{ match: match<{}>; namespace: string; model: ExtensionK8sModel; }>>	non	Le composant à afficher lorsque l'itinéraire correspond.

console.page/route

Ajoute une nouvelle page au routeur de la console web. Voir [React Router](#).

Nom	Type de valeur	En option	Description
component	CodeRef<React.ComponentType<RouteComponentProps<{}>, StaticContext, any>>>	non	Le composant à afficher lorsque l'itinéraire correspond.
path	string string[]	non	Chemin d'URL valide ou tableau de chemins d'accès que path-to-regexp@^1.7.0 comprend.
perspective	string	yes	La perspective à laquelle cette page appartient. Si elle n'est pas spécifiée, elle contribue à toutes les perspectives.
exact	boolean	yes	Si la valeur est vraie, la correspondance ne se fera que si le chemin d'accès correspond exactement à location.pathname .

console.page/route/standalone

Ajoute une nouvelle page autonome, rendue en dehors de la mise en page commune, au routeur de la console web. Voir [React Router](#).

Nom	Type de valeur	En option	Description
component	CodeRef<React.ComponentType<RouteComponentProps<{}, StaticContext, any>>>	non	Le composant à afficher lorsque l'itinéraire correspond.
path	string string[]	non	Chemin d'URL valide ou tableau de chemins d'accès que path-to-regexp@^1.7.0 comprend.
exact	boolean	yes	Si la valeur est vraie, la correspondance ne se fera que si le chemin d'accès correspond exactement à location.pathname .

console.perspective

Nom	Type de valeur	En option	Description
id	string	non	L'identificateur de perspective.
name	string	non	Le nom d'affichage de la perspective.
icon	CodeRef<LazyComponent>	non	L'icône d'affichage de la perspective.
landingPageURL	CodeRef<(flags: { [key: string]: boolean; }, isFirstVisit: boolean) ⇒ string>	non	Fonction permettant d'obtenir l'URL de la page d'atterrissage de la perspective.
importRedirectURL	CodeRef<(namespace: string) ⇒ string>	non	La fonction pour obtenir l'URL de redirection pour le flux d'importation.
default	boolean	yes	Si la perspective est la valeur par défaut. Il ne peut y avoir qu'une seule valeur par défaut.

Nom	Type de valeur	En option	Description
defaultPins	ExtensionK8sModel[]	yes	Ressources épinglées par défaut dans la navigation
usePerspectiveDetection	CodeRef<() => [boolean, boolean]>	yes	Le crochet pour détecter la perspective par défaut

console.project-overview/inventory-item

Ajoute un nouvel élément d'inventaire à la page **Project Overview**.

Nom	Type de valeur	En option	Description
component	CodeRef<React.ComponentType<{ projectName: string; }>>	non	Le composant à rendre.

console.project-overview/utilization-item

Ajoute un nouvel élément d'utilisation de la vue d'ensemble du projet.

Nom	Type de valeur	En option	Description
title	string	non	Titre de l'élément d'utilisation.
getUtilizationQuery	CodeRef<GetProject Query>	non	Demande d'utilisation du Prométhée.
humanize	CodeRef<Humanize>	non	Convertir les données Prometheus en format lisible par l'homme.
getTotalQuery	CodeRef<GetProject Query>	yes	Requête totale de Prometheus.
getRequestQuery	CodeRef<GetProject Query>	yes	Requête de demande de Prométhée.
getLimitQuery	CodeRef<GetProject Query>	yes	Demande de limite de Prométhée.

Nom	Type de valeur	En option	Description
TopConsumerPopover	CodeRef<React.ComponentType<TopConsumerPopoverProps>>	yes	Affiche le popover du consommateur supérieur au lieu d'une simple valeur.

console.pvc/alert

Nom	Type de valeur	En option	Description
alert	CodeRef<React.ComponentType< { pvc: K8sResourceComm on; }>>	non	Le composant d'alerte.

console.pvc/create-prop

Nom	Type de valeur	En option	Description
label	string	non	Étiquette pour l'action de création d'un accessoire.
path	string	non	Chemin d'accès à l'action de création d'un accessoire.

console.pvc/delete

Nom	Type de valeur	En option	Description
predicate	CodeRef<(pvc: K8sResourceComm on) ⇒ boolean>	non	Prédicat qui indique si l'extension doit être utilisée ou non.
onPVCKill	CodeRef<(pvc: K8sResourceComm on) ⇒ Promise<void>>	non	Méthode pour l'opération de suppression du PVC.
alert	CodeRef<React.ComponentType< { pvc: K8sResourceComm on; }>>	non	Composant d'alerte pour afficher des informations supplémentaires.

console.pvc/status

Nom	Type de valeur	En option	Description
priority	number	non	Priorité du composant d'état. Une valeur plus élevée signifie une plus grande priorité.
status	CodeRef<React.ComponentType<{ pvc: K8sResourceComm on; }>>	non	L'élément d'état.
predicate	CodeRef<(pvc: K8sResourceComm on) ⇒ boolean>	non	Prédicat qui indique si le composant d'état doit être rendu ou non.

console.redux-reducer

Ajoute un nouveau réducteur au magasin Console Redux qui opère sur **plugins.<scope>** substate.

Nom	Type de valeur	En option	Description
scope	string	non	La clé pour représenter le substate géré par le réducteur dans l'objet d'état Redux.
reducer	CodeRef<Reducer<any, AnyAction>>	non	La fonction du réducteur, qui opère sur le sous-état géré par le réducteur.

console.resource/create

Nom	Type de valeur	En option	Description
model	ExtensionK8sModel	non	Le modèle pour lequel cette page de création de ressources sera rendue.
component	CodeRef<React.ComponentType<Create ResourceComponentProps>>	non	Le composant à rendre lorsque le modèle correspond

console.storage-provider

Nom	Type de valeur	En option	Description
name	string	non	
Component	CodeRef<React.ComponentType<Partial<RouteComponentProps<{}>, StaticContext, any>>>>	non	

console.tab/horizontalNav

Nom	Type de valeur	En option	Description
model	ExtensionK8sKindVersionModel	non	Le modèle pour lequel cet onglet de présentation du fournisseur est utilisé.
page	{ name: string; href: string; }	non	La page à afficher dans l'onglet horizontal. Il prend le nom de l'onglet comme nom et href de l'onglet
component	CodeRef<React.ComponentType<PageComponentProps<K8sResourceCommon>>>	non	Le composant à afficher lorsque l'itinéraire correspond.

console.telemetry/listener

Nom	Type de valeur	En option	Description
listener	CodeRef<TelemetryEventListener>	non	Écouter les événements de télémétrie

console.topology/adapt/build

BuildAdapter contribue à un adaptateur pour adapter l'élément aux données qui peuvent être utilisées par le composant **Build**.

Nom	Type de valeur	En option	Description
adapt	`CodeRef<(element : GraphElement) ⇒ AdapterDataType<BuildConfigData>	undefined`	non

console.topology/adapter/network

NetworkAdapater contribue à un adaptateur pour adapter l'élément aux données qui peuvent être utilisées par le composant **Networking**.

Nom	Type de valeur	En option	Description
adapt	<code>`CodeRef<(element : GraphElement) => NetworkAdapterType`</code>	undefined>`	non

console.topology/adapter/pod

PodAdapter contribue à un adaptateur pour adapter l'élément aux données qui peuvent être utilisées par le composant **Pod**.

Nom	Type de valeur	En option	Description
adapt	<code>`CodeRef<(element : GraphElement) => AdapterDataType<Pods AdapterDataType>`</code>	undefined>`	non

console.topology/component/factory

Getter pour un **ViewComponentFactory**.

Nom	Type de valeur	En option	Description
getFactory	<code>CodeRef<ViewComponentFactory></code>	non	Getter pour un ViewComponentFactory .

console.topology/create/connector

Getter pour la fonction de création de connecteur.

Nom	Type de valeur	En option	Description
getCreateConnector	<code>CodeRef<CreateConnectionGetter></code>	non	Getter pour la fonction de création de connecteur.

console.topology/data/factory

Modèle de données topologiques Extension de l'usine

Nom	Type de valeur	En option	Description
id	string	non	Identifiant unique de l'usine.

Nom	Type de valeur	En option	Description
priority	number	non	Priorité à l'usine
resources	WatchK8sResources Generic	yes	Ressources à récupérer à partir du crochet <code>useK8sWatchResources</code> .
workloadKeys	string[]	yes	Clés dans les ressources contenant des charges de travail.
getDataModel	CodeRef<TopologyDataModelGetter>	yes	Obteneur pour la fabrique de modèles de données.
isResourceDepicted	CodeRef<TopologyDataModelDepicted>	yes	Getter pour la fonction permettant de déterminer si une ressource est représentée par cette fabrique de modèles.
getDataModelReconciler	CodeRef<TopologyDataModelReconciler>	yes	Getter pour la fonction de réconciliation du modèle de données après le chargement des modèles de toutes les extensions.

console.topology/decorator/provider

Extension du fournisseur de décorateur de topologie

Nom	Type de valeur	En option	Description
id	string	non	
priority	number	non	
quadrant	TopologyQuadrant	non	
decorator	CodeRef<TopologyDecoratorGetter>	non	

console.topology/details/resource-alert

DetailsResourceAlert génère une alerte pour un contexte topologique ou un élément graphique spécifique.

Nom	Type de valeur	En option	Description
id	string	non	L'identifiant de cette alerte. Utilisé pour sauvegarder l'état si l'alerte ne doit pas être affichée après avoir été supprimée.
contentProvider	<code>`CodeRef<(element : GraphElement) => DetailsResourceAlertContent`</code>	non	non

console.topology/details/resource-link

DetailsResourceLink apporte un lien pour un contexte topologique ou un élément de graphe spécifique.

Nom	Type de valeur	En option	Description
link	<code>`CodeRef<(element : GraphElement) => React.Component`</code>	non	non
Renvoie le lien de la ressource s'il est fourni, sinon il n'est pas défini. Utilisez les propriétés ResourceIcon et ResourceLink pour les styles.	priority	number	yes

console.topology/details/tab

DetailsTab crée un onglet pour le panneau des détails de la topologie.

Nom	Type de valeur	En option	Description
id	string	non	Un identifiant unique pour cet onglet de détails.
label	string	non	L'étiquette de l'onglet à afficher dans l'interface utilisateur.

Nom	Type de valeur	En option	Description
insertBefore	string string[]	yes	Insérer cet élément avant l'élément référencé ici. Pour les tableaux, le premier élément trouvé dans l'ordre est utilisé.
insertAfter	string string[]	yes	Insérer cet élément après l'élément référencé ici. Pour les tableaux, le premier élément trouvé dans l'ordre est utilisé. La valeur insertBefore est prioritaire.

console.topology/details/tab-section

DetailsTabSection contribue à la création d'une section pour un onglet spécifique dans le panneau des détails de la topologie.

Nom	Type de valeur	En option	Description
id	string	non	Identifiant unique pour cette section de l'onglet Détails.
tab	string	non	L'identifiant de l'onglet parent auquel cette section doit contribuer.
provider	CodeRef<DetailsTabSectionExtensionHook>	non	Composant SDK : <code><Section title=\{\}\>...</code> zone matelassée
section	<code>`CodeRef<(element : GraphElement, renderNull? : () => null) => React.Component`</code>	undefined`	non
@deprecated Fallback si aucun fournisseur n'est défini. renderNull est déjà un no-op.	insertBefore	string string[]	yes

Nom	Type de valeur	En option	Description
Pour les tableaux, le premier élément trouvé dans l'ordre est utilisé.	insertAfter	string string[]	yes

console.topology/display/filters

Extension des filtres d'affichage de la topologie

Nom	Type de valeur	En option	Description
getTopologyFilters	CodeRef<() ⇒ TopologyDisplayOption[]>	non	
applyDisplayOptions	CodeRef<TopologyApplyDisplayOptions>	non	

console.topology/relationship/provider

Extension du connecteur du fournisseur de relations topologiques

Nom	Type de valeur	En option	Description
provides	CodeRef<RelationshipProviderProvides>	non	
tooltip	string	non	
create	CodeRef<RelationshipProviderCreate>	non	
priority	number	non	

console.user-preference/group

Nom	Type de valeur	En option	Description
id	string	non	ID utilisé pour identifier le groupe de préférences des utilisateurs.
label	string	non	L'étiquette du groupe de préférences de l'utilisateur

Nom	Type de valeur	En option	Description
insertBefore	string	yes	ID du groupe de préférences de l'utilisateur devant lequel ce groupe doit être placé
insertAfter	string	yes	ID du groupe de préférences de l'utilisateur après lequel ce groupe doit être placé

console.user-preference/item

Nom	Type de valeur	En option	Description
id	string	non	ID utilisé pour identifier l'élément de préférence de l'utilisateur et référencé dans insertAfter et insertBefore pour définir l'ordre des éléments.
label	string	non	L'étiquette de la préférence de l'utilisateur
description	string	non	La description de la préférence de l'utilisateur.
field	UserPreferenceField	non	Les options du champ de saisie utilisées pour rendre les valeurs afin de définir la préférence de l'utilisateur.
groupId	string	yes	ID utilisés pour identifier les groupes de préférences des utilisateurs auxquels l'élément appartient.

Nom	Type de valeur	En option	Description
insertBefore	string	yes	ID de l'élément de préférence de l'utilisateur devant lequel cet élément doit être placé
insertAfter	string	yes	ID de l'élément de préférence de l'utilisateur après lequel cet élément doit être placé

console.yaml-template

Modèles YAML pour l'édition des ressources via l'éditeur YAML.

Nom	Type de valeur	En option	Description
model	ExtensionK8sModel	non	Modèle associé au modèle.
template	CodeRef<string>	non	Le modèle YAML.
name	string	non	Le nom du modèle. Utilisez le nom default pour indiquer qu'il s'agit du modèle par défaut.

dev-console.add/action

Nom	Type de valeur	En option	Description
id	string	non	ID utilisé pour identifier l'action.
label	string	non	Le libellé de l'action
description	string	non	La description de l'action.
href	string	non	Le href vers lequel naviguer.
groupId	string	yes	ID utilisés pour identifier les groupes d'action auxquels l'action doit appartenir.

Nom	Type de valeur	En option	Description
icon	CodeRef<React.ReactNode>	yes	L'icône d'affichage de la perspective.
accessReview	AccessReviewResourceAttributes[]	yes	Contrôle d'accès facultatif pour contrôler la visibilité ou l'activation de l'action.

dev-console.add/action-group

Nom	Type de valeur	En option	Description
id	string	non	ID utilisé pour identifier le groupe d'action.
name	string	non	Le titre du groupe d'action
insertBefore	string	yes	ID du groupe d'action devant lequel ce groupe doit être placé
insertAfter	string	yes	ID du groupe d'action après lequel ce groupe doit être placé

dev-console.import/environment

Nom	Type de valeur	En option	Description
imageStreamName	string	non	Nom du flux d'images pour lequel des variables d'environnement personnalisées doivent être fournies
imageStreamTags	string[]	non	Liste des balises de flux d'images prises en charge
environments	ImageEnvironment[]	non	Liste des variables d'environnement

console.page/resource/tab

Déclassé. Utilisez plutôt **console.tab/horizontalNav**. Ajoute un nouvel onglet de ressources au routeur de la console.

Nom	Type de valeur	En option	Description
model	ExtensionK8sGroupKindModel	non	Le modèle pour lequel cette page de ressources renvoie.
component	CodeRef<React.ComponentType<RouteComponentProps<{}>, StaticContext, any>>>	non	Le composant à afficher lorsque l'itinéraire correspond.
name	string	non	Le nom de l'onglet.
href	string	yes	L'adresse facultative du lien de l'onglet. S'il n'est pas fourni, le premier path est utilisé.
exact	boolean	yes	Si la valeur est vraie, la correspondance ne se fera que si le chemin d'accès correspond exactement à location.pathname .

7.5.2. API de la console OpenShift Container Platform

useActivePerspective

Crochet qui fournit la perspective actuellement active et un rappel pour définir la perspective active. Il renvoie un tuple contenant la perspective active actuelle et le callback du setter.

```
const Component: React.FC = (props) => {
  const [activePerspective, setActivePerspective] = useActivePerspective();
  return <select
    value={activePerspective}
    onChange={(e) => setActivePerspective(e.target.value)}
  >
    {
      // ...perspective options
    }
  </select>
}
```

GreenCheckCircleIcon

Composant permettant d'afficher l'icône d'un cercle vert coché.

```
<CheckCircleIcon vert title="Healthy" />
```


Nom du paramètre	Description
className	(facultatif) nom de classe supplémentaire pour le composant
title	(facultatif) titre de l'icône
size	(facultatif) taille de l'icône : ("sm", "md", "lg", "xl")

RedExclamationCircleIcon

Composant permettant d'afficher une icône de cercle avec un point d'exclamation rouge.

```
<RedExclamationCircleIcon title="Failed" />
```

Nom du paramètre	Description
className	(facultatif) nom de classe supplémentaire pour le composant
title	(facultatif) titre de l'icône
size	(facultatif) taille de l'icône : ("sm", "md", "lg", "xl")

YellowExclamationTriangleIcon

Composant permettant d'afficher une icône d'exclamation en forme de triangle jaune.

```
<YellowExclamationTriangleIcon title="Warning" />
```

Nom du paramètre	Description
className	(facultatif) nom de classe supplémentaire pour le composant
title	(facultatif) titre de l'icône
size	(facultatif) taille de l'icône : ("sm", "md", "lg", "xl")

BlueInfoCircleIcon

Composant permettant d'afficher une icône bleue en forme de cercle d'information.

```
<BlueInfoCircleIcon title="Info" />
```

Nom du paramètre	Description
------------------	-------------

Nom du paramètre	Description
className	(facultatif) nom de classe supplémentaire pour le composant
title	(facultatif) titre de l'icône
size	(facultatif) taille de l'icône : ("sm", "md", "lg", "xl")

ErrorStatus

Composant permettant d'afficher une fenêtre contextuelle sur l'état de l'erreur.

```
<ErrorStatus title={errorMsg} />
```

Nom du paramètre	Description
title	(facultatif) texte d'état
iconOnly	(optionnel) si vrai, affiche uniquement l'icône
noTooltip	(facultatif) si vrai, l'infobulle ne sera pas affichée
className	(facultatif) nom de classe supplémentaire pour le composant
popoverTitle	(facultatif) titre du popover

InfoStatus

Composant permettant d'afficher une fenêtre contextuelle sur l'état des informations.

```
<InfoStatus title={infoMsg} />
```

Nom du paramètre	Description
title	(facultatif) texte d'état
iconOnly	(optionnel) si vrai, affiche uniquement l'icône
noTooltip	(facultatif) si vrai, l'infobulle ne sera pas affichée
className	(facultatif) nom de classe supplémentaire pour le composant
popoverTitle	(facultatif) titre du popover

ProgressStatus

Composant permettant d'afficher une fenêtre contextuelle sur l'état d'avancement des travaux.

```
<ProgressStatus title={progressMsg} />
```

Nom du paramètre	Description
title	(facultatif) texte d'état
iconOnly	(optionnel) si vrai, affiche uniquement l'icône
noTooltip	(facultatif) si vrai, l'infobulle ne sera pas affichée
className	(facultatif) nom de classe supplémentaire pour le composant
popoverTitle	(facultatif) titre du popover

SuccessStatus

Composant permettant d'afficher une fenêtre contextuelle sur l'état de la réussite.

```
<SuccessStatus title={successMsg} />
```

Nom du paramètre	Description
title	(facultatif) texte d'état
iconOnly	(optionnel) si vrai, affiche uniquement l'icône
noTooltip	(facultatif) si vrai, l'infobulle ne sera pas affichée
className	(facultatif) nom de classe supplémentaire pour le composant
popoverTitle	(facultatif) titre du popover

checkAccess

Fournit des informations sur l'accès des utilisateurs à une ressource donnée. Il renvoie un objet contenant des informations sur l'accès à la ressource.

Nom du paramètre	Description
resourceAttributes	attributs des ressources pour l'examen de l'accès
impersonate	détails de l'usurpation d'identité

useAccessReview

Crochet qui fournit des informations sur l'accès des utilisateurs à une ressource donnée. Il renvoie un tableau contenant les valeurs **isAllowed** et **loading**.

Nom du paramètre	Description
resourceAttributes	attributs des ressources pour l'examen de l'accès
impersonate	détails de l'usurpation d'identité

useResolvedExtensions

Crochet React pour consommer les extensions Console avec les propriétés **CodeRef** résolues. Ce hook accepte le(s) même(s) argument(s) que le hook **useExtensions** et renvoie une liste adaptée d'instances d'extensions, en résolvant toutes les références de code dans les propriétés de chaque extension. Initialement, le hook renvoie un tableau vide. Une fois la résolution terminée, le composant React est re-rendu avec le hook renvoyant une liste adaptée d'extensions. Lorsque la liste des extensions correspondantes change, la résolution est relancée. Le hook continuera à renvoyer le résultat précédent jusqu'à ce que la résolution soit terminée. Les éléments de résultat du crochet sont garantis pour être référentiellement stables à travers les re-renders. Il renvoie un tuple contenant une liste d'instances d'extensions adaptées avec des références de code résolues, un drapeau booléen indiquant si la résolution est complète, et une liste d'erreurs détectées au cours de la résolution.

```
const [navItemExtensions, navItemsResolved] = useResolvedExtensions<NavItem>(isNavItem);
// process adapted extensions and render your component
```

Nom du paramètre	Description
typeGuards	Une liste de callbacks qui acceptent chacun une extension de plugin dynamique comme argument et renvoient un drapeau booléen indiquant si l'extension répond ou non aux contraintes de type souhaitées

HorizontalNav

Un composant qui crée une barre de navigation pour une page - Le routage est géré en tant que partie du composant - **console.tab/horizontalNav** peut être utilisé pour ajouter du contenu supplémentaire à n'importe quelle barre de navigation horizontale.

```
const HomePage: React.FC = (props) => {
  const page = {
    href: '/home',
    name: 'Home',
    component: () => <>Home</>
  }
  return <HorizontalNav match={props.match} pages={[page]} />
}
```

Nom du paramètre	Description
------------------	-------------

Nom du paramètre	Description
resource	La ressource associée à cette navigation, un objet de type <code>K8sResourceCommon</code>
pages	Un tableau d'objets de page
match	objet de correspondance fourni par React Router

VirtualizedTable

Composant permettant de réaliser des tableaux virtualisés.

```
const MachineList: React.FC<MachineListProps> = (props) => {
  return (
    <VirtualizedTable<MachineKind>
      {...props}
      aria-label='Machines'
      columns={getMachineColumns}
      Row={getMachineTableRow}
    />
  );
}
```

Nom du paramètre	Description
data	données pour le tableau
loaded	drapeau indiquant que les données sont chargées
loadError	objet d'erreur en cas de problème lors du chargement des données
columns	configuration de la colonne
Row	configuration des rangs
unfilteredData	données originales sans filtre
NoDataEmptyMsg	(facultatif) pas de données composant de message vide
EmptyMsg	(facultatif) composant de message vide
scrollNode	(facultatif) pour gérer le défilement
label	(facultatif) étiquette pour le tableau
ariaLabel	(facultatif) étiquette aria

Nom du paramètre	Description
gridBreakPoint	dimensionnement du découpage de la grille pour la réactivité
onSelect	fonction (optionnelle) pour la gestion de la sélection de la table
rowData	(facultatif) données spécifiques à la ligne

TableData

Composant permettant d'afficher les données d'un tableau à l'intérieur d'une ligne du tableau.

```
const PodRow: React.FC<RowProps<K8sResourceCommon>> = ({ obj, activeColumnIDs }) => {
  return (
    <>
      <TableData id={columns[0].id} activeColumnIDs={activeColumnIDs}>
        <ResourceLink kind="Pod" name={obj.metadata.name} namespace={obj.metadata.namespace} />
      </TableData>
      <TableData id={columns[1].id} activeColumnIDs={activeColumnIDs}>
        <ResourceLink kind="Namespace" name={obj.metadata.namespace} />
      </TableData>
    </>
  );
};
```

Nom du paramètre	Description
id	identifiant unique pour la table
activeColumnIDs	colonnes actives
className	(optionnel) nom de la classe d'option pour le style

useActiveColumns

Un crochet qui fournit une liste de TableColumn actives sélectionnées par l'utilisateur.

```
// See implementation for more details on TableColumn type
const [activeColumns, userSettingsLoaded] = useActiveColumns({
  columns,
  showNamespaceOverride: false,
  columnManagementID,
});
return userSettingsAreLoaded ? <VirtualizedTable columns= {activeColumns} {...otherProps} /> :
null
```

Nom du paramètre	Description
options	Qui sont transmises sous la forme d'une carte clé-valeur
<code>options.columns</code>	<code>{TableColumn[]}</code> options.columns - Un tableau de toutes les colonnes de table disponibles
<code>options.showNamespaceOverride</code>	<code>{boolean}</code> [options.showNamespaceOverride] - (facultatif) Si true, une colonne d'espace de noms sera incluse, indépendamment des sélections de gestion des colonnes
<code>options.columnManagementID</code>	<code>{string}</code> [options.columnManagementID] - (facultatif) Un identifiant unique utilisé pour persister et récupérer les sélections de gestion de colonnes vers et depuis les paramètres utilisateur. Il s'agit généralement d'une chaîne "groupverionkind" pour une ressource.

Un tuple contenant les colonnes actives sélectionnées par l'utilisateur actuel (un sous-ensemble de options.columns), et un drapeau booléen indiquant si les paramètres de l'utilisateur ont été chargés.

ListPageHeader

Composant permettant de générer un en-tête de page.

```
const exampleList: React.FC = () => {
  return (
    <>
      <ListPageHeader title="Example List Page"/>
    </>
  );
};
```

Nom du paramètre	Description
title	titre de la rubrique
helpText	(optionnel) section d'aide comme react node
badge	(facultatif) icône de badge comme react node

ListPageCreate

Composant permettant d'ajouter un bouton de création pour un type de ressource spécifique qui génère automatiquement un lien vers le YAML de création de cette ressource.

```
const exampleList: React.FC<MyProps> = () => {
  return (
    <>
```

```

    <ListPageHeader title="Example Pod List Page"/>
    <ListPageCreate groupVersionKind="Pod">Create Pod</ListPageCreate>
  </ListPageHeader>
</>
);
};

```

Nom du paramètre	Description
groupVersionKind	le groupe/la version/la nature de la ressource à représenter

ListPageCreateLink

Composant permettant de créer un lien stylisé.

```

const exampleList: React.FC<MyProps> = () => {
  return (
    <>
      <ListPageHeader title="Example Pod List Page"/>
      <ListPageCreateLink to={'/link/to/my/page'}>Create Item</ListPageCreateLink>
    </ListPageHeader>
    </>
  );
};

```

Nom du paramètre	Description
to	chaîne de caractères emplacement où le lien doit être dirigé
createAccessReview	(facultatif) objet avec l'espace de noms et le type utilisés pour déterminer l'accès
children	(facultatif) enfants du composant

ListPageCreateButton

Composant permettant de créer un bouton.

```

const exampleList: React.FC<MyProps> = () => {
  return (
    <>
      <ListPageHeader title="Example Pod List Page"/>
      <ListPageCreateButton createAccessReview={access}>Create Pod</ListPageCreateButton>
    </ListPageHeader>
    </>
  );
};

```


Nom du paramètre	Description
createAccessReview	(facultatif) objet avec l'espace de noms et le type utilisés pour déterminer l'accès
pfButtonProps	(facultatif) Accessoires du bouton Patternfly

ListPageCreateDropdown

Composant permettant de créer une liste déroulante avec contrôle des autorisations.

```
const exampleList: React.FC<MyProps> = () => {
  const items = {
    SAVE: 'Save',
    DELETE: 'Delete',
  }
  return (
    <>
      <ListPageHeader title="Example Pod List Page"/>
      <ListPageCreateDropdown createAccessReview={access} items=
{items}>Actions</ListPageCreateDropdown>
      </ListPageHeader>
    </>
  );
};
```

Nom du paramètre	Description
items	key:ReactNode paires d'éléments à afficher dans le composant déroulant
onClick	fonction de rappel pour les clics sur les éléments de la liste déroulante
createAccessReview	(facultatif) objet avec l'espace de noms et le type utilisés pour déterminer l'accès
children	(facultatif) enfants pour le menu déroulant à bascule

ListPageFilter

Composant qui génère un filtre pour la page de liste.

```
// See implementation for more details on RowFilter and FilterValue types
const [staticData, filteredData, onFilterChange] = useListPageFilter(
  data,
  rowFilters,
  staticFilters,
);
// ListPageFilter updates filter state based on user interaction and resulting filtered data can be
rendered in an independent component.
return (
```

```

<>
<ListPageHeader .../>
<ListPagBody>
  <ListPageFilter data={staticData} onFilterChange={onFilterChange} />
  <List data={filteredData} />
</ListPageBody>
</>
)

```

Nom du paramètre	Description
data	Un tableau de points de données
loaded	indique que les données ont été chargées
onFilterChange	fonction de rappel pour la mise à jour du filtre
rowFilters	(facultatif) Un tableau d'éléments RowFilter définissant les options de filtrage disponibles
nameFilterPlaceholder	(facultatif) caractère générique pour le filtre de nom
labelFilterPlaceholder	(facultatif) caractère générique pour le filtre de l'étiquette
hideLabelFilter	(optionnel) affiche uniquement le filtre de nom au lieu des filtres de nom et d'étiquette
hideNameLabelFilter	(facultatif) masque les filtres de nom et d'étiquette
columnLayout	(facultatif) objet de mise en page de la colonne
hideColumnManagement	(optionnel) pour masquer la gestion de la colonne

useListPageFilter

Un crochet qui gère l'état des filtres pour le composant ListPageFilter. Il renvoie un tuple contenant les données filtrées par tous les filtres statiques, les données filtrées par tous les filtres statiques et les filtres de ligne, ainsi qu'un rappel qui met à jour les filtres de ligne.

```

// See implementation for more details on RowFilter and FilterValue types
const [staticData, filteredData, onFilterChange] = useListPageFilter(
  data,
  rowFilters,
  staticFilters,
);
// ListPageFilter updates filter state based on user interaction and resulting filtered data can be
// rendered in an independent component.
return (
  <>
    <ListPageHeader .../>
    <ListPagBody>

```

```

    <ListPageFilter data={staticData} onFilterChange={onFilterChange} />
    <List data={filteredData} />
  </ListPageBody>
</>
)

```

Nom du paramètre	Description
data	Un tableau de points de données
rowFilters	(facultatif) Un tableau d'éléments RowFilter définissant les options de filtrage disponibles
staticFilters	(facultatif) Un tableau d'éléments FilterValue qui sont appliqués de manière statique aux données

ResourceLink

Composant qui crée un lien vers un type de ressource spécifique avec un badge en forme d'icône.

```

<ResourceLink
  kind="Pod"
  name="testPod"
  title={metadata.uid}
/>

```

Nom du paramètre	Description
kind	(facultatif) le type de ressource, c'est-à-dire Pod, Deployment, Namespace
groupVersionKind	(optionnel) objet avec group, version et type
className	(facultatif) style de classe pour le composant
displayName	(facultatif) nom d'affichage du composant, remplace le nom de la ressource s'il est défini
inline	(optionnel) drapeau pour créer un badge d'icône et un nom en ligne avec les enfants
linkTo	(optionnel) drapeau pour créer un objet Link - valeur par défaut : true
name	(facultatif) nom de la ressource
namespace	(facultatif) espace de noms spécifique pour le type de ressource à lier

Nom du paramètre	Description
hideIcon	(optionnel) drapeau pour cacher l'icône du badge
title	(facultatif) titre de l'objet lien (non affiché)
dataTest	(facultatif) identifiant pour les tests
onClick	(optionnel) fonction de rappel lorsque le composant est cliqué
truncate	(optionnel) drapeau pour tronquer le lien s'il est trop long

ResourceIcon

Composant qui crée une icône de badge pour un type de ressource spécifique.

```
<ResourceIcon kind="Pod"/>
```

Nom du paramètre	Description
kind	(facultatif) le type de ressource, c'est-à-dire Pod, Deployment, Namespace
groupVersionKind	(facultatif) objet avec groupe, version et type
className	(facultatif) style de classe pour le composant

useK8sModel

Crochet qui récupère le modèle k8s pour le type K8sGroupVersionKind fourni par redux. Il renvoie un tableau dont le premier élément est le modèle k8s et le deuxième élément est le statut **inFlight**.

```
const Component: React.FC = () => {
  const [model, inFlight] = useK8sModel({ group: 'app'; version: 'v1'; kind: 'Deployment' });
  return ...
}
```

Nom du paramètre	Description
groupVersionKind	group, version, kind of k8s resource \N{@link K8sGroupVersionKind} est préférable ; il est également possible de passer une référence pour group, version, kind, qui est obsolète, c'est-à-dire groupversionkind \{@link K8sResourceKindReference}.

useK8sModels

Hook qui récupère tous les modèles k8s actuels de redux. Il renvoie un tableau dont le premier élément est la liste des modèles k8s et le deuxième élément est le statut **inFlight**.

```
const Component: React.FC = () => {
  const [models, inFlight] = UseK8sModels();
  return ...
}
```

useK8sWatchResource

Crochet qui récupère la ressource k8s avec l'état de chargement et d'erreur. Il renvoie un tableau dont le premier élément est la (les) ressource(s), le deuxième l'état de chargement et le troisième l'état d'erreur, le cas échéant.

```
const Component: React.FC = () => {
  const watchRes = {
    ...
  }
  const [data, loaded, error] = useK8sWatchResource(watchRes)
  return ...
}
```

Nom du paramètre	Description
initResource	les options nécessaires pour surveiller les ressources.

useK8sWatchResources

Hook qui récupère les ressources k8s avec leur statut respectif (loaded et error). Il renvoie une carte dont les clés sont celles fournies dans initResources et la valeur a trois propriétés : data, loaded et error.

```
const Component: React.FC = () => {
  const watchResources = {
    'deployment': {...},
    'pod': {...}
    ...
  }
  const {deployment, pod} = useK8sWatchResources(watchResources)
  return ...
}
```

Nom du paramètre	Description
initResources	les ressources doivent être surveillées sous la forme d'une paire clé-valeur, dans laquelle la clé est unique pour la ressource et la valeur correspond aux options nécessaires pour surveiller la ressource en question.

consoleFetch

Une enveloppe personnalisée autour de **fetch** qui ajoute des en-têtes spécifiques à la console et permet des tentatives et des délais d'attente. Elle valide également le code d'état de la réponse et lance l'erreur appropriée ou déconnecte l'utilisateur si nécessaire. Il renvoie une promesse qui résout la réponse.

Nom du paramètre	Description
url	L'URL à rechercher
options	Les options à transmettre à fetch
timeout	Le délai d'attente en millisecondes

consoleFetchJSON

Une enveloppe personnalisée autour de **fetch** qui ajoute des en-têtes spécifiques à la console et permet des tentatives et des délais d'attente. Il valide également le code d'état de la réponse et lance l'erreur appropriée ou déconnecte l'utilisateur si nécessaire. Il renvoie la réponse sous la forme d'un objet JSON. Il utilise **consoleFetch** en interne. Il retourne une promesse qui résout la réponse en tant qu'objet JSON.

Nom du paramètre	Description
url	L'URL à rechercher
method	La méthode HTTP à utiliser. La valeur par défaut est GET
options	Les options à transmettre à fetch
timeout	Le délai d'attente en millisecondes
cluster	Le nom de la grappe à laquelle adresser la demande. Par défaut, il s'agit de la grappe active sélectionnée par l'utilisateur

consoleFetchText

Une enveloppe personnalisée autour de **fetch** qui ajoute des en-têtes spécifiques à la console et permet des tentatives et des délais d'attente. Il valide également le code d'état de la réponse et lance l'erreur appropriée ou déconnecte l'utilisateur si nécessaire. Il renvoie la réponse sous forme de texte. Elle utilise **consoleFetch** en interne. Il retourne une promesse qui résout la réponse en tant que texte.

Nom du paramètre	Description
url	L'URL à rechercher
options	Les options à transmettre à fetch
timeout	Le délai d'attente en millisecondes
cluster	Le nom de la grappe à laquelle adresser la demande. Par défaut, il s'agit de la grappe active sélectionnée par l'utilisateur

getConsoleRequestHeaders

Une fonction qui crée des en-têtes relatifs à l'impersonnalisation et au multicluster pour les demandes d'API en utilisant l'état actuel de redux. Elle renvoie un objet contenant les en-têtes de demande d'impersonnalisation et de multicluster appropriés, en fonction de l'état de redux.

Nom du paramètre	Description
targetCluster	remplacer le cluster actif actuel par le targetCluster fourni

k8sGetResource

Il récupère une ressource dans le cluster, en fonction des options fournies. Si le nom est fourni, elle renvoie une ressource, sinon elle renvoie toutes les ressources correspondant au modèle. Il renvoie une promesse qui correspond à la réponse sous forme d'objet JSON avec une ressource si le nom est fourni, sinon il renvoie toutes les ressources correspondant au modèle. En cas d'échec, la promesse est rejetée avec une réponse d'erreur HTTP.

Nom du paramètre	Description
options	Qui sont transmises sous forme de paires clé-valeur dans la carte
``	options.model - modèle k8s
``	options.name - Le nom de la ressource, s'il n'est pas fourni, il recherchera toutes les ressources correspondant au modèle.
``	options.ns - L'espace de noms à examiner, ne doit pas être spécifié pour les ressources à l'échelle d'un cluster.
``	options.path - Ajoute un sous-chemin s'il est fourni
``	options.queryParams - Les paramètres de requête à inclure dans l'URL.
``	options.requestInit - L'objet d'initialisation de la recherche à utiliser. Il peut contenir des en-têtes de requête, une méthode, une redirection, etc. En savoir plus (lien : https://microsoft.github.io/PowerBI-JavaScript/interfaces/node_modules_typedoc_node_modules_typescript_lib_lib_dom_d.requestinit.html)

k8sCreateResource

Il crée une ressource dans le cluster, sur la base des options fournies. Elle renvoie une promesse qui correspond à la réponse de la ressource créée. En cas d'échec, la promesse est rejetée avec une réponse d'erreur HTTP.

Nom du paramètre	Description
options	Qui sont transmises sous forme de paires clé-valeur dans la carte
``	options.model - modèle k8s
``	options.data - données utiles pour la ressource à créer
``	options.path - Ajoute un sous-chemin s'il est fourni
``	options.queryParams - Les paramètres de requête à inclure dans l'URL.

k8sUpdateResource

Il met à jour l'ensemble de la ressource dans le cluster, en fonction des options fournies. Lorsqu'un client doit remplacer entièrement une ressource existante, il peut utiliser `k8sUpdate`. Il peut également utiliser `k8sPatch` pour effectuer une mise à jour partielle. Il renvoie une promesse qui se résout en réponse à la ressource mise à jour. En cas d'échec, la promesse est rejetée avec une réponse d'erreur HTTP.

Nom du paramètre	Description
options	qui sont transmises sous forme de paires clé-valeur dans la carte
``	options.model - modèle k8s
``	options.data - charge utile pour la ressource k8s à mettre à jour
``	options.ns - espace de noms à examiner, il ne doit pas être spécifié pour les ressources à l'échelle d'un cluster.
``	options.name - nom de la ressource à mettre à jour.
``	options.path - Ajoute un sous-chemin s'il est fourni
``	options.queryParams - Les paramètres de requête à inclure dans l'URL.

k8sPatchResource

Il patche n'importe quelle ressource dans le cluster, sur la base des options fournies. Lorsqu'un client a besoin d'effectuer une mise à jour partielle, il peut utiliser `k8sPatch`. Il peut également utiliser `k8sUpdate` pour remplacer entièrement une ressource existante. En savoir plus [Data Tracker](#). Il renvoie une promesse qui se résout à la réponse de la ressource patchée. En cas d'échec, la promesse est rejetée avec une réponse d'erreur HTTP.

Nom du paramètre	Description
options	Qui sont transmises sous forme de paires clé-valeur dans la carte.
` `	options.model - modèle k8s
` `	options.resource - La ressource à corriger.
` `	options.data - Uniquement les données à patcher sur la ressource existante avec l'opération, le chemin et la valeur.
` `	options.path - Ajoute un sous-chemin si celui-ci est fourni.
` `	options.queryParams - Les paramètres de requête à inclure dans l'URL.

k8sDeleteResource

Il supprime les ressources du cluster, en se basant sur le modèle et la ressource fournis. Le ramassage des ordures fonctionne sur la base de 'Foreground'|'Background', peut être configuré avec la propriété propagationPolicy dans le modèle fourni ou passé en json. Il renvoie une promesse qui se résout en une réponse de type Status. En cas d'échec, la promesse est rejetée avec une réponse d'erreur HTTP.

```
{ kind : 'DeleteOptions', apiVersion : 'v1', propagationPolicy }
```

Nom du paramètre	Description
options	qui sont transmises sous la forme d'une paire clé-valeur dans la carte.
` `	options.model - modèle k8s
` `	options.resource - La ressource à supprimer.
` `	options.path - Ajoute un sous-chemin s'il est fourni
` `	options.queryParams - Les paramètres de requête à inclure dans l'URL.
` `	options.requestInit - L'objet d'initialisation de la recherche à utiliser. Il peut contenir des en-têtes de requête, une méthode, une redirection, etc. En savoir plus \N{@link https://microsoft.github.io/PowerBI-JavaScript/interfaces/node_modules_typedoc_node_modules_typescript_lib_lib_dom_d.requestinit.html } }

Nom du paramètre	Description
``	options.json - Peut contrôler explicitement le ramassage des ordures des ressources s'il est fourni, sinon le modèle sera appliqué par défaut (propagationPolicy).

k8sListResource

Liste les ressources sous forme de tableau dans le cluster, en fonction des options fournies. Elle renvoie une promesse qui se résout en réponse.

Nom du paramètre	Description
options	Qui sont transmises sous forme de paires clé-valeur dans la carte
``	options.model - modèle k8s
``	options.queryParams - Les paramètres de requête à inclure dans l'URL et peuvent également passer des sélecteurs d'étiquettes avec la clé "labelSelector".
``	options.requestInit - L'objet d'initialisation de la recherche à utiliser. Il peut contenir des en-têtes de requête, une méthode, une redirection, etc. En savoir plus \N{@link https://microsoft.github.io/PowerBI-JavaScript/interfaces/node_modules_typedoc_node_modules_typescript_lib_lib_dom_d.requestinit.html } }

k8sListResourceItems

Même interface que \N{@link k8sListResource} mais renvoie les sous-éléments. Elle renvoie la version de l'api pour le modèle, c'est-à-dire **group/version**.

getAPIVersionForModel

Fournit la version de l'api pour un modèle k8s.

Nom du paramètre	Description
model	modèle k8s

getGroupVersionKindForResource

Fournit un groupe, une version et un type de ressource. Il renvoie le groupe, la version et le type de la ressource fournie. Si la ressource n'a pas de groupe API, le groupe "core" sera renvoyé. Si la ressource a une version API invalide, une erreur sera levée.

Nom du paramètre	Description
resource	ressource k8s

getGroupVersionKindForModel

Fournit un groupe, une version et un type pour un modèle k8s. Cette commande renvoie le groupe, la version et le type du modèle fourni. Si le modèle n'a pas d'apiGroup, le groupe "core" sera renvoyé.

Nom du paramètre	Description
model	modèle k8s

StatusPopupSection

Composant qui affiche l'état dans une fenêtre contextuelle. Composant utile pour construire des extensions **console.dashboards/overview/health/resource**.

```
<StatusPopupSection
  firstColumn={
    <>
      <span>{title}</span>
      <span className="text-secondary">
        My Example Item
      </span>
    </>
  }
  secondColumn='Status'
>
```

Nom du paramètre	Description
firstColumn	valeurs de la première colonne de la fenêtre contextuelle
secondColumn	(facultatif) valeurs pour la deuxième colonne de la fenêtre contextuelle
children	(optionnel) enfants pour le popup

StatusPopupItem

Élément d'état utilisé dans la fenêtre contextuelle d'état ; utilisé dans **StatusPopupSection**.

```
<StatusPopupSection
  firstColumn='Example'
  secondColumn='Status'
>
  <StatusPopupItem icon={healthStateMapping[MCGMetrics.state]?.icon}>
    Complete
  </StatusPopupItem>
```

```

    <StatusPopuItem icon={healthStateMapping[RGWMetrics.state]?.icon}>
      Pending
    </StatusPopuItem>
  </StatusPopupSection>

```

Nom du paramètre	Description
value	(facultatif) valeur du texte à afficher
icon	(facultatif) icône à afficher
children	éléments enfants

Overview

Crée un composant enveloppant pour un tableau de bord.

```

<Overview>
  <OverviewGrid mainCards={mainCards} leftCards={leftCards} rightCards={rightCards} />
</Overview>

```

Nom du paramètre	Description
className	(optionnel) classe de style pour le div
children	(facultatif) éléments du tableau de bord

OverviewGrid

Crée une grille d'éléments de cartes pour un tableau de bord ; utilisé dans **Overview**.

```

<Overview>
  <OverviewGrid mainCards={mainCards} leftCards={leftCards} rightCards={rightCards} />
</Overview>

```

Nom du paramètre	Description
mainCards	cartes pour la grille
leftCards	(facultatif) cartes pour le côté gauche de la grille
rightCards	(facultatif) cartes pour le côté droit de la grille

InventoryItem

Crée un élément de carte d'inventaire.

```

return (
  <InventoryItem>
    <InventoryItemTitle>{title}</InventoryItemTitle>

```

```

    <InventoryItemBody error={loadError}>
      {loaded && <InventoryItemStatus count={workerNodes.length} icon={<MonitoringIcon />} />}
    </InventoryItemBody>
  </InventoryItem>
)

```

Nom du paramètre	Description
children	les éléments à rendre à l'intérieur de l'élément

InventoryItemTitle

Crée un titre pour un article de carte d'inventaire ; utilisé dans **InventoryItem**.

```

return (
  <InventoryItem>
    <InventoryItemTitle>{title}</InventoryItemTitle>
    <InventoryItemBody error={loadError}>
      {loaded && <InventoryItemStatus count={workerNodes.length} icon={<MonitoringIcon />} />}
    </InventoryItemBody>
  </InventoryItem>
)

```

Nom du paramètre	Description
children	à rendre à l'intérieur du titre

InventoryItemBody

Crée le corps d'une carte d'inventaire ; utilisé dans **InventoryCard** et peut être utilisé avec **InventoryTitle**.

```

return (
  <InventoryItem>
    <InventoryItemTitle>{title}</InventoryItemTitle>
    <InventoryItemBody error={loadError}>
      {loaded && <InventoryItemStatus count={workerNodes.length} icon={<MonitoringIcon />} />}
    </InventoryItemBody>
  </InventoryItem>
)

```

Nom du paramètre	Description
children	à rendre dans la carte d'inventaire ou le titre
error	des éléments de la div

InventoryItemStatus

Crée un compte et une icône pour une fiche d'inventaire avec une adresse de lien optionnelle

InventoryItemBody

```

return (
  <InventoryItem>
    <InventoryItemTitle>{title}</InventoryItemTitle>
    <InventoryItemBody error={loadError}>
      {loaded && <InventoryItemStatus count={workerNodes.length} icon={<MonitoringIcon />} />}
    </InventoryItemBody>
  </InventoryItem>
)

```

Nom du paramètre	Description
count	compter pour l'affichage
icon	icône pour l'affichage
linkTo	(facultatif) adresse du lien

InventoryItemLoading

Crée un conteneur squelette pour le chargement d'une carte d'inventaire ; utilisé avec **InventoryItem** et les composants connexes

```

if (loadError) {
  title = <Link to={workerNodesLink}>{t('Worker Nodes')}</Link>;
} else if (!loaded) {
  title = <><InventoryItemLoading /><Link to={workerNodesLink}>{t('Worker Nodes')}</Link></>;
}
return (
  <InventoryItem>
    <InventoryItemTitle>{title}</InventoryItemTitle>
  </InventoryItem>
)

```

useFlag

Crochet qui renvoie l'indicateur de fonctionnalité donné à partir de l'état redux FLAGS. Il renvoie la valeur booléenne de l'indicateur demandé ou undefined.

Nom du paramètre	Description
flag	L'indicateur de caractéristique à renvoyer

YAMLEditor

Un éditeur YAML basique à chargement paresseux avec aide au survol et complétion.

```

<React.Suspense fallback={<LoadingBox />}>
  <YAMLEditor
    value={code}
  />
</React.Suspense>

```

Nom du paramètre	Description
value	Chaîne représentant le code yaml à rendre.
options	Les options de l'éditeur Monaco. Pour plus de détails, veuillez consulter le site https://microsoft.github.io/monaco-editor/api/interfaces/monaco.editor.IStandaloneEditorConstructionOptions.html .
minHeight	Hauteur minimale de l'éditeur dans les valeurs de hauteur CSS valides.
showShortcuts	Booléen pour afficher les raccourcis en haut de l'éditeur.
toolbarLinks	Tableau de ReactNode rendu dans la section des liens de la barre d'outils en haut de l'éditeur.
onChange	Rappel pour l'événement de changement de code.
onSave	Rappel appelé lorsque la commande CTRL / CMD S est déclenchée.
ref	Référence React à { editor?: IStandaloneCodeEditor } En utilisant la propriété "editor", vous pouvez accéder à toutes les méthodes de contrôle de l'éditeur. Pour plus d'informations, consultez le site https://microsoft.github.io/monaco-editor/api/interfaces/monaco.editor.IStandaloneCodeEditor.html .

ResourceYAMLEditor

Un éditeur YAML à chargement paresseux pour les ressources Kubernetes avec aide au survol et complétion. Le composant utilise l'éditeur YAMLEditor et y ajoute des fonctionnalités telles que la gestion de la mise à jour des ressources, des alertes, des boutons de sauvegarde, d'annulation et de rechargement, l'accessibilité et plus encore. Sauf si **onSave** callback est fourni, la mise à jour des ressources est automatiquement gérée. Il doit être enveloppé dans un composant React.Suspense.

```
<React.Suspense fallback={<LoadingBox />}>
  <ResourceYAMLEditor
    initialResource={resource}
    header="Create resource"
    onSave={(content) => updateResource(content)}
  />
</React.Suspense>
```

Nom du paramètre	Description
------------------	-------------

Nom du paramètre	Description
initialResource	YAML/Objet représentant une ressource à afficher par l'éditeur. Cet objet n'est utilisé que lors du rendu initial
header	Ajouter un en-tête au dessus de l'éditeur YAML
onSave	Rappel pour le bouton Enregistrer. Le fait de le passer permet de remplacer la mise à jour par défaut effectuée sur la ressource par l'éditeur

ResourceEventStream

Composant permettant d'afficher les événements liés à une ressource particulière.

```
const [resource, loaded, loadError] = useK8sWatchResource(clusterResource);
return <ResourceEventStream resource={resource} />
```

Nom du paramètre	Description
resource	Objet dont les événements connexes doivent être affichés.

usePrometheusPoll

Met en place une interrogation de Prometheus pour une seule requête. Il renvoie un tuple contenant la réponse à la requête, un drapeau booléen indiquant si la réponse a été complétée, et toutes les erreurs rencontrées pendant la requête ou le post-traitement de la requête.

Nom du paramètre	Description
<code>props.endpoint</code>	{PrometheusEndpoint} props.endpoint - un des PrometheusEndpoint (label, query, range, rules, targets)
<code>props.query</code>	{chaîne} [props.query] - (facultatif) Chaîne de requête Prometheus. Si elle est vide ou indéfinie, l'interrogation n'est pas lancée.
<code>props.delay</code>	{number} [props.delay] - Intervalle (facultatif) du délai d'interrogation (ms)
<code>props.endTime</code>	{number} [props.endTime] - (optionnel) pour l'endpoint QUERY_RANGE, fin de l'intervalle de la requête

Nom du paramètre	Description
<code>number</code>	{number} [props.samples] - (optionnel) pour l'enpoint QUERY_RANGE
<code>timespan</code>	{number} [options.timespan] - (optionnel) pour l'enpoint QUERY_RANGE
<code>namespace</code>	{chaîne} [options.namespace] - (optionnel) un paramètre de recherche à ajouter
<code>timeout</code>	{chaîne} [options.timeout] - (optionnel) un paramètre de recherche à ajouter

Timestamp

Composant permettant de restituer les horodatages. Les horodatages sont synchronisés entre les différentes instances du composant Horodatage. L'horodatage fourni est formaté en fonction de la locale de l'utilisateur.

Nom du paramètre	Description
timestamp	l'horodatage à rendre. Le format doit être ISO 8601 (utilisé par Kubernetes), epoch timestamp, ou une instance de Date.
simple	rendre une version simple du composant en omettant l'icône et l'infobulle.
omitSuffix	formate la date en omettant le suffixe.
className	nom de classe supplémentaire pour le composant.

useModal

Un crochet pour lancer les modales.

```
const context: AppPage: React.FC = () => {<br/> const [launchModal] = useModal();<br/> const
onClick = () => launchModal(ModalComponent);<br/> return (<br/> <Button onClick=
{onClick}>Launch a Modal</Button><br/> )<br/>}<br/>`
```

ActionServiceProvider

Composant permettant de recevoir des contributions d'autres plugins pour le type d'extension **console.action/provider**.

```
const context: ActionContext = { 'a-context-id': { dataFromDynamicPlugin } };

...

<ActionServiceProvider context={context}>
  {{{ actions, options, loaded }}} =>
```

```

        loaded && (
          <ActionMenu actions={actions} options={options} variant={ActionMenuVariant.DROPDOWN}
        />
        )
      }
    </ActionServiceProvider>

```

Nom du paramètre	Description
context	Objet avec contextId et des données de plugin optionnelles

NamespaceBar

Un composant qui rend une barre d'outils horizontale avec un menu déroulant d'espace de noms dans la position la plus à gauche. D'autres composants peuvent être ajoutés en tant qu'enfants et seront affichés à droite du menu déroulant de l'espace de noms. Ce composant est conçu pour être utilisé en haut de la page. Il doit être utilisé sur les pages où l'utilisateur doit pouvoir changer l'espace de noms actif, par exemple sur les pages contenant des ressources k8s.

```

const logNamespaceChange = (namespace) => console.log(`New namespace: ${namespace}`);

...

<NamespaceBar onNamespaceChange={logNamespaceChange}>
  <NamespaceBarApplicationSelector />
</NamespaceBar>
<Page>

...

```

Nom du paramètre	Description
onNamespaceChange	(facultatif) Fonction exécutée lorsqu'une option d'espace de noms est sélectionnée. Elle accepte comme seul argument le nouvel espace de noms sous la forme d'une chaîne de caractères. L'espace de noms actif est mis à jour automatiquement lorsqu'une option est sélectionnée, mais une logique supplémentaire peut être appliquée via cette fonction. Lorsque l'espace de noms est modifié, le paramètre d'espace de noms dans l'URL passe de l'espace de noms précédent à l'espace de noms nouvellement sélectionné.
isDisabled	(facultatif) Indicateur booléen qui désactive la liste déroulante de l'espace de noms s'il vaut true. Cette option ne s'applique qu'à la liste déroulante de l'espace de noms et n'a aucun effet sur les composants enfants.

Nom du paramètre	Description
children	(facultatif) Éléments supplémentaires à afficher dans la barre d'outils à droite de la liste déroulante de l'espace de noms.

ErrorBoundaryFallbackPage

Crée un composant `ErrorBoundaryFallbackPage` pleine page pour afficher le message "Oh non ! Quelque chose n'a pas fonctionné." ainsi que la trace de la pile et d'autres informations de débogage utiles. Ce composant doit être utilisé en conjonction avec un autre composant.

```
//in ErrorBoundary component
return (
  if (this.state.hasError) {
    return <ErrorBoundaryFallbackPage errorMessage={errorString} componentStack=
{componentStackString}
    stack={stackTraceString} title={errorString}/>;
  }

  return this.props.children;
)
```

Nom du paramètre	Description
errorMessage	description textuelle du message d'erreur
componentStack	trace du composant de l'exception
stack	trace de la pile de l'exception
title	titre à rendre en tant qu'en-tête de la page limite d'erreur

PerspectiveContext

@deprecated - use the provided **usePerspectiveContext** instead Crée le contexte de la perspective.

Nom du paramètre	Description
PerspectiveContextType	avec une perspective active et un filtre

useAccessReviewAllowed

@deprecated - Use `useAccessReview` from `@console/dynamic-plugin-sdk` instead. Hook that provides allowed status about user access to a given resource. Il renvoie la valeur booléenne **isAllowed**.

Nom du paramètre	Description
resourceAttributes	attributs des ressources pour l'examen de l'accès
impersonate	détails de l'usurpation d'identité

useSafetyFirst

@deprecated - Ce hook n'est pas lié à la fonctionnalité de la console. Hook qui assure un réglage asynchrone sûr de l'état de React dans le cas où un composant donné pourrait être démonté. Il retourne un tableau avec une paire de valeur d'état et sa fonction set.

Nom du paramètre	Description
initialState	valeur de l'état initial

7.5.3. Dépannage de votre plugin dynamique

Consultez cette liste de conseils de dépannage si vous rencontrez des problèmes lors du chargement de votre plugin.

- Vérifiez que vous avez activé votre plugin dans la configuration de l'opérateur de la console et que le nom de votre plugin apparaît dans la sortie en exécutant la commande suivante :


```
$ oc get console.operator.openshift.io cluster -o jsonpath='{.spec.plugins}'
```

 - Vérifiez les plugins activés sur la carte de statut de la page **Overview** dans le site **Administrator perspective**. Vous devrez rafraîchir votre navigateur si le plugin a été récemment activé.
- Vérifiez que votre service de plugin est sain :
 - Vérification de l'état de votre plugin pod et de la disponibilité de vos conteneurs.
 - Vérifier que le sélecteur d'étiquette de service correspond au pod et que le port cible est correct.
 - Curl le **plugin-manifest.json** du service dans un terminal sur le pod de console ou un autre pod sur le cluster.
- Vérifiez que le nom de la ressource **ConsolePlugin** (**consolePlugin.name**) correspond au nom du plugin utilisé dans **package.json**.
- Vérifiez que le nom du service, l'espace de noms, le port et le chemin d'accès sont correctement déclarés dans la ressource **ConsolePlugin**.
- Vérifiez que votre service de plugin utilise HTTPS et des certificats de service.
- Vérifiez les certificats ou les erreurs de connexion dans les journaux de la console.
- Vérifiez que l'indicateur de fonctionnalité sur lequel votre plugin s'appuie n'est pas désactivé.

- Vérifiez que votre plugin n'a pas de **consolePlugin.dependencies** dans **package.json** qui ne soit pas respecté.
 - Il peut s'agir de dépendances de la version de la console ou de dépendances d'autres plugins. Filtrez la console JS de votre navigateur sur le nom de votre plugin pour voir les messages qui sont enregistrés.
- Vérifier qu'il n'y a pas de fautes de frappe dans la perspective de l'extension navale ou dans les identifiants de section.
 - Votre plugin peut être chargé, mais des éléments de navigation manquent si les identifiants sont incorrects. Essayez de naviguer vers une page de plugin directement en modifiant l'URL.
- Vérifiez qu'aucune stratégie de réseau ne bloque le trafic entre le pod de la console et votre service de plugin.
 - Si nécessaire, ajustez les politiques de réseau pour permettre aux pods de console dans l'espace de noms openshift-console de faire des requêtes à votre service.
- Vérifiez la liste des plugins dynamiques à charger dans votre navigateur dans l'onglet **Console** du navigateur des outils de développement.
 - Évaluez **window.SERVER_FLAGS.consolePlugins** pour voir le plugin dynamique sur le frontend de la console.

Ressources supplémentaires

[Comprendre les certificats de service après-vente](#)

CHAPITRE 8. A PROPOS DU TERMINAL WEB DANS LA CONSOLE WEB

Vous pouvez lancer une instance de terminal de ligne de commande intégrée dans la console web. Vous devez d'abord installer le Web Terminal Operator pour utiliser le terminal web.



NOTE

Les administrateurs de cluster peuvent accéder au terminal web dans OpenShift Container Platform 4.7 et les versions ultérieures.

Cette instance de terminal est préinstallée avec les outils CLI courants pour interagir avec le cluster, tels que **oc**, **kubectlo**, **kn**, **tkn**, **helm**, **kubens**, **subctl**, et **kubectx**. Il a également le contexte du projet sur lequel vous travaillez et vous connecte automatiquement à l'aide de vos informations d'identification.

8.1. INSTALLATION DU TERMINAL WEB

Vous pouvez installer le terminal Web à l'aide de l'opérateur de terminal Web répertorié dans l'OperatorHub de OpenShift Container Platform. Lorsque vous installez l'opérateur de terminal Web, les définitions de ressources personnalisées (CRD) requises pour la configuration de la ligne de commande, telles que la CRD **DevWorkspace**, sont automatiquement installées. La console web crée les ressources requises lorsque vous ouvrez le terminal web.

Conditions préalables

- Accès à un cluster OpenShift Container Platform à l'aide d'un compte disposant des autorisations **cluster-admin**.

Procédure

1. Dans la perspective **Administrator** de la console web, naviguez vers **Operators** → **OperatorHub**.
2. Utilisez la boîte **Filter by keyword** pour rechercher l'opérateur **Web Terminal** dans le catalogue, puis cliquez sur la tuile **Web Terminal**.
3. Lisez la brève description de l'opérateur sur la page **Web Terminal**, puis cliquez sur **Install**.
4. Sur la page **Install Operator**, conservez les valeurs par défaut pour tous les champs.
 - L'option **fast** du menu **Update Channel** permet d'installer la dernière version du Web Terminal Operator.
 - L'option **All namespaces on the cluster** du menu **Installation Mode** permet à l'opérateur de surveiller tous les espaces de noms du cluster et d'y être disponible.
 - L'option **openshift-operators** du menu **Installed Namespace** installe l'opérateur dans l'espace de noms par défaut **openshift-operators**.
 - L'option **Automatic** du menu **Approval Strategy** garantit que les futures mises à jour de l'opérateur sont gérées automatiquement par le gestionnaire du cycle de vie de l'opérateur.
5. Cliquez sur **Install**.

6. Dans la page **Installed Operators**, cliquez sur **View Operator** pour vérifier que l'opérateur figure sur la page **Installed Operators**.




NOTE

Avant OpenShift Container Platform 4.8, l'opérateur de terminal Web regroupait toutes ses fonctionnalités dans un seul opérateur. Depuis OpenShift Container Platform 4.8, l'Opérateur Web Terminal installe l'Opérateur DevWorkspace en tant que dépendance pour fournir les mêmes fonctionnalités.

7. Une fois l'opérateur installé, rafraîchissez votre page pour voir l'icône du terminal de ligne de commande en haut à droite de la console.

8.2. UTILISATION DU TERMINAL WEB

Une fois le Web Terminal Operator installé, vous pouvez utiliser le terminal web comme suit :

1. Pour lancer le terminal web, cliquez sur l'icône du terminal de ligne de commande () en haut à droite de la console. Une instance de terminal web s'affiche dans le volet **Command line terminal**. Cette instance est automatiquement connectée à l'aide de vos informations d'identification.
2. Sélectionnez le projet dans lequel le CR **DevWorkspace** doit être créé dans la liste déroulante **Project**. Par défaut, le projet actuel est sélectionné.



NOTE

- Le CR **DevWorkspace** n'est créé que s'il n'existe pas déjà.
- Le projet **openshift-terminal** est le projet par défaut utilisé par les administrateurs de clusters. Ils n'ont pas la possibilité de choisir un autre projet.

3. Cliquez sur **Start** pour initialiser le terminal web en utilisant le projet sélectionné. Une fois le terminal web initialisé, vous pouvez utiliser les outils CLI préinstallés tels que **oc**, **kubectl**, **odo**, **kn**, **tkn**, **helm**, **kubens**, **subctl**, et **kubectx** dans le terminal web.
4. Cliquez sur pour ouvrir plusieurs onglets dans le terminal web de la console.



NOTE

- Vous pouvez réexécuter des commandes en les sélectionnant dans la liste des commandes que vous avez exécutées dans le terminal. Ces commandes sont conservées pendant plusieurs sessions du terminal.
- Le terminal web reste ouvert jusqu'à ce que vous le fermiez ou que vous fermiez la fenêtre ou l'onglet du navigateur.

8.2.1. Terminal web et politiques de réseau

Le lancement des terminaux Web peut échouer si le cluster a des politiques de réseau configurées. Pour initialiser une instance de terminal Web, l'opérateur de terminal Web doit communiquer avec le pod du terminal Web pour vérifier qu'il fonctionne, et la console Web d'OpenShift Container Platform doit

envoyer des informations pour se connecter automatiquement au cluster dans le terminal. Si l'une de ces étapes échoue, le terminal web ne s'initialise pas et le panneau du terminal semble être dans un état de chargement.

Pour éviter ce problème, assurez-vous que les politiques de réseau pour les espaces de noms utilisés pour les terminaux autorisent l'entrée à partir des espaces de noms **openshift-console** et **openshift-operators**.

8.3. DÉINSTALLATION DU TERMINAL WEB

La désinstallation du terminal web se fait en deux étapes :

1. Désinstallez l'Opérateur de terminal Web et les ressources personnalisées (CR) associées qui ont été ajoutées lors de l'installation de l'Opérateur.
2. Désinstaller l'opérateur DevWorkspace et ses ressources personnalisées connexes qui ont été ajoutées en tant que dépendance de l'opérateur Web Terminal.

La désinstallation de l'Opérateur de terminal Web ne supprime pas les définitions de ressources personnalisées (CRD) ni les ressources gérées créées lors de l'installation de l'Opérateur. Ces composants doivent être désinstallés manuellement pour des raisons de sécurité. La suppression de ces composants vous permet également d'économiser les ressources du cluster en garantissant que les terminaux ne restent pas inactifs lorsque l'opérateur est désinstallé.

Conditions préalables

- Accès à un cluster OpenShift Container Platform à l'aide d'un compte disposant des autorisations **cluster-admin**.

8.3.1. Suppression de l'opérateur de terminal Web et des ressources personnalisées qui le prennent en charge

Utilisez la console et le CLI pour supprimer tous les terminaux web et CR existants qui ont été créés lors de l'installation du Web Terminal Operator.




NOTE

Avant OpenShift Container Platform 4.8, l'opérateur de terminal Web utilisait différents CRD pour fournir des fonctionnalités de terminal Web. Pour désinstaller les versions 1.2.1 et inférieures du Web Terminal Operator, reportez-vous à la documentation d'OpenShift Container Platform 4.7.

Procédure

1. Désinstallez le Web Terminal Operator à l'aide de la console web :
 - a. Dans la perspective **Administrator** de la console web, naviguez vers **Operators → Installed Operators**.
 - b. Faites défiler la liste des filtres ou tapez un mot-clé dans la case **Filter by name** pour trouver l'opérateur **Web Terminal**.

- c. Cliquez sur le menu Options  pour l'opérateur du terminal Web, puis sélectionnez **Uninstall Operator**.
- d. Dans la boîte de dialogue de confirmation **Uninstall Operator**, cliquez sur **Uninstall** pour supprimer l'opérateur, les déploiements de l'opérateur et les pods du cluster. L'opérateur cesse de fonctionner et ne reçoit plus de mises à jour.

2. Supprimer les CR utilisés par l'Opérateur.

```
$ oc delete devworkspaces.workspace.devfile.io --all-namespaces \
--selector 'console.openshift.io/terminal=true' --wait
```

```
$ oc delete devworkspacetemplates.workspace.devfile.io --all-namespaces \
--selector 'console.openshift.io/terminal=true' --wait
```

8.3.2. Suppression de la dépendance de l'opérateur DevWorkspace

Utilisez le CLI pour supprimer les définitions de ressources personnalisées (CRD) et les ressources supplémentaires créées lors de l'installation de Web Terminal Operator.



IMPORTANT

L'opérateur DevWorkspace fonctionne comme un opérateur autonome et peut être requis en tant que dépendance pour d'autres opérateurs installés sur le cluster (par exemple, l'opérateur Red Hat OpenShift Dev Spaces peut en dépendre). Suivez les étapes ci-dessous uniquement si vous êtes sûr que l'Opérateur DevWorkspace n'est plus nécessaire.

Procédure

1. Supprimer les ressources personnalisées **DevWorkspace** utilisées par l'opérateur, ainsi que tous les objets Kubernetes associés, tels que les déploiements.

```
$ oc delete devworkspaces.workspace.devfile.io --all-namespaces --all --wait
```

```
$ oc delete devworkspaceroutings.controller.devfile.io --all-namespaces --all --wait
```



AVERTISSEMENT

Si cette étape n'est pas terminée, les finaliseurs rendent difficile la désinstallation complète de l'opérateur.

2. Retirer les CRD utilisés par l'opérateur :



AVERTISSEMENT

L'opérateur DevWorkspace fournit des définitions de ressources personnalisées (CRD) qui utilisent des webhooks de conversion. Le fait de ne pas supprimer ces CRD peut entraîner des problèmes sur le cluster.

```
$ oc delete customresourcedefinitions.apiextensions.k8s.io
devworkspaceroutings.controller.devfile.io
```

```
$ oc delete customresourcedefinitions.apiextensions.k8s.io
devworkspaces.workspace.devfile.io
```

```
$ oc delete customresourcedefinitions.apiextensions.k8s.io
devworkspacetemplates.workspace.devfile.io
```

```
$ oc delete customresourcedefinitions.apiextensions.k8s.io
devworkspaceoperatorconfigs.controller.devfile.io
```

- Vérifiez que toutes les définitions de ressources personnalisées concernées sont supprimées. La commande suivante ne doit afficher aucun résultat.

```
$ oc get customresourcedefinitions.apiextensions.k8s.io | grep "devfile.io"
```

- Supprimez les webhooks de déploiement, de mutation et de validation de **devworkspace-webhook-server**:

```
$ oc delete deployment/devworkspace-webhook-server -n openshift-operators
```

```
$ oc delete mutatingwebhookconfigurations controller.devfile.io
```

```
$ oc delete validatingwebhookconfigurations controller.devfile.io
```



NOTE

Si vous supprimez le déploiement **devworkspace-webhook-server** sans supprimer les webhooks de mutation et de validation, vous ne pourrez pas utiliser les commandes **oc exec** pour exécuter des commandes dans un conteneur sur le cluster. Après avoir supprimé les webhooks, vous pourrez à nouveau utiliser les commandes **oc exec**.

- Supprimez tous les services, secrets et cartes de configuration restants. En fonction de l'installation, certaines ressources incluses dans la commande suivante peuvent ne pas exister sur le cluster.

```
$ oc delete all --selector app.kubernetes.io/part-of=devworkspace-
operator,app.kubernetes.io/name=devworkspace-webhook-server -n openshift-operators
```

```
$ oc delete serviceaccounts devworkspace-webhook-server -n openshift-operators
```

```
$ oc delete configmap devworkspace-controller -n openshift-operators
```

```
$ oc delete clusterrole devworkspace-webhook-server
```

```
$ oc delete clusterrolebinding devworkspace-webhook-server
```

6. Désinstallez l'opérateur à l'aide de la console web :

- a. Dans la perspective **Administrator** de la console web, naviguez vers **Operators → Installed Operators**.
- b. Faites défiler la liste des filtres ou tapez un mot-clé dans la case **Filter by name** pour trouver l'opérateur **DevWorkspace**.
- c. Cliquez sur le menu Options  pour l'opérateur DevWorkspace, puis sélectionnez **Uninstall Operator**.
- d. Dans la boîte de dialogue de confirmation **Uninstall Operator**, cliquez sur **Uninstall** pour supprimer l'opérateur, les déploiements de l'opérateur et les pods du cluster. L'opérateur cesse de fonctionner et ne reçoit plus de mises à jour.

CHAPITRE 9. DÉSACTIVER LA CONSOLE WEB DANS OPENSHIFT CONTAINER PLATFORM

Vous pouvez désactiver la console web de OpenShift Container Platform.

9.1. CONDITIONS PRÉALABLES

- Déployer un cluster OpenShift Container Platform.

9.2. DÉSACTIVATION DE LA CONSOLE WEB

Vous pouvez désactiver la console web en modifiant la ressource **consoles.operator.openshift.io**.

- Modifier la ressource **consoles.operator.openshift.io**:

```
$ oc edit consoles.operator.openshift.io cluster
```

L'exemple suivant affiche les paramètres de cette ressource que vous pouvez modifier :

```
apiVersion: operator.openshift.io/v1
kind: Console
metadata:
  name: cluster
spec:
  managementState: Removed 1
```

- 1 Définissez la valeur du paramètre **managementState** sur **Removed** pour désactiver la console web. Les autres valeurs valables pour ce paramètre sont **Managed**, qui active la console sous le contrôle du cluster, et **Unmanaged**, qui signifie que vous prenez le contrôle de la gestion de la console web.

CHAPITRE 10. CRÉER DES TUTORIELS DE DÉMARRAGE RAPIDE DANS LA CONSOLE WEB

Si vous créez des tutoriels de démarrage rapide pour la console web d'OpenShift Container Platform, suivez ces directives pour maintenir une expérience utilisateur cohérente dans tous les démarrages rapides.

10.1. COMPRENDRE LES DÉMARRAGES RAPIDES

Un démarrage rapide est un tutoriel guidé avec des tâches d'utilisateur. Dans la console web, vous pouvez accéder aux démarrages rapides dans le menu **Help**. Ils sont particulièrement utiles pour se familiariser avec une application, un opérateur ou un autre produit.

Un démarrage rapide se compose principalement de tâches et d'étapes. Chaque tâche comporte plusieurs étapes, et chaque démarrage rapide comporte plusieurs tâches. Par exemple :

- Tâche 1
 - Étape 1
 - Étape 2
 - Étape 3
- Tâche 2
 - Étape 1
 - Étape 2
 - Étape 3
- Tâche 3
 - Étape 1
 - Étape 2
 - Étape 3

10.2. DÉMARRAGE RAPIDE DU FLUX DE TRAVAIL DE L'UTILISATEUR

Lorsque vous interagissez avec un didacticiel de démarrage rapide existant, c'est l'expérience attendue en matière de flux de travail :

1. Dans la perspective **Administrator** ou **Developer**, cliquez sur **Help icon** et sélectionnez **Quick Starts**.
2. Cliquez sur une carte de démarrage rapide.
3. Dans le panneau qui apparaît, cliquez sur **Start**.
4. Complétez les instructions à l'écran, puis cliquez sur **Next**.
5. Dans le module **Check your work** qui s'affiche, répondez à la question pour confirmer que vous avez effectué la tâche avec succès.

- a. Si vous sélectionnez **Yes**, cliquez sur **Next** pour passer à la tâche suivante.
 - b. Si vous sélectionnez **No**, répétez les instructions de la tâche et vérifiez à nouveau votre travail.
6. Répétez les étapes 1 à 6 ci-dessus pour effectuer les tâches restantes de la procédure de démarrage rapide.
 7. Après avoir effectué la dernière tâche, cliquez sur **Close** pour fermer le démarrage rapide.

10.3. COMPOSANTS DE DÉMARRAGE RAPIDE

Un démarrage rapide se compose des sections suivantes :

- **Card**: La tuile du catalogue qui fournit les informations de base du démarrage rapide, y compris le titre, la description, le temps consacré et l'état d'avancement
- **Introduction**: Un bref aperçu de l'objectif et des tâches du démarrage rapide
- **Task headings**: Titres hyperliés pour chaque tâche de l'introduction rapide
- **Check your work module** Un module permettant à un utilisateur de confirmer qu'il a terminé une tâche avec succès avant de passer à la tâche suivante dans le démarrage rapide
- **Hints**: Une animation pour aider les utilisateurs à identifier des zones spécifiques du produit
- **Buttons**
 - **Next and back buttons** Boutons permettant de naviguer dans les étapes et les modules de chaque tâche d'un démarrage rapide
 - **Final screen buttons** Boutons permettant de fermer le démarrage rapide, de revenir aux tâches précédentes dans le démarrage rapide et d'afficher tous les démarrages rapides

Le contenu principal d'un démarrage rapide comprend les sections suivantes :

- **Card copy**
- **Introduction**
- **Task steps**
- **Modals and in-app messaging**
- **Check your work module**

10.4. CONTRIBUER AUX DÉMARRAGES RAPIDES

OpenShift Container Platform introduit la ressource personnalisée quick start, qui est définie par un objet **ConsoleQuickStart**. Les opérateurs et les administrateurs peuvent utiliser cette ressource pour contribuer aux démarrages rapides du cluster.

Conditions préalables

- Vous devez disposer des privilèges d'administrateur de cluster.

Procédure

1. Pour créer un nouveau démarrage rapide, exécutez :

```
oc get -o yaml consolequickstart spring-with-s2i > my-quick-start.yaml
```

2. Exécutez :

```
$ oc create -f my-quick-start.yaml
```

3. Mettez à jour le fichier YAML en suivant les instructions données dans cette documentation.
4. Enregistrez vos modifications.

10.4.1. Consulter la documentation de l'API de démarrage rapide

Procédure

- Pour consulter la documentation de l'API de démarrage rapide, exécutez :

```
$ oc explain consolequickstarts
```

Exécutez **oc explain -h** pour plus d'informations sur l'utilisation de **oc explain**.

10.4.2. Mise en correspondance des éléments de l'aide-mémoire avec le CR de l'aide-mémoire

Cette section vous aide à établir une correspondance visuelle entre les parties de la ressource personnalisée (CR) du démarrage rapide et l'endroit où elles apparaissent dans le démarrage rapide au sein de la console Web.

10.4.2.1. élément de conclusion

Visualisation de l'élément de conclusion dans le fichier YAML

```
...
summary:
  failed: Try the steps again.
  success: Your Spring application is running.
title: Run the Spring application
conclusion: >-
  Your Spring application is deployed and ready. 1
```

- 1 texte de conclusion

Visualisation de l'élément de conclusion dans la console web

La conclusion figure dans la dernière section de l'introduction rapide.

Get started with Spring 10 minutes



- 1 Create a Spring application
- 2 View the build status
- 3 View the associated Git repository
- 4 View the pod status
- 5 Change the deployment icon to Spring
- 6 Run the Spring application

Your Spring application is deployed and ready.

10.4.2.2. élément de description

Visualisation de l'élément description dans le fichier YAML

```
apiVersion: console.openshift.io/v1
kind: ConsoleQuickStart
metadata:
  name: spring-with-s2i
spec:
  description: 'Import a Spring Application from git, build, and deploy it onto OpenShift.' 1
  ...
```

1 texte de description

Visualisation de l'élément de description dans la console web

La description apparaît sur la tuile d'introduction de l'introduction rapide à la page **Quick Starts**.



Get started with Spring

🕒 10 minutes

Import a Spring Application from git, build, and deploy it onto OpenShift.

10.4.2.3. élément displayName

Visualisation de l'élément displayName dans le fichier YAML

```
apiVersion: console.openshift.io/v1
kind: ConsoleQuickStart
metadata:
  name: spring-with-s2i
spec:
  description: 'Import a Spring Application from git, build, and deploy it onto OpenShift.'
  displayName: Get started with Spring ❶
  durationMinutes: 10
```

❶ **displayName** texte.

Visualisation de l'élément displayName dans la console web

Le nom d'affichage apparaît sur la tuile d'introduction de l'introduction rapide à la page **Quick Starts**.



Get started with Spring

🕒 10 minutes

Import a Spring Application from git, build, and deploy it onto OpenShift.

10.4.2.4. élément `durationMinutes`

Visualisation de l'élément `durationMinutes` dans le fichier YAML

```
apiVersion: console.openshift.io/v1
kind: ConsoleQuickStart
metadata:
  name: spring-with-s2i
spec:
  description: 'Import a Spring Application from git, build, and deploy it onto OpenShift.'
  displayName: Get started with Spring
  durationMinutes: 10 ❶
```

❶ **`durationMinutes`** en minutes. Cette valeur définit la durée du démarrage rapide.

Visualisation de l'élément `durationMinutes` dans la console web

L'élément durée minutes apparaît sur la tuile d'introduction de l'introduction rapide à la page **Quick Starts**.

🕒 10 minutes

111

```

0LjctMzAuMTUsNDkuNzctNDAuMTFhMjEyLDIxMiwwLDAsMSw2NS45My0yNS43M0ExOTgsMTk4LDA
sMCwXLDUxMiwxMTYUuMjdHMTk2LjExLDE5Ni4xMSwwLDAsMSwzMiwwLjFjNC41LjKxLDkuMzYsMi4wN
wxNC41MywzLjUyLDYwLjQxLDIwLjQ4LDg0LjkyLDkxLjA1LTQ3LjQ0LDI0OC4wNi0yOC43NSwzNC4x
Mi0xNDAuNywxOTQuODQtMTg0LjY2LDI2OC40MmE2MzAuODYsNjMwLjg2LDAsMCwwLTMzLjlyLD
U4LjMyQzI3NiW2NTUuMzQsMjY1LjQsNTk4LDI2NS40LDUyMC4yOSwyNjUuNCwzNDAuNjEsMzExLjY
5LDI0MC43NCwzNjQuMTUsMTg1LjIzWiIvPjxwYXR0IGNsYXNzPSJjbHMtMyIgZD0iTTUyNy41NCwzO
DQuODNjODQuMDYtOTkuNywxMTYUuMDYtMTc3LjI4LDk1LjlyLTlzMCA3NCwzMS42MiwwLjY5LDI0LD
E5LjIsMzcuMDYsMzEuMTMsNTluNDgsNTUuNSw5OC43OCwwNTUuMzgsOTguNzgsMzM1LjA3LDAs
NzcuNzEtMTAuNiwxMzUuMDUtMjcuNzcsMTc3LjRhNjI4LjczLDYyOC43MywwLDAsMC0zMy4yMy01OC
4zMmMtMzktNjUuMjYtMTMxLjQ1LTE5OS0xNzEuOTMtMjUyLjI3QzUyNi4zMjYwODYUuMjksNTI3LDM4
NS41MiwwMjcuNTQsMzg0LjgzWiIvPjxwYXR0IGNsYXNzPSJjbHMtNCIgZD0iTTEzNC41OCw5MDguM
DdoLS4wNmEuMzkuMzksMCwwLDEtLjI3LS4xMWMtMTE5LjUyLTEyMS4wNy0xNTUtMjg3LjQtNDcuN
TQtNDA0LjU4LDM0LjYzLTQxLjE0LDEyMC0xNTEuNiwyMDluNzUtMjYyLjE5LTMuMTMsNy02LjEyLDE
0LjI1LTguOTIsMjEuNjktMjQuMzQsNjQuNDUtMzYuNjcsMTQ0LjMyLTM2LjY3LDIzNy40MSwwLDU2LjU
zLDUuNTgsMTA2LDE2LjU5LDE0Ny4xNEEzMDcuNDksMzA3LjQ5LDAsMCwwLDI4MC45MSwzMjND
MjM3LDgxNi44OCwyMTYUuOTMsODkzLjkzLDEzNC41OCw5MDguMDdali8+PHBhdGggY2xhc3M9ImN
scy01IiBkPSJNNTgzLjQzLDgxMy43OUUM1NjAuMTgsNzI3LjcyLDUxMiwwNjQuMTUsNTEyLDY2NC4xN
XMtNDguMTcsNjMuNTctNzEuNDMsMTQ5LjY0Yy00OC40NS02Ljc0LTEwMC45MS0yNy41Mi0xMzUu
NjYtOTEuMThhNjQ1LjY4LDY0NS42OCwwLDAsMSwzOS41Ny03MS41NGwuMjEtLjMyLjE5LS4zM2M
zOC02My42MywxMjYuNC0xOTEuMzcsMTY3LjEyLTI0NS42NiwwMC43MSw1NC4yOCwwMjkuMSwXO
DIsMTY3LjEyLDI0NS42NmwwMTkuMzMuMjEuMzJhNjQ1LjY4LDY0NS42OCwwLDAsMSwzOS41Nyw
3MS41NEM2ODQuMzQsNzg2LjI3LDYzMS44OCw4MDcuMDUsNTgzLjQzLDgxMy43OVoiLz48cGF0a
CBjbGFzc20iY2xzLTQilGQ9Ik04ODkuNzUsOTA4YS4zOS4zOSwwLDAsMS0uMjcuMTFoLS4wNkM4M
DcuMDcsODkzLjkzLDc4NywwMTYUuODgsNzQzLjA5LDcyM2EzMDcuNDksMzA3LjQ5LDAsMCwwLDIwL
jQ1LTU1LjU0YzExLTQxLjExLDE2LjU5LTkwLjYxLDE2LjU5LTE0Ny4xNCwwLTkzLjA4LTEyLjMzLDE3M
y0zNi42Ni0yMzcuNHEtNC4yMi0xMS4xNi04LjkzLTlxdjODluNzUsOTAuNTksMTY4LjEyLDIwMS4wNS
wyMDluNzUsMjYyLjE5QzEwNDQuNzksNjIwLjU2LDEwMDkuMjcsNzg2Ljg5LDg4OS43NSw5MDhali8+
PC9zdmccCg==

```

...

- 1 L'icône est définie comme une valeur base64.

Visualisation de l'élément icône dans la console web

L'icône apparaît sur la tuile d'introduction de l'introduction rapide à la page **Quick Starts**.



Get started with Spring

🕒 10 minutes

Import a Spring Application from git,
build, and deploy it onto OpenShift.

10.4.2.6. élément d'introduction

Visualisation de l'élément d'introduction dans le fichier YAML

```
...
introduction: >- 1
  **Spring** is a Java framework for building applications based on a distributed microservices
  architecture.

  - Spring enables easy packaging and configuration of Spring applications into a self-contained
  executable application which can be easily deployed as a container to OpenShift.

  - Spring applications can integrate OpenShift capabilities to provide a natural "Spring on
  OpenShift" developer experience for both existing and net-new Spring applications. For example:

  - Externalized configuration using Kubernetes ConfigMaps and integration with Spring Cloud
  Kubernetes

  - Service discovery using Kubernetes Services

  - Load balancing with Replication Controllers

  - Kubernetes health probes and integration with Spring Actuator

  - Metrics: Prometheus, Grafana, and integration with Spring Cloud Sleuth

  - Distributed tracing with Istio & Jaeger tracing

  - Developer tooling through Red Hat OpenShift and Red Hat CodeReady developer tooling to
  quickly scaffold new Spring projects, gain access to familiar Spring APIs in your favorite IDE, and
  deploy to Red Hat OpenShift
...
```

1 L'introduction présente le démarrage rapide et énumère les tâches qu'il contient.

Visualisation de l'élément d'introduction dans la console web

Après avoir cliqué sur une carte de démarrage rapide, un panneau latéral s'affiche pour présenter le démarrage rapide et énumérer les tâches qu'il contient.

Get started with Spring 10 minutes



Spring is a Java framework for building applications based on a distributed microservices architecture.

- Spring enables easy packaging and configuration of Spring applications into a self-contained executable application which can be easily deployed as a container to OpenShift.
- Spring applications can integrate OpenShift capabilities to provide a natural "Spring on OpenShift" developer experience for both existing and net-new Spring applications. For example:
 - Externalized configuration using Kubernetes ConfigMaps and integration with Spring Cloud Kubernetes
 - Service discovery using Kubernetes Services
 - Load balancing with Replication Controllers
 - Kubernetes health probes and integration with Spring Actuator
 - Metrics: Prometheus, Grafana, and integration with Spring Cloud Sleuth
 - Distributed tracing with Istio & Jaeger tracing
- Developer tooling through Red Hat OpenShift and Red Hat CodeReady developer tooling to quickly scaffold new Spring projects, gain access to familiar Spring APIs in your favorite IDE, and deploy to Red Hat OpenShift

In this quick start, you will complete 6 tasks:

- 1 Create a Spring application
- 2 View the build status
- 3 View the associated Git repository
- 4 View the pod status
- 5 Change the deployment icon to Spring
- 6 Run the Spring application

Start

10.4.3. Ajout d'une icône personnalisée à un démarrage rapide

Une icône par défaut est fournie pour tous les démarrages rapides. Vous pouvez créer votre propre icône.

Procédure

1. Trouvez le fichier **.svg** que vous souhaitez utiliser comme icône personnalisée.
2. Utilisez un [outil en ligne pour convertir le texte en base64](#) .
3. Dans le fichier YAML, ajoutez **icon: >**, puis sur la ligne suivante, incluez **data:image/svg+xml;base64** suivi de la sortie de la conversion base64. Par exemple :

data:image/svg+xml;base64,PHN2ZyB4bWxucz0iaHR0cDovL3d3dy53My5vcmcvMjAwMC9zdmcilHJvbGU9ImltZylddmllid.

10.4.4. Limiter l'accès à un démarrage rapide

Tous les programmes de démarrage rapide ne doivent pas être accessibles à tout le monde. La section **accessReviewResources** du fichier YAML permet de limiter l'accès au démarrage rapide.

Pour permettre à l'utilisateur d'accéder au démarrage rapide uniquement s'il a la possibilité de créer des ressources **HelmChartRepository**, utilisez la configuration suivante :

```
accessReviewResources:
- group: helm.openshift.io
  resource: helmchartrepositories
  verb: create
```

Pour n'autoriser l'utilisateur à accéder au démarrage rapide que s'il a la possibilité de répertorier les groupes d'opérateurs et les manifestes de paquets, et donc d'installer des opérateurs, utilisez la configuration suivante :

```
accessReviewResources:
- group: operators.coreos.com
  resource: operatorgroups
  verb: list
- group: packages.operators.coreos.com
  resource: packagemanifests
  verb: list
```

10.4.5. Liens vers d'autres démarrages rapides

Procédure

- Dans la section **nextQuickStart** du fichier YAML, indiquez le **name**, et non le **displayName**, de l'introduction rapide vers laquelle vous souhaitez établir un lien. Par exemple :

```
nextQuickStart:
- add-healthchecks
```

10.4.6. Balises prises en charge pour un démarrage rapide

Rédigez votre contenu en markdown à l'aide de ces balises. Le markdown est converti en HTML.

Étiquette	Description
'b',	Définit le texte en gras.
'img',	Incorpore une image.
'i',	Définit le texte en italique.
'strike',	Définit le texte barré.
's',	Définit un texte plus petit
'del',	Définit un texte plus petit.
'em',	Définit le texte mis en valeur.
'strong',	Définit le texte important.
'a',	Définit une balise d'ancrage.
'p',	Définit le texte du paragraphe.
'h1',	Définit une rubrique de niveau 1.
'h2',	Définit une rubrique de niveau 2.
'h3',	Définit une rubrique de niveau 3.
'h4',	Définit une rubrique de niveau 4.
'ul',	Définit une liste non ordonnée.
'ol',	Définit une liste ordonnée.
'li',	Définit un élément de liste.
'code',	Définit un texte comme un code.
'pre',	Définit un bloc de texte préformaté.
'button',	Définit un bouton dans le texte.

10.4.7. Démarrage rapide de la mise en évidence de la référence au markdown

La fonction de mise en évidence ou d'indication permet aux démarrages rapides de contenir un lien qui peut mettre en évidence et animer un composant de la console web.

La syntaxe markdown contient :

- Texte du lien entre crochets
- Le mot-clé **highlight**, suivi de l'ID de l'élément à animer

10.4.7.1. Changeur de perspective

```
[Perspective switcher]{{highlight qs-perspective-switcher}}
```

10.4.7.2. Liens de navigation du point de vue de l'administrateur

```
[Home]{{highlight qs-nav-home}}
[Operators]{{highlight qs-nav-operators}}
[Workloads]{{highlight qs-nav-workloads}}
[Serverless]{{highlight qs-nav-serverless}}
[Networking]{{highlight qs-nav-networking}}
[Storage]{{highlight qs-nav-storage}}
[Service catalog]{{highlight qs-nav-servicecatalog}}
[Compute]{{highlight qs-nav-compute}}
[User management]{{highlight qs-nav-usermanagement}}
[Administration]{{highlight qs-nav-administration}}
```

10.4.7.3. Liens de navigation du point de vue du développeur

```
[Add]{{highlight qs-nav-add}}
[Topology]{{highlight qs-nav-topology}}
[Search]{{highlight qs-nav-search}}
[Project]{{highlight qs-nav-project}}
[Helm]{{highlight qs-nav-helm}}
```

10.4.7.4. Liens de navigation communs

```
[Builds]{{highlight qs-nav-builds}}
[Pipelines]{{highlight qs-nav-pipelines}}
[Monitoring]{{highlight qs-nav-monitoring}}
```

10.4.7.5. Liens dans l'en-tête

```
[CloudShell]{{highlight qs-masthead-cloudshell}}
[Utility Menu]{{highlight qs-masthead-utilitymenu}}
[User Menu]{{highlight qs-masthead-usermenu}}
[Applications]{{highlight qs-masthead-applications}}
[Import]{{highlight qs-masthead-import}}
[Help]{{highlight qs-masthead-help}}
[Notifications]{{highlight qs-masthead-notifications}}
```

10.4.8. Référence de l'extrait de code

Vous pouvez exécuter un extrait de code CLI lorsqu'il est inclus dans un démarrage rapide à partir de la console web. Pour utiliser cette fonctionnalité, vous devez d'abord installer le Web Terminal Operator. Les actions du terminal web et des extraits de code qui s'exécutent dans le terminal web ne sont pas présentes si vous n'installez pas l'Opérateur de terminal web. Vous pouvez également copier un extrait de code dans le presse-papiers, que l'Opérateur de terminal Web soit installé ou non.

10.4.8.1. Syntaxe pour les extraits de code en ligne

```
`code block`{{copy}}
`code block`{{execute}}
```



NOTE

Si la syntaxe **execute** est utilisée, l'action **Copy to clipboard** est présente, que le Web Terminal Operator soit installé ou non.

10.4.8.2. Syntaxe pour les extraits de code multi-lignes

```
```
multi line code block
```{{copy}}

```
multi line code block
```{{execute}}
```

10.5. LIGNES DIRECTRICES SUR LE CONTENU POUR UN DÉMARRAGE RAPIDE

10.5.1. Copie de la carte

Vous pouvez personnaliser le titre et la description d'une carte de démarrage rapide, mais vous ne pouvez pas personnaliser le statut.

- La description doit se limiter à une ou deux phrases.
- Commencez par un verbe et communiquez l'objectif de l'utilisateur. Exemple correct :

```
Create a serverless application.
```

10.5.2. Introduction

Après avoir cliqué sur une carte de démarrage rapide, un panneau latéral s'affiche pour présenter le démarrage rapide et énumérer les tâches qu'il contient.

- Le contenu de votre introduction doit être clair, concis, informatif et convivial.
- Indiquer le résultat de la procédure de démarrage rapide. L'utilisateur doit comprendre l'objectif du démarrage rapide avant de commencer.
- Donner une action à l'utilisateur, et non un démarrage rapide.

- **Correct example:**

In this quick start, you will deploy a sample application to {product-title}.

- **Incorrect example:**

This quick start shows you how to deploy a sample application to {product-title}.

- L'introduction ne doit pas dépasser quatre ou cinq phrases, en fonction de la complexité de la fonctionnalité. Une longue introduction peut submerger l'utilisateur.
- Dressez la liste des tâches de démarrage rapide après le contenu de l'introduction et commencez chaque tâche par un verbe. Ne précisez pas le nombre de tâches, car la copie devrait être mise à jour chaque fois qu'une tâche est ajoutée ou supprimée.

- **Correct example:**

Tasks to complete: Create a serverless application; Connect an event source; Force a new revision

- **Incorrect example:**

You will complete these 3 tasks: Creating a serverless application; Connecting an event source; Forcing a new revision

10.5.3. Étapes de la tâche

Une fois que l'utilisateur a cliqué sur **Start**, une série d'étapes s'affiche, qu'il doit exécuter pour terminer le démarrage rapide.

Suivez ces lignes directrices générales lorsque vous rédigez les étapes de la tâche :

- Utilisez "Click" pour les boutons et les étiquettes. Utilisez "Select" pour les cases à cocher, les boutons radio et les menus déroulants.
- Utilisez "Cliquez" au lieu de "Cliquez"
 - **Correct example:**

Click OK.
 - **Incorrect example:**

Click on the OK button.
- Expliquez aux utilisateurs comment naviguer entre les perspectives **Administrator** et **Developer**. Même si vous pensez qu'un utilisateur se trouve déjà dans la perspective appropriée, donnez-lui des instructions sur la manière de s'y rendre afin qu'il soit bien là où il doit être. Exemples :

Enter the Developer perspective: In the main navigation, click the dropdown menu and select Developer.
Enter the Administrator perspective: In the main navigation, click the dropdown menu and

- select Admin.
- Utilisez la structure "Lieu, action". Indiquez à l'utilisateur où aller avant de lui dire quoi faire.
 - **Correct example:**
 - In the node.js deployment, hover over the icon.
 - **Incorrect example:**
 - Hover over the icon in the node.js deployment.
- Veillez à ce que la terminologie de vos produits soit toujours en majuscules.
- Si vous devez spécifier un type de menu ou une liste sous forme de liste déroulante, écrivez "liste déroulante" en un seul mot sans trait d'union.
- Établir une distinction claire entre une action de l'utilisateur et des informations supplémentaires sur les fonctionnalités du produit.
 - **User action:**
 - Change the time range of the dashboard by clicking the dropdown menu and selecting time range.
 - **Additional information:**
 - To look at data in a specific time frame, you can change the time range of the dashboard.
- Évitez le langage directionnel, comme "Dans le coin supérieur droit, cliquez sur l'icône". Le langage directionnel devient obsolète chaque fois que la présentation de l'interface utilisateur change. En outre, une indication destinée aux utilisateurs d'ordinateurs de bureau peut ne pas être exacte pour les utilisateurs dont la taille d'écran est différente. Identifiez plutôt un élément par son nom.
 - **Correct example:**
 - In the navigation menu, click Settings.
 - **Incorrect example:**
 - In the left-hand menu, click Settings.
- N'identifiez pas les éléments uniquement par leur couleur, comme "Cliquez sur le cercle gris". Les identificateurs de couleur ne sont pas utiles pour les utilisateurs malvoyants, en particulier les daltoniens. Identifiez plutôt un élément à l'aide de son nom ou de sa copie, comme la copie d'un bouton.
 - **Correct example:**
 - The success message indicates a connection.
 - **Incorrect example:**

The message with a green icon indicates a connection.

- Utilisez systématiquement le point de vue de la deuxième personne, vous :

- **Correct example:**

Set up your environment.

- **Incorrect example:**

Let's set up our environment.

10.5.4. Vérifiez votre module de travail

- Une fois qu'un utilisateur a terminé une étape, un module **Check your work** apparaît. Ce module invite l'utilisateur à répondre par oui ou par non à une question sur les résultats de l'étape, ce qui lui donne la possibilité de revoir son travail. Pour ce module, il suffit de rédiger une seule question de type oui ou non.
 - Si l'utilisateur répond **Yes**, une coche apparaît.
 - Si l'utilisateur répond **No**, un message d'erreur apparaît avec un lien vers la documentation pertinente, si nécessaire. L'utilisateur a alors la possibilité de revenir en arrière et de réessayer.

10.5.5. Formatage des éléments de l'interface utilisateur

Formatez les éléments de l'interface utilisateur en suivant ces lignes directrices :

- Copie des boutons, des menus déroulants, des onglets, des champs et des autres contrôles de l'interface utilisateur : Rédigez le texte tel qu'il apparaît dans l'interface utilisateur et mettez-le en gras.
- Tous les autres éléments de l'interface utilisateur, y compris les noms de page, de fenêtre et de panneau : Rédigez le texte tel qu'il apparaît dans l'interface utilisateur et mettez-le en gras.
- Code ou texte saisi par l'utilisateur : Utiliser une police monospaciale.
- Indices : Si un indice vers un élément de navigation ou de tête de page est inclus, stylisez le texte comme vous le feriez pour un lien.
- Commandes CLI : Utilisez une police monospaciale.
- Dans le texte courant, utilisez une police de caractères gras et monospaciale pour une commande.
- Si un paramètre ou une option est une valeur variable, utilisez une police de caractères italique et monospaciale.
- Utilisez une police de caractères gras et monospaciale pour le paramètre et une police de caractères monospaciale pour l'option.

10.6. RESSOURCES SUPPLÉMENTAIRES

- Pour les exigences en matière de voix [et de ton](#), reportez-vous aux [lignes directrices de PatternFly relatives à la voix et au ton de la marque](#).
- Pour d'autres conseils sur le contenu UX, reportez-vous à toutes les sections du [guide de style de rédaction UX de PatternFly](#).