

学習環境としてOpenShift Localをインストールする

Fusayuki Minamoto

Version 1.0, 2024-11-16

目次

1. はじめに.....	1
2. OpenShift Localとは.....	2
3. OpenShift Container Platformとの違い.....	3
4. OpenShift Localのインストール.....	4
4.1. 最小システム要件.....	4
4.2. 筆者のPC環境.....	4
4.3. インストール手順.....	4
5. crcコマンドの使い方.....	7
5.1. インスタンスの起動.....	7
5.2. インスタンスの停止.....	9
5.3. インスタンスの削除.....	9
5.4. インスタンスの設定.....	10
6. Webコンソールの使い方.....	11
6.1. Webコンソールの起動.....	11
6.2. Webコンソールに管理者としてログイン.....	11
6.3. Webコンソールに開発者としてログイン.....	12
7. CLIコマンドの使い方.....	13
7.1. ocコマンドの準備.....	13
7.2. ocコマンドで管理者としてログイン.....	13
8. 管理者向け機能確認.....	14
8.1. Clusterversionを調べる.....	14
8.2. Cluster Operatorを調べる.....	14
8.3. Nodeの状態を調べる.....	15
8.4. モニタリングを有効にしてみる.....	16
9. おわりに.....	18

1. はじめに

OpenShiftの学習者向けに、OpenShift Localのインストール手順と基本的なツールの使い方を説明します。OpenShift Localは開発者向けの環境ではありますが、kubeadminというアカウントでログインをすることで管理者権限が必要なコマンドも操作できることを示します。

2. OpenShift Localとは

Red Hat OpenShift Localは、無償で使うことができるシングルノードのOpenShiftです。開発者がデスクトップにインストールして開発やテスト用途に使うことが想定されています。この記事では、OpenShift Local 2.43を使ってOpenShiftをインストールします。詳細は以下のドキュメントを参照してください。

第1章 [Red Hat OpenShift Local の紹介](#) | [Red Hat Product Documentation](#)

OpenShift Localは、通常のOpenShift Container Platformよりも軽量になっているため起動時間が比較的早いです。OpenShift Localは、筆者のノートPC上では 約3分で起動 します（マシンスペックは後述）。OpenShiftの持ち運びができて、必要なときにサクッと起動できるので、デモや調査目的に気軽に使えてとても便利です。

3. OpenShift Container Platformとの違い

OpenShift Localは本番環境では使えませんし、アップグレードもできません。詳細は以下のドキュメントで確認してください。

[1.2. 実稼働環境の OpenShift Container Platform インストールとの相違点 | Red Hat Product Documentation](#)

4. OpenShift Localのインストール

OpenShift Localのインストールはとても簡単です。インストールのための最小システム要件の詳細は、以下のドキュメントを参照してください。

第2章 Red Hat OpenShift Local のインストール | Red Hat Product Documentation

4.1. 最小システム要件

CPUは、AMD64、インテル64、Appleシリコンをサポートします。OpenShift Localを動作させるためのハードウェアリソースは以下になります。

- 物理 CPU コア 4 個
- 空きメモリー 10.5 GB
- ストレージ領域の 35 GB

OSは、Windows、MacOS、Linuxをサポートします。ただし、Windowsの場合は、**Microsoft Windows Home Edition** ではインストールできません。ノートPC上への導入を検討されている方は注意してください。

4.2. 筆者のPC環境

参考までに、この記事を書くにあたり使用した筆者のノートPCのスペックは以下になります。

- Fedora Linux 41 (Workstation Edition)
- Lenovo ThinkPad X1 Carbon Gen 11
- Intel Core i7-1370P
- メモリ64GB
- ディスク1TB (外付けSSD)

ちなみに、筆者はこのノートPCを普段はWindowsマシンとして使っています。Linuxを使いたい場合は、Fedora Linux 41をインストール済みの **極小SSD** をUSBの口に指してそこからOSをブートしてLinuxマシンとして使っています。

注記

以後のインストール手順は、Fedora 41上で確認済みです。CoreOSやRHELでも同様の手順でインストールできるはずです。

4.3. インストール手順

1. まずインストールに必要なファイルを以下のサイトからダウンロードします。
https://docs.redhat.com/ja/documentation/red_hat_openshift_local/2.43/html/getting_started_guide/index

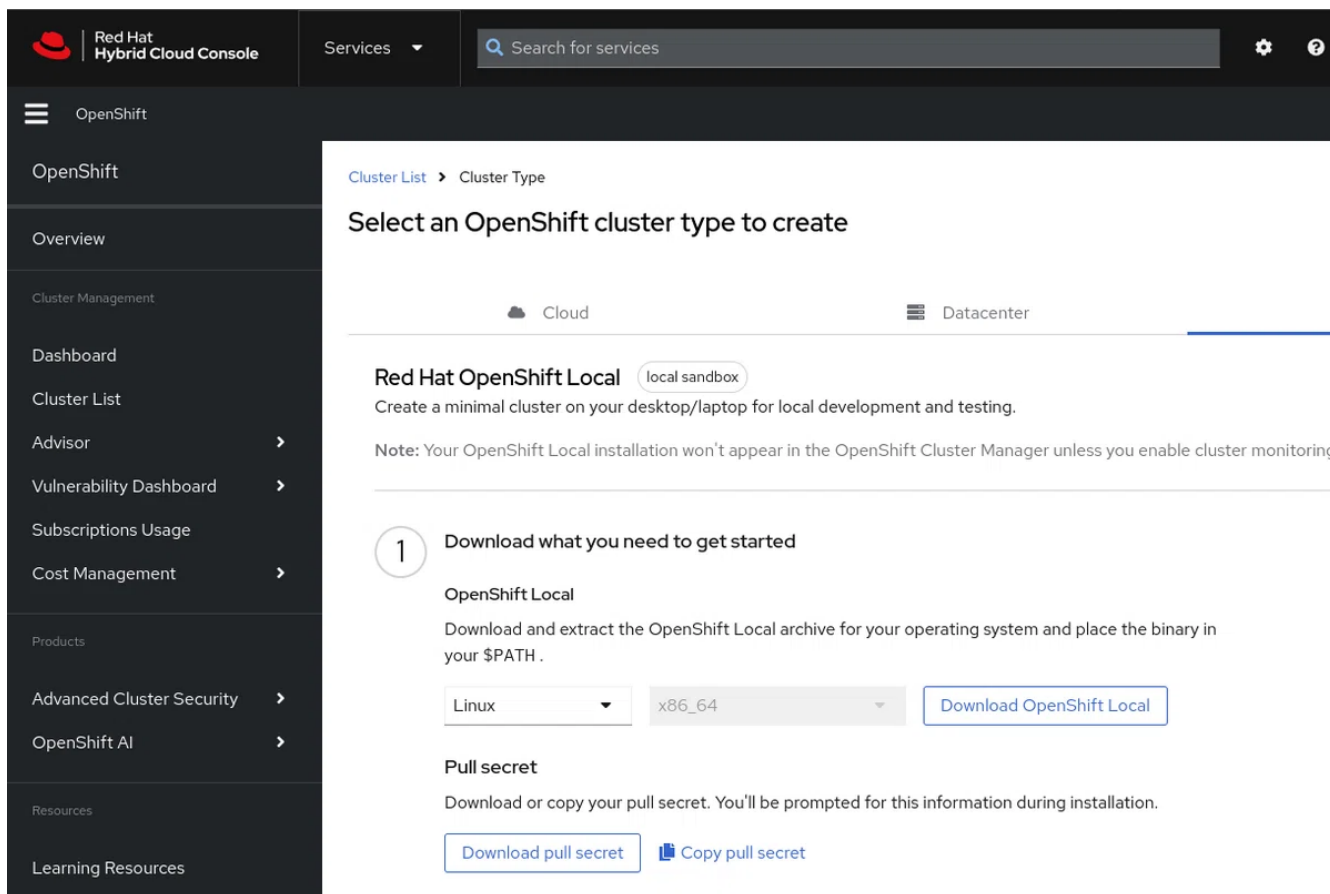


Figure 1. OpenShift Localインストーラーのダウンロード

- **[Download OpenShift Local]** ボタンを押してcrc-linux-amd64.tar.xzという名前のファイルをダウンロードします。
- **[Download pull secret]** ボタンを押して **pull-secret.txt** という名前のテキストファイルをダウンロードします。

2. ターミナルを開いて以下のコマンドを実行し、crcコマンドが使えるように準備します。

```
$ cd ~/Downloads
$ tar xvf crc-linux-amd64.tar.xz
$ mkdir -p ~/bin
$ cp ~/Downloads/crc-linux-*-amd64/crc ~/bin
$ export PATH=$PATH:$HOME/bin
$ echo 'export PATH=$PATH:$HOME/bin' >> ~/.bashrc
$ source ~/.bashrc
```

crc versionコマンドを実行して、crcが使えることを確認しましょう。

```
$ crc version
CRC version: 2.43.0+268795
OpenShift version: 4.17.1
MicroShift version: 4.17.1
```

3. **crc setup** コマンドを使ってインストールを開始します。インストールの後半で5GB以上のサイズのフ

ファイルをネットからダウンロードします。

```
$ crc setup
CRC is constantly improving and we would like to know more about usage (more details
at https://developers.redhat.com/article/tool-data-collection)
Your preference can be changed manually if desired using 'crc config set consent-
telemetry <yes/no>'
Would you like to contribute anonymous usage statistics? [y/N]: y
Thanks for helping us! You can disable telemetry with the command 'crc config set
consent-telemetry no'.
<略>
INFO Downloading bundle:
/home/student/.crc/cache/crc_libvirt_4.17.1_amd64.crcbundle...
5.04 GiB / 5.04 GiB [-----] 100.00%
1.83 MiB/s
INFO Uncompressing /home/student/.crc/cache/crc_libvirt_4.17.1_amd64.crcbundle
crc.qcow2: 18.27 GiB / 18.27 GiB [-----]
] 100.00%
oc: 158.70 MiB / 158.70 MiB [-----]
] 100.00%
Your system is correctly setup for using CRC. Use 'crc start' to start the instance
```


5. crcコマンドの使い方

crcコマンド一覧

OpenShift Localの操作は、crcコマンドを使います。 **crc help** コマンドによってcrcコマンドの一覧を表示します。

```
$ crc help

Available Commands:
bundle      Manage CRC bundles
cleanup     Undo config changes
completion  Generate the autocompletion script for the specified shell
config      Modify crc configuration
console     Open the OpenShift Web Console in the default browser
delete      Delete the instance
help        Help about any command
ip          Get IP address of the running OpenShift cluster
oc-env      Add the 'oc' executable to PATH
podman-env  Setup podman environment
setup       Set up prerequisites for using CRC
start       Start the instance
status      Display status of the OpenShift cluster
stop        Stop the instance
version     Print version information
```

5.1. インスタンスの起動

crc start コマンドでOpenShift Localの仮想マシンを起動します。crc startを実行すると、初回だけ**Pull Secret**を入力するように促されます。この際、インストールのステップ(1)でダウンロード済みのpull-secret.txtの内容を貼り付けます。

```
$ crc start
INFO Using bundle path /home/student/.crc/cache/crc_libvirt_4.17.1_amd64.crcbundle
<略>
CRC requires a pull secret to download content from Red Hat.
You can copy it from the Pull Secret section of
https://console.redhat.com/openshift/create/local.
? Please enter the pull secret
*****
INFO Creating CRC VM for OpenShift 4.17.1...
```

crc startコマンドの最後で、以下のように、コンソールのURLとログイン情報が表示されます。これでOpenShiftを利用できる準備が整いました。

```
Started the OpenShift cluster. +
The server is accessible via web console at: +
```

```
https://console-openshift-console.apps-crc.testing/[https://console-openshift-console.apps-crc.testing]
```

```
Log in as administrator: +  
Username: kubeadmin +  
Password: XXXXXXXXX
```

```
Log in as user: +  
Username: developer +  
Password: developer
```

```
Use the ``oc`` command line interface: +  
$ eval latexmath:[(crc oc-env)] oc login -u developer  
https://api.crc.testing:6443/[https://api.crc.testing:6443]
```

OpenShift Localを起動すると、仮想マシンマネージャー上でcrcという名前の仮想マシンが起動していることが確認できます。

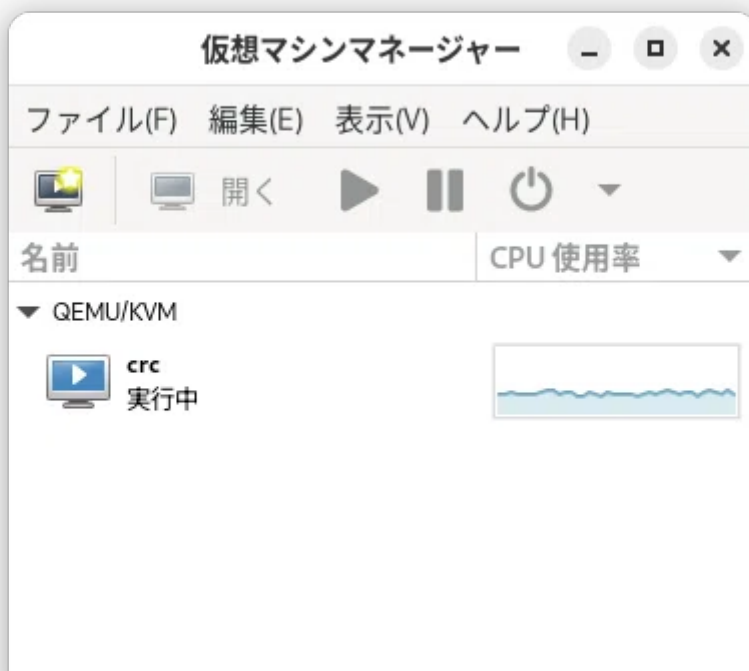


Figure 2. crc仮想マシン (起動)

crc status コマンドによって、OpenShift Localのバージョン、メモリ、ディスクの使用状況を調べることができます。

```
$ crc status  
CRC VM:      Running  
OpenShift:   Running (v4.17.1)  
RAM Usage:   6.218GB of 10.95GB
```

```
Disk Usage:    22.01GB of 32.68GB (Inside the CRC VM)
Cache Usage:    25.24GB
Cache Directory: /home/student/.crc/cache
```

5.2. インスタンスの停止

crc stop コマンドでOpenShift Localの仮想マシンを停止します。コマンド実行後、仮想マシンマネージャー上でcrcという名前の仮想マシンが停止していることが確認できます。

```
$ crc stop
INFO Stopping kubelet and all containers...
INFO Stopping the instance, this may take a few minutes...
Stopped the instance
```



Figure 3. crc仮想マシン (停止)

5.3. インスタンスの削除

OpenShiftの仮想マシンを削除するには **crc delete** を使います。

```
$ crc delete
Do you want to delete the instance? [y/N]: y
Deleted the instance
```

OpenShift Localを完全にアンインストールするコマンドはありません。 ~/.crc以下には、以下のようにキャッシュされているファイルが存在しています。これらのファイルを手で削除する必要があります。

```
$ du -h ~/.crc
8.0K    /home/student/.crc/bin/podman
4.0K    /home/student/.crc/bin/oc
17M     /home/student/.crc/bin
19G     /home/student/.crc/cache/crc_libvirt_4.17.1_amd64
24G     /home/student/.crc/cache
24G     /home/student/.crc
```

5.4. インスタンスの設定

OpenShift Localの初期設定では、CPUは4、メモリは10.5G使うように設定されていますが、環境に合わせてこれらを設定変更することができます。

crc config コマンドで現在のCPU、メモリの設定値の確認ができます。

```
crc config get cpus
Configuration property 'cpus' is not set. Default value '4' is used

$ crc config get memory
Configuration property 'memory' is not set. Default value '10752' is used
```

crc config set コマンドで現在の設定値の変更ができます。 **crc config view** コマンドで、現在の設定値の確認ができます。設定値の変更は次の起動から反映されます。

```
$ crc config set cpus 6
Changes to configuration property 'cpus' are only applied when the CRC instance is
started.
If you already have a running CRC instance, then for this configuration change to take
effect, stop the CRC instance with 'crc stop' and restart it with 'crc start'.

$ crc config set memory 21504
Changes to configuration property 'memory' are only applied when the CRC instance is
started.
If you already have a running CRC instance, then for this configuration change to take
effect, stop the CRC instance with 'crc stop' and restart it with 'crc start'.

$ crc config view
- consent-telemetry      : yes
- cpus                   : 6
- memory                 : 21504
```

6. Webコンソールの使い方

6.1. Webコンソールの起動

`crc console` コマンドによって、Webブラウザが起動し、OpenShift Webコンソールが開きます。

```
$ crc console
Opening the OpenShift Web Console in the default browser...
既存のブラウザ セッションで開いています。
```

6.2. Webコンソールに管理者としてログイン

管理コンソールにkubeadminというユーザー名でログインしてみましょう。kubadminのパスワードは、oc startコマンドの完了時に表示されたものを使います。

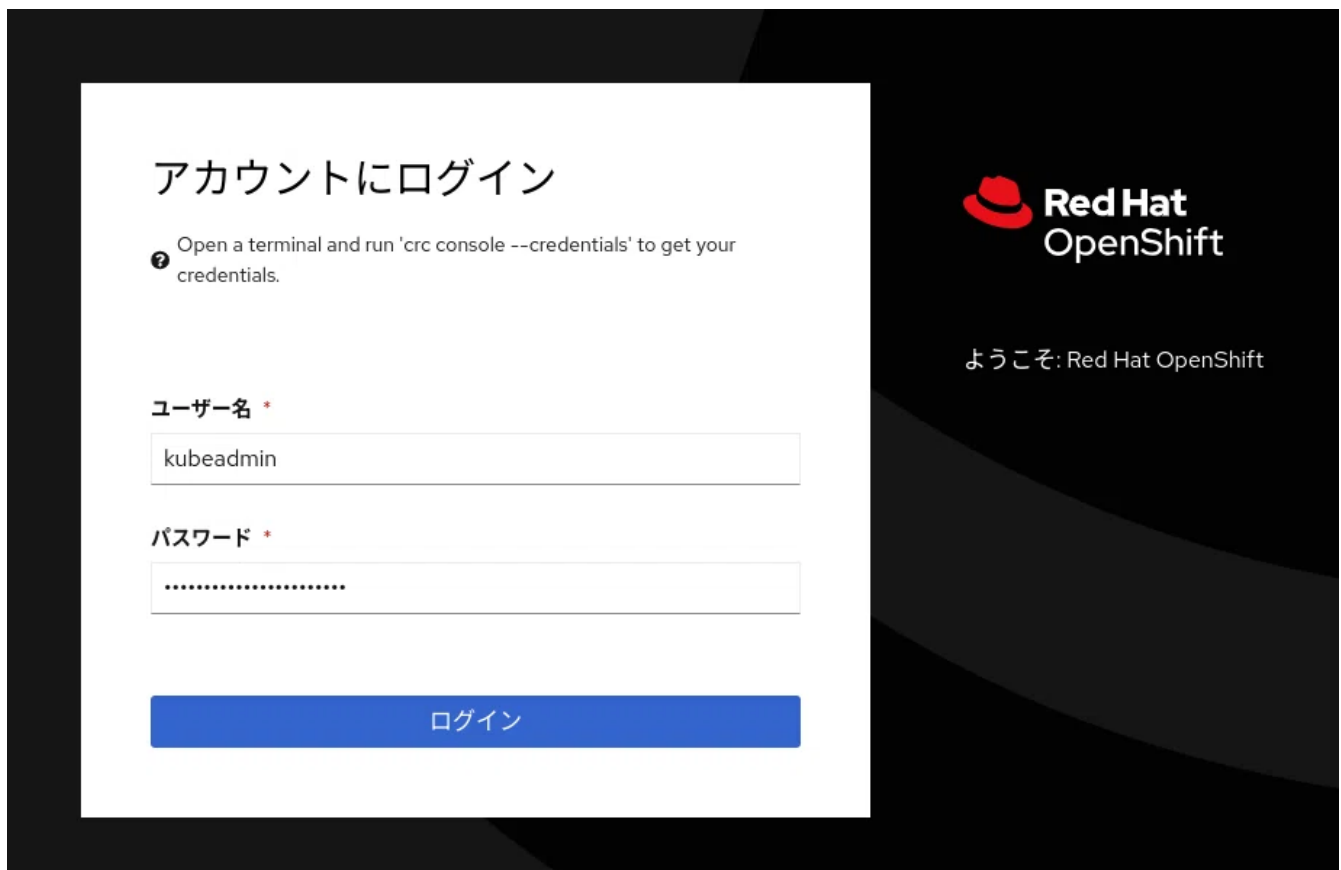


Figure 4. Webコンソールログイン

注意

OpenShift Container Platformインストーラーは、kubeadminという初期管理者アカウントを作成します。OpenShift Localもこの点では同じです。このkubeadminというアカウントはcluster-adminというクラスターの全権限をもったアカウントです。kubeadminは一時的なアカウントです。普通の運用では、インストール後にkubeadminを使って本当の管理者アカウントを作成し、管理作業はそのアカウントで実施するようにします。kubeadminは本当の管理者アカウントを作成したあとは削除します。

ログインが成功するとOpenShiftの管理コンソールが開きます。OpenShift Localではモニタリングの機能

が含まれていないので、CPU使用率のようなリソースの使用状況のグラフやアラート機能は使用できません。

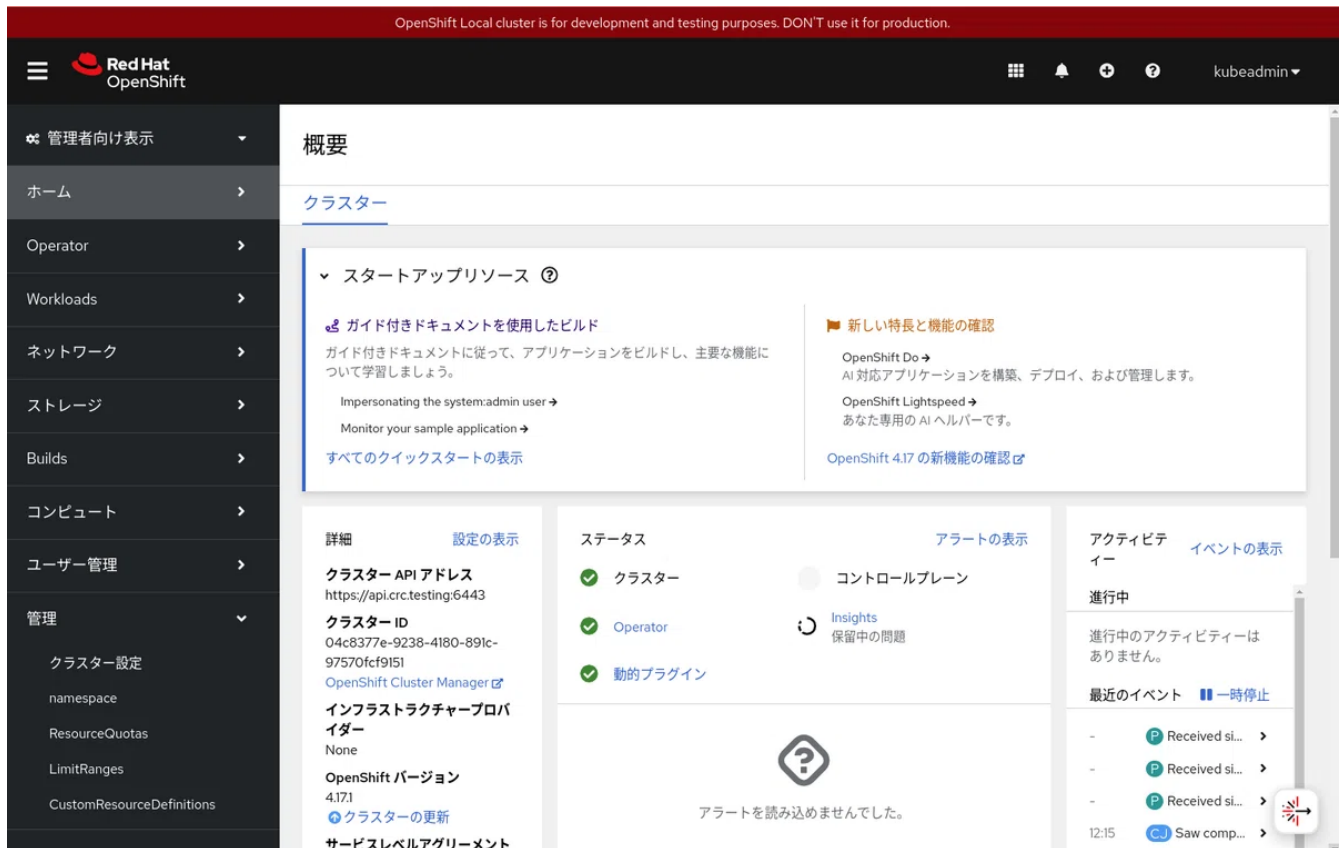


Figure 5. Webコンソールホーム

ヒント | OpenShift Localのモニタリング機能を有効にするには、`$ crc config set enable-cluster-monitoring true`を設定します。

6.3. Webコンソールに開発者としてログイン

OpenShift Localに管理者としてログインするには、developerというユーザー名でログインします。パスワードもdeveloperです。

7. CLIコマンドの使い方

7.1. ocコマンドの準備

oc コマンドは、OpenShiftクラスターにアクセスするCLIです。
OpenShift Localでは、以下のコマンドでocコマンドが使えるようになります

```
$ crc oc-env
export PATH="/home/student/.crc/bin/oc:$PATH"
# Run this command to configure your shell:
# eval $(crc oc-env)
```

このコマンドがやっていることは、ocコマンド~/crc/bin/ocを一時的にPATHに通しているだけです。

7.2. ocコマンドで管理者としてログイン

oc login コマンドを使ってOpenShiftクラスターにkubeadminでログインします。kubeadminのパスワードはcrc startコマンドのログに出力されます。

ヒント	<p>crc console --credentials コマンドを使えば、ocコマンドを使ったログイン方法を教えてください。</p> <pre>\$ crc console --credentials</pre> <p>To login as a regular user, run 'oc login -u developer -p developer https://api.crc.testing:6443'.</p> <p>To login as an admin, run 'oc login -u kubeadmin -p XXXXXXXX https://api.crc.testing:6443'</p>
-----	--

8. 管理者向け機能確認

ここはOpenShift Localの内部に興味の管理者の方のみ読んでください。以下のコマンドはkubeadminでログインが必要になります。

8.1. Clusterversionを調べる

oc get clusterversion コマンドでOpenShiftのバージョンを調べることができます。

```
$ oc get clusterversion
NAME      VERSION  AVAILABLE  PROGRESSING  SINCE   STATUS
version   4.17.1   True       False        26d    Cluster version is 4.17.1
```

8.2. Cluster Operatorを調べる

OpenShiftの内部サービスは、Cluster Operatorと呼ばれるもので管理されています。**oc get co** コマンドを実行すると、各Operatorのバージョンや状態がわかります。AVAILABLEの列がすべてTrueになっていることを確認します。

```
$ oc get co
NAME                                VERSION  AVAILABLE  PROGRESSING
DEGRADED SINCE  MESSAGE
authentication                       4.17.1   True       False       False
13m
config-operator                     4.17.1   True       False       False
26d
console                             4.17.1   True       False       False
13m
control-plane-machine-set            4.17.1   True       False       False
26d
dns                                  4.17.1   True       False       False
13m
etcd                                 4.17.1   True       False       False
26d
image-registry                       4.17.1   True       False       False
32m
ingress                             4.17.1   True       False       False
26d
kube-apiserver                       4.17.1   True       False       False
26d
kube-controller-manager              4.17.1   True       False       False
26d
kube-scheduler                       4.17.1   True       False       False
26d
kube-storage-version-migrator        4.17.1   True       False       False
32m
machine-api                          4.17.1   True       False       False
```


26d	machine-approver	4.17.1	True	False	False
26d	machine-config	4.17.1	True	False	False
26d	marketplace	4.17.1	True	False	False
26d	network	4.17.1	True	False	False
26d	openshift-apiserver	4.17.1	True	False	False
13m	openshift-controller-manager	4.17.1	True	False	False
13m	openshift-samples	4.17.1	True	False	False
26d	operator-lifecycle-manager	4.17.1	True	False	False
26d	operator-lifecycle-manager-catalog	4.17.1	True	False	False
26d	operator-lifecycle-manager-packageserver	4.17.1	True	False	False
14m	service-ca	4.17.1	True	False	False
26d					

8.3. Nodeの状態を調べる

oc get node コマンドで、OpenShift Localはシングルノードであること、ステータスがReadyであることを確認します。

```
$ oc get node
NAME      STATUS    ROLES                                AGE    VERSION
crc       Ready     control-plane,master,worker         26d    v1.30.4
```

oc get adm node-logs コマンドでノードのログを見ることができます。ここではkubeletのログを覗いてみます。

```
$ oc adm node-logs crc --unit kubelet | tail -5
Nov 16 10:02:39.473751 crc kubenswrapper[4221]: I1116 10:02:39.473438    4221
kubelet_getters.go:218] "Pod status updated" pod="openshift-etcd/etcd-crc"
status="Running"
Nov 16 10:02:39.474409 crc kubenswrapper[4221]: I1116 10:02:39.473786    4221
kubelet_getters.go:218] "Pod status updated" pod="openshift-kube-controller-
manager/kube-controller-manager-crc" status="Running"
Nov 16 10:02:39.474409 crc kubenswrapper[4221]: I1116 10:02:39.473812    4221
kubelet_getters.go:218] "Pod status updated" pod="openshift-machine-config-
operator/kube-rbac-proxy-crio-crc" status="Running"
Nov 16 10:02:39.474409 crc kubenswrapper[4221]: I1116 10:02:39.473849    4221
kubelet_getters.go:218] "Pod status updated" pod="openshift-kube-scheduler/openshift-
```

```
kube-scheduler-crc" status="Running"
Nov 16 10:02:39.474409 crc kubenswrapper[4221]: I1116 10:02:39.473864    4221
kubelet_getters.go:218] "Pod status updated" pod="openshift-kube-apiserver/kube-
apiserver-crc" status="Running"
```

8.4. モニタリングを有効にしてみる

crc config setでモニタリングを有効にすることができます。モニタリングを有効にすることで、メトリクスを収集し、それらの値をもとにグラフ表示をしたりアラートを出したりすることができます。

```
$ crc config set enable-cluster-monitoring true
Successfully configured enable-cluster-monitoring to true
```

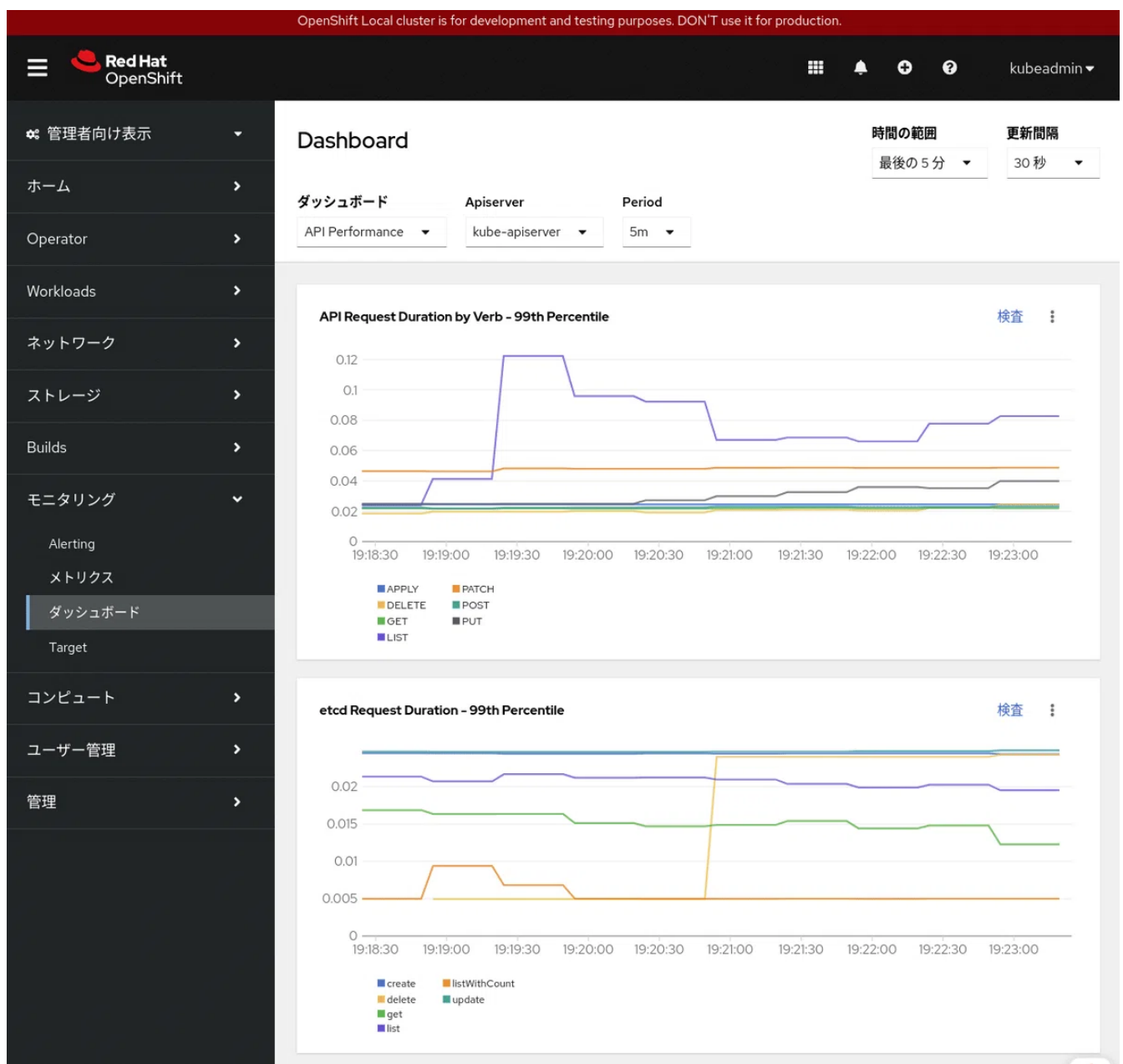


Figure 6. モニタリングダッシュボード

oc adm top node コマンドを使えば、メトリクスから得たリソースの使用状況を表示することができます

ます。

```
$ oc adm top node
NAME      CPU(cores)   CPU%    MEMORY(bytes)  MEMORY%
crc       869m         14%     10097Mi        49%
```

モニタリングを有効にしてOpenShift Localを再起動してみたところ、起動時間が **10分** になりました(モニタリング無効のときは3分)。開発者の方は、モニタリング機能をデフォルトのfalseのままにしておいた方が快適だと思います。

9. おわりに

OpenShift Localのインストール手順とよく使うコマンドを紹介しました。OpenShift Localは、開発者が使う環境というのが想定されてはいますが、kubeadminのアカウントが用意されているので、管理者がOpenShiftの学習用に使うのにも適していると思います。kubeadminでいろいろ試した結果、クラスターが壊れてしまったとしても、`crc delete`; `crc setup`をすればよいので気軽に実験ができますね。