



# Red Hat OpenShift Local 2.38

## 入门指南

使用 Red Hat OpenShift Local 的快速入门指南



# Red Hat OpenShift Local 2.38 入门指南

---

使用 Red Hat OpenShift Local 的快速入门指南

Fabrice Flore-Thebault

ffloreth@redhat.com

## 法律通告

Copyright © 2024 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<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

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

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> 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<sup>®</sup> 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.

## 摘要

本指南介绍了如何使用 Red Hat OpenShift Local 加快速度。包括说明和示例指南，执行从主机工作站(Microsoft Windows、macOS 或 Red Hat Enterprise Linux)使用 Red Hat OpenShift Container Platform 4 开发容器化应用程序的步骤。

---

# 目录

使开源包含更多 .....	3
<b>第 1 章 RED HAT OPENSIFT LOCAL 简介 .....</b>	<b>4</b>
1.1. 关于 RED HAT OPENSIFT LOCAL .....	4
1.2. 与生产环境的 OPENSIFT CONTAINER PLATFORM 安装的不同 .....	4
<b>第 2 章 安装 RED HAT OPENSIFT LOCAL .....</b>	<b>5</b>
2.1. 最低系统要求 .....	5
2.2. LINUX 所需的软件包 .....	6
2.3. 安装 RED HAT OPENSIFT LOCAL .....	6
2.4. 关于使用数据收集 .....	7
2.5. 配置使用数据收集 .....	7
2.6. 升级 RED HAT OPENSIFT LOCAL .....	8
<b>第 3 章 使用 RED HAT OPENSIFT LOCAL .....</b>	<b>9</b>
3.1. 关于预设置 .....	9
3.2. 设置 RED HAT OPENSIFT LOCAL .....	9
3.3. 启动实例 .....	10
3.4. 访问 OPENSIFT 集群 .....	10
3.5. 使用 ODO部署示例应用程序 .....	13
3.6. 停止实例 .....	14
3.7. 删除实例 .....	14
<b>第 4 章 配置 RED HAT OPENSIFT LOCAL .....</b>	<b>16</b>
4.1. 关于 RED HAT OPENSIFT LOCAL 配置 .....	16
4.2. 查看 RED HAT OPENSIFT LOCAL 配置 .....	16
4.3. 更改所选预设置 .....	16
4.4. 配置实例 .....	17
<b>第 5 章 网络 .....</b>	<b>18</b>
5.1. DNS 配置详情 .....	18
5.2. 保留的 IP 子网 .....	19
5.3. 在代理后面启动 RED HAT OPENSIFT LOCAL .....	19
5.4. 在远程服务器中设置 RED HAT OPENSIFT LOCAL .....	20
5.5. 连接到远程 RED HAT OPENSIFT LOCAL 实例 .....	21
<b>第 6 章 管理任务 .....</b>	<b>23</b>
6.1. 启动监控 .....	23
6.2. 启用覆盖 OPERATOR .....	23
<b>第 7 章 RED HAT OPENSIFT LOCAL 故障排除 .....</b>	<b>25</b>
7.1. 获取 OPENSIFT 集群的 SHELL 访问权限 .....	25
7.2. 过期的证书故障排除 .....	25
7.3. 捆绑包版本不匹配故障排除 .....	26
7.4. 未知问题故障排除 .....	26



---

## 使开源包含更多

红帽致力于替换我们的代码、文档和 Web 属性中存在问题的语言。我们从这四个术语开始：master、slave、黑名单和白名单。由于此项工作十分艰巨，这些更改将在即将推出的几个发行版本中逐步实施。详情请查看 [CTO Chris Wright 的信息](#)。

# 第 1 章 RED HAT OPENSIFT LOCAL 简介

## 1.1. 关于 RED HAT OPENSIFT LOCAL

Red Hat OpenShift Local 将最小的 OpenShift Container Platform 4 集群引入本地计算机。此运行时为开发和测试目的提供最小的环境。Red Hat OpenShift Local 主要用于在开发人员桌面上运行。对于其他 OpenShift Container Platform 用例，如无头或多开发人员设置，请使用 [完整的 OpenShift 安装程序](#)。

如需了解 [OpenShift Container Platform](#) 的完整介绍，请参阅 [OpenShift Container Platform 文档](#)。

Red Hat OpenShift Local 包括 **crc** 命令行界面(CLI)，用于使用所需的容器运行时与 Red Hat OpenShift Local 实例交互。

## 1.2. 与生产环境的 OPENSIFT CONTAINER PLATFORM 安装的不同

OpenShift 预设置 Red Hat OpenShift Local 提供常规 OpenShift Container Platform 安装，有以下显著区别：

- OpenShift Container Platform 集群是临时的，不适用于生产环境。
- Red Hat OpenShift Local 没有到较新的 OpenShift Container Platform 版本的升级路径。升级 OpenShift Container Platform 版本可能会导致很难重现的问题。
- 它使用单一节点，它的行为如同 control plane 和 worker 节点。
- 它默认禁用 Cluster Monitoring Operator。这个禁用的 Operator 会导致 web 控制台的对应部分无法正常工作。
- 它禁用以下 Operator：
  - Cloud Credential Operator。
  - Cluster Autoscaler Operator。
  - Cloud Controller Manager Operator。
- OpenShift Container Platform 集群在称为 *实例的虚拟机* 中运行。这可能导致其他区别，特别是外部网络。

Red Hat OpenShift Local 提供的 OpenShift Container Platform 集群还包括以下非定制集群设置。这些设置不应修改：

- prefixed **crc.testing** 域。
- 用于内部集群通信的地址范围。
  - 集群使用 172 地址范围。例如，这可能会导致代理在同一地址空间中运行。



## 第 2 章 安装 RED HAT OPENSIFT LOCAL

### 2.1. 最低系统要求

Red Hat OpenShift Local 有以下最低硬件和操作系统要求。

#### 2.1.1. 硬件要求

Red Hat OpenShift Local 在这些构架中被支持：

表 2.1. 预设置和架构兼容性

preset	AMD64	Intel 64	Apple silicon
OpenShift Container Platform	是	是	是
MicroShift	是	是	是

Red Hat OpenShift Local 不支持嵌套虚拟化。

根据所需的容器运行时，Red Hat OpenShift Local 需要以下系统资源：

#### 2.1.1.1. 对于 OpenShift Container Platform

- 4 个物理 CPU 内核
- 10.5 GB 可用内存
- 35 GB 存储空间

#### 2.1.1.2. 对于 MicroShift

- 2 个物理 CPU 内核
- 4 GB 可用内存
- 35 GB 存储空间



#### 注意

OpenShift Container Platform 和 MicroShift 预设需要这些最小资源在 Red Hat OpenShift Local 实例中运行。有些工作负载可能需要更多资源。要为 Red Hat OpenShift Local 实例分配更多资源，[请参阅配置实例](#)。

### 2.1.2. 操作系统要求

Red Hat OpenShift Local 需要以下支持的操作系统的最低版本：

#### 2.1.2.1. Microsoft Windows 的要求

- 在 Microsoft Windows 上，Red Hat OpenShift Local 需要完全更新 Microsoft Windows 10 或 Microsoft Windows 11。Red Hat OpenShift Local 不适用于早期版本的 Microsoft Windows。
- Red Hat OpenShift Local 不适用于 Microsoft Windows Home Edition。

### 2.1.2.2. macOS 的要求

- 在 macOS 上，Red Hat OpenShift Local 需要 macOS 12 Monterey 或更高版本。Red Hat OpenShift Local 不适用于早期版本的 macOS。

### 2.1.2.3. Linux 的要求

- 在 Linux 中，只有最新的两个 Red Hat Enterprise Linux/CentOS 8 和 9 次版本以及最新的两个稳定 Fedora 版本才支持 Red Hat OpenShift Local。
- 使用 Red Hat Enterprise Linux 时，运行 Red Hat OpenShift Local 的 [机器必须使用红帽客户门户网站 进行注册](#)。
- Ubuntu 18.04 LTS 或更高版本不支持 Debian 10 或更高版本，可能需要手动设置主机。
- [请参阅所需的软件包](#) 来安装 Linux 发行版本所需的软件包。

## 2.2. LINUX 所需的软件包

Red Hat OpenShift Local 需要 **libvirt** 和 **NetworkManager** 软件包在 Linux 上运行。下表查找用于为您的 Linux 发行版安装这些软件包的命令：

表 2.2. 按分发包安装命令

Linux Distribution	安装命令
Fedora/Red Hat Enterprise Linux/CentOS	<code>sudo dnf install NetworkManager</code>
Debian/Ubuntu	<code>sudo apt install qemu-kvm libvirt-daemon libvirt-daemon-system network-manager</code>

## 2.3. 安装 RED HAT OPENSIFT LOCAL

Red Hat OpenShift Local 作为 Red Hat Enterprise Linux 的可移植可执行文件提供。在 Microsoft Windows 和 macOS 上，可使用引导安装程序 Red Hat OpenShift Local 提供。

### 先决条件

- 您的主机必须满足最低系统要求。如需更多信息，请参阅 [最小系统要求](#)。

### 流程

1. 为您的平台下载 [最新版本的 Red Hat OpenShift Local](#)。
2. 在 Microsoft Windows 上，提取存档的内容。
3. 在 macOS 或 Microsoft Windows 上，运行引导安装程序并按照说明进行操作。



### 注意

在 Microsoft Windows 上，您必须将 Red Hat OpenShift Local 安装到本地 **C:** 驱动器。您无法从网络驱动器运行 Red Hat OpenShift Local。

在 Red Hat Enterprise Linux 中，假设归档位于 **~/Downloads** 目录中，请按照以下步骤执行：

- a. 提取存档内容：

```
$ cd ~/Downloads
$ tar xvf crc-linux-amd64.tar.xz
```

- b. 如果 **~/bin** 目录不存在，并将 **crc** 可执行文件复制到其中：

```
$ mkdir -p ~/bin
$ cp ~/Downloads/crc-linux-*-amd64/crc ~/bin
```

- c. 将 **~/bin** 目录添加到您的 **\$PATH** 中：

```
$ export PATH=$PATH:$HOME/bin
$ echo 'export PATH=$PATH:$HOME/bin' >> ~/.bashrc
```

## 2.4. 关于使用数据收集

Red Hat OpenShift Local 会提示您使用可选的匿名使用数据收集来协助开发。不会收集个人识别的信息。可以随时授予或撤销对使用数据收集的同意。

### 其他资源

- 有关收集数据的更多信息，请参阅 Red Hat [Telemetry 数据收集通知](#)。
- 要授予或撤销使用数据收集的同意，请参阅 [配置使用数据收集](#)。

## 2.5. 配置使用数据收集

您可以随时使用以下配置命令授予或撤销使用数据收集的同意。



### 注意

对遥测同意的更改不会修改正在运行的实例。每次运行 **crc start** 命令时，这些更改将生效。

### 流程

- 要手动启用遥测功能，请运行以下命令：

```
$ crc config set consent-telemetry yes
```

- 要手动禁用遥测功能，请运行以下命令：

```
$ crc config set consent-telemetry no
```

## 其他资源

- 有关收集的数据的更多信息，请参阅 Red Hat [Telemetry 数据收集通知](#)。

## 2.6. 升级 RED HAT OPENSIFT LOCAL

较新版本的 Red Hat OpenShift Local 可执行文件需要手动设置，以防止与早期版本不兼容。

### 流程

1. 下载最新版本的 Red Hat OpenShift Local。
2. 删除现有的 Red Hat OpenShift Local 实例：

```
$ crc delete
```



#### 警告

**crc delete** 命令会导致 Red Hat OpenShift Local 实例中存储的数据丢失。在运行此命令前，保存存储在实例中的需要的信息。

3. 将之前的 **crc** 可执行文件替换为最新版本的 **crc** 可执行文件。通过检查其版本来验证新的 **crc** 可执行文件是否正在使用：

```
$ crc version
```

4. 设置新的 Red Hat OpenShift Local 发行版本：

```
$ crc setup
```

5. 启动新的 Red Hat OpenShift Local 实例：

```
$ crc start
```

## 第 3 章 使用 RED HAT OPENSIFT LOCAL

### 3.1. 关于预设置

Red Hat OpenShift Local presets 代表受管容器运行时，以及实例运行它所需的系统资源的低限。Red Hat OpenShift Local 为以下提供预设：

#### openshift

最小的、预先配置的 OpenShift Container Platform 4.15 集群。

#### microshift

MicroShift.

在 Microsoft Windows 和 macOS 上，Red Hat OpenShift Local 指南安装程序会提示您输入所需的预设置。在 Linux 上，默认选择 OpenShift Container Platform 预设置。在运行 **crc setup** 命令前，您可以使用 **crc config** 命令更改此选择。次只能有一个预设置处于活动状态。

#### 其他资源

- 有关每个预设置的最低系统要求的更多信息，请参阅 [最小系统要求](#)。
- 有关更改所选预设置的更多信息，请参阅 [更改所选预设置](#)。

### 3.2. 设置 RED HAT OPENSIFT LOCAL

**crc setup** 命令执行操作，为 Red Hat OpenShift Local 实例设置主机机器的环境。

如果不存在，**crc setup** 命令创建 `~/.crc` 目录。



#### 警告

如果您要设置新版本，请在设置新的 Red Hat OpenShift Local 版本前捕获对实例所做的任何更改。

#### 先决条件

- 在 Linux 或 macOS 中，请确保您的用户帐户有权使用 **sudo** 命令。在 Microsoft Windows 上，请确保您的用户帐户可以提升到管理员特权。



#### 注意

不要以 **root** 用户身份或管理员运行 **crc** 可执行文件。始终使用您的用户帐户运行 **crc** 可执行文件。

#### 流程

1. 为 Red Hat OpenShift Local 设置主机机器：

```
$ crc setup
```

■

## 其他资源

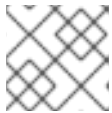
- 有关可用容器运行时预设置的更多信息，[请参阅关于预设。](#)

## 3.3. 启动实例

**crc start** 命令启动 Red Hat OpenShift Local 实例并配置的容器运行时。

### 先决条件

- 为了避免与网络相关的问题，请确保您没有连接到 VPN，且您的网络连接是可靠的。
- 您可以使用 **crc setup** 命令设置主机机器。如需更多信息，[请参阅设置 Red Hat OpenShift Local。](#)
- 在 Microsoft Windows 上，请确保您的用户帐户可以提升到管理员特权。
- 对于 OpenShift 预设置，请确保具有有效的 OpenShift 用户 pull secret。从 Red Hat [Hybrid Cloud Console](#) 上的 [Red Hat OpenShift Local 页面](#) 的 Pull Secret 部分复制或下载 pull secret。



### 注意

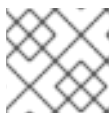
访问用户 pull secret 需要红帽帐户。

### 流程

1. 启动 Red Hat OpenShift Local 实例：

```
$ crc start
```

2. 对于 OpenShift 预设置，在提示时提供您的用户 pull secret。



### 注意

在提供请求前，集群至少需要 4 分钟才能启动必要的容器和 Operator。

## 其他资源

- 要更改分配给实例的默认资源，[请参阅配置实例。](#)
- 如果您在 **crc start** 中看到错误，请参阅 [Red Hat OpenShift Local 故障排除部分](#) 以了解潜在的解决方案。

## 3.4. 访问 OPENSIFT 集群

使用 OpenShift Container Platform Web 控制台或 OpenShift CLI (**oc**) 访问在 Red Hat OpenShift Local 实例中运行的 OpenShift Container Platform 集群。

### 3.4.1. 访问 OpenShift Web 控制台

使用您的 Web 浏览器访问 OpenShift Container Platform Web 控制台。

使用 **kubeadmin** 或 **developer** 用户访问集群。使用 **developer** 用户创建项目或 OpenShift 应用以及应用部署。使用 **kubeadmin** 用户仅用于管理任务，如创建新用户或设置角色。

### 先决条件

- Red Hat OpenShift Local 被配置为使用 OpenShift 预设置。如需更多信息，请参阅 [更改所选预设置](#)。
- 正在运行的 Red Hat OpenShift Local 实例。如需更多信息，请参阅 [启动实例](#)。

### 流程

1. 要使用默认 Web 浏览器访问 OpenShift Container Platform Web 控制台，请运行以下命令：

```
$ crc console
```

2. 以 **developer** 用户身份登录，密码在 **crc start** 命令的输出中打印。您还可以通过运行以下命令来查看 **developer** 和 **kubeadmin** 用户的密码：

```
$ crc console --credentials
```

如果您无法访问由 Red Hat OpenShift Local 管理的 OpenShift Container Platform 集群，请参阅对 Red Hat OpenShift Local 进行故障排除。

### 其他资源

- [OpenShift Container Platform 文档](#) 涵盖了创建项目和应用程序。

## 3.4.2. 使用 OpenShift CLI 访问 OpenShift 集群

使用 OpenShift CLI (**oc**)访问由 Red Hat OpenShift Local 管理的 OpenShift Container Platform 集群。

### 先决条件

- Red Hat OpenShift Local 被配置为使用 OpenShift 预设置。如需更多信息，请参阅 [更改所选预设置](#)。
- 正在运行的 Red Hat OpenShift Local 实例。如需更多信息，请参阅 [启动实例](#)。

### 流程

1. 运行 **crc oc-env** 命令，以打印将缓存的 **oc** 可执行文件添加到 **\$PATH** 所需的命令：

```
$ crc oc-env
```

2. 运行打印的命令。

3. 以 **developer** 用户登录：

```
$ oc login -u developer https://api.crc.testing:6443
```



### 注意

`crc start` 命令显示 **developer** 用户的密码。您可以通过运行 `crc console --credentials` 命令查看它。

4. 现在，您可以使用 **oc** 与 OpenShift Container Platform 集群交互。例如，要验证 OpenShift Container Platform 集群 Operator 是否可用，请以 **kubeadmin** 用户身份登录并运行以下命令：

```
$ oc config use-context crc-admin
$ oc whoami
kubeadmin
$ oc get co
```



### 注意

Red Hat OpenShift Local 默认禁用 Cluster Monitoring Operator。

如果您无法访问由 Red Hat OpenShift Local 管理的 OpenShift Container Platform 集群，请参阅对 Red Hat OpenShift Local 进行故障排除。

### 其他资源

- [OpenShift Container Platform 文档](#) 涵盖了创建项目和应用程序。

### 3.4.3. 访问内部 OpenShift registry

在 Red Hat OpenShift Local 实例中运行的 OpenShift Container Platform 集群默认包含一个内部容器镜像 registry。此内部容器镜像 registry 可用作本地开发容器镜像的发布目标。要访问内部 OpenShift Container Platform registry，请按照以下步骤操作。

### 先决条件

- Red Hat OpenShift Local 被配置为使用 OpenShift 预设置。如需更多信息，请参阅 [更改所选预设置](#)。
- 正在运行的 Red Hat OpenShift Local 实例。如需更多信息，请参阅 [启动实例](#)。
- 正常工作的 OpenShift CLI (**oc**) 命令。如需更多信息，请参阅使用 [OpenShift CLI 访问 OpenShift 集群](#)。

### 流程

1. 检查哪个用户登录到集群：

```
$ oc whoami
```



### 注意

出于演示目的，假定当前用户为 **kubeadmin**。

2. 以该用户身份登录 registry，及其令牌：



```
$ oc registry login --insecure=true
```

3. 创建一个新项目

```
$ oc new-project demo
```

4. 镜像示例容器镜像：

```
$ oc image mirror registry.access.redhat.com/ubi8/ubi:latest=default-route-openshift-image-registry.apps-crc.testing/demo/ubi8:latest --insecure=true --filter-by-os=linux/amd64
```

5. 获取镜像流并验证是否已列出推送的镜像：

```
$ oc get is
```

6. 在镜像流中启用镜像查找：

```
$ oc set image-lookup ubi8
```

此设置允许镜像流作为镜像的来源，而无需向内部 registry 提供完整的 URL。

7. 使用最近推送的镜像创建 pod：

```
$ oc run demo --image=ubi8 --command -- sleep 600s
```

### 3.5. 使用 `odo` 部署示例应用程序

您可以使用 `odo` 通过命令行创建 OpenShift 项目和应用程序。此流程将示例应用程序部署到在 Red Hat OpenShift Local 实例中运行的 OpenShift Container Platform 集群。

#### 先决条件

- 已安装了 `odo`。如需更多信息，请参阅 [odo 文档中的安装 odo](#)。
- Red Hat OpenShift Local 被配置为使用 OpenShift 预设置。如需更多信息，请参阅 [更改所选预设置](#)。
- Red Hat OpenShift Local 实例正在运行。如需更多信息，请参阅 [启动实例](#)。

#### 流程

1. 以 `developer` 用户身份登录由 Red Hat OpenShift Local 管理的正在运行的 OpenShift Container Platform 集群：

```
$ odo login -u developer -p developer
```

2. 为应用程序创建一个项目：

```
$ odo project create sample-app
```

3. 为您的组件创建一个新目录：

```
$ mkdir sample-app  
$ cd sample-app
```

4. 克隆一个 Node.js 应用程序示例：

```
$ git clone https://github.com/openshift/nodejs-ex  
$ cd nodejs-ex
```

5. 在应用程序中添加 **nodejs** 组件：

```
$ odo create nodejs
```

6. 创建 URL 并在本地配置文件中添加条目：

```
$ odo url create --port 8080
```

7. 推送更改：

```
$ odo push
```

现在，您的组件可以通过可访问的 URL 部署到集群中。

8. 列出 URL 并检查组件所需的 URL：

```
$ odo url list
```

9. 使用生成的 URL 查看部署的应用程序。

## 其他资源

- 有关使用 **odo** 的更多信息，请参阅 [odo 文档](#)。

## 3.6. 停止实例

**crc stop** 命令停止正在运行的 Red Hat OpenShift Local 实例和容器运行时。在关闭集群时，停止过程需要几分钟时间。

### 流程

- 停止 Red Hat OpenShift Local 实例和容器运行时：

```
$ crc stop
```

## 3.7. 删除实例

**crc delete** 命令删除现有的 Red Hat OpenShift Local 实例。

### 流程

- 删除 Red Hat OpenShift Local 实例：

---

## \$ crc delete



### 警告

**crc delete** 命令会导致 Red Hat OpenShift Local 实例中存储的数据丢失。在运行此命令前，保存存储在实例中的需要的信息。

## 第 4 章 配置 RED HAT OPENSIFT LOCAL

### 4.1. 关于 RED HAT OPENSIFT LOCAL 配置

使用 **crc config** 命令配置 **crc** 可执行文件和 Red Hat OpenShift Local 实例。**crc config** 命令需要一个子命令来操作配置。可用的子命令包括 **get**、**set**、**unset** 和 **view**。**get**、**set** 和 **unset** 子命令对命名的属性进行操作。运行 **crc config --help** 命令来列出可用的属性。

您还可以使用 **crc config** 命令配置 **crc start** 和 **crc setup** 命令的启动检查行为。默认情况下，启动检查会在不满足它们条件时报告错误和停止执行。将以 **skip-check** 开头的属性值设置为 **true** 以跳过检查。

### 4.2. 查看 RED HAT OPENSIFT LOCAL 配置

Red Hat OpenShift Local 可执行文件提供命令来查看可配置属性和当前的 Red Hat OpenShift Local 配置。

#### 流程

- 查看可用的可配置属性：

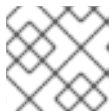
```
$ crc config --help
```

- 查看可配置属性的值：

```
$ crc config get <property>
```

- 查看完整的当前配置：

```
$ crc config view
```



#### 注意

如果配置由默认值组成，则 **crc config view** 命令不会返回任何信息。

### 4.3. 更改所选预设置

您可以选择所需的预设置来更改用于 Red Hat OpenShift Local 实例的容器运行时。

您可以使用命令行界面更改所选预设置。



#### 重要

您无法更改现有 Red Hat OpenShift Local 实例的预设置。只有在创建 Red Hat OpenShift Local 实例时，才会应用预设置更改。要启用预设置更改，您必须删除现有实例并启动新的实例。

#### 流程

- 从命令行更改所选预设置：

```
$ crc config set preset <name>
```

有效的预设置名称是：

表 4.1. 预设置名称

Name	preset
<b>openshift</b>	OpenShift Container Platform
<b>microshift</b>	MicroShift

## 其他资源

- 有关每个预设置的最低系统要求的更多信息，请参阅 [最小系统要求](#)。

## 4.4. 配置实例

使用 **cpus** 和 **memory** 属性，分别配置 Red Hat OpenShift Local 实例可用的默认 vCPU 数和内存量。

或者，也可以使用 **crc start** 命令中的 **- cpus** and **- memory** 标志来 分别分配 vCPU 和内存量。



### 重要

您无法更改正在运行的 Red Hat OpenShift Local 实例的配置。要启用配置更改，您必须停止正在运行的实例并再次启动它。

## 流程

- 配置实例可用的 vCPU 数量：

```
$ crc config set cpus <number>
```

**cpus** 属性的默认值为 **4**。要分配的 vCPU 数量必须大于或等于默认值。

- 使用所需 vCPU 数量启动实例：

```
$ crc start --cpus <number>
```

- 配置实例可用内存：

```
$ crc config set memory <number-in-mib>
```



### 注意

可用内存的值以兆字节(MiB)为单位。一个 gibibyte (GiB)内存等于 1024 MiB。

**memory** 属性的默认值为 **10752**。要分配的内存量必须大于或等于默认值。

- 使用所需内存量启动实例：

```
$ crc start --memory <number-in-mib>
```

## 第 5 章 网络

### 5.1. DNS 配置详情

#### 5.1.1. 常规 DNS 设置

由 Red Hat OpenShift Local 管理的 OpenShift Container Platform 集群使用 2 个 DNS 域名、**crc.testing** 和 **apps-crc.testing**。**crc.testing** 域用于 OpenShift Container Platform 核心服务。**apps-crc.testing** 域用于访问集群中部署的 OpenShift 应用。

例如，OpenShift Container Platform API 服务器会公开为 **api.crc.testing**，而 OpenShift Container Platform 控制台则作为 **console-openshift-console.apps-crc.testing** 访问。这些 DNS 域由在 Red Hat OpenShift Local 实例中运行的 **dnsmasq** DNS 容器提供。

**crc setup** 命令会检测并调整您的系统 DNS 配置，以便它能够解析这些域。完成了其他检查，以验证在运行 **crc start** 时正确配置了 DNS。

#### 5.1.2. Linux 中的 DNS

在 Linux 上，根据您的发行版本，Red Hat OpenShift Local 需要以下 DNS 配置：

##### 5.1.2.1. NetworkManager + systemd-resolved

默认情况下，在 Fedora 33 或更新版本和 Ubuntu 桌面版本中使用此配置。

- Red Hat OpenShift Local 期望 NetworkManager 管理网络。
- Red Hat OpenShift Local 配置 **systemd-resolved**，将测试域的请求转发到 **192.168.130.11** DNS 服务器。**192.168.130.11** 是 Red Hat OpenShift Local 实例的 IP。
- **systemd-resolved** 配置是使用 `/etc/NetworkManager/dispatcher.d/99-crc.sh` 中的 NetworkManager 分配程序脚本完成的：

```
#!/bin/sh

export LC_ALL=C

systemd-resolve --interface crc --set-dns 192.168.130.11 --set-domain ~testing

exit 0
```



#### 注意

**systemd-resolved** 还在 Red Hat Enterprise Linux 和 CentOS 8.3 中作为不受支持的技术预览提供。[将主机配置为使用 systemd-resolved](#) 后，停止任何正在运行的集群并重新运行 **crc setup**。

##### 5.1.2.2. NetworkManager + dnsmasq

默认情况下，Red Hat Enterprise Linux 和 CentOS 上的 Fedora 32 或更早版本使用此配置。

- Red Hat OpenShift Local 期望 NetworkManager 管理网络。

- NetworkManager 使用带有 `/etc/NetworkManager/conf.d/crc-nm-dnsmasq.conf` 配置文件的 dnsmasq。
- 此 dnsmasq 实例的配置文件是 `/etc/NetworkManager/dnsmasq.d/crc.conf` :

```
server=/crc.testing/192.168.130.11
server=/apps-crc.testing/192.168.130.11
```

- NetworkManager dnsmasq 实例将对 `crc.testing` 和 `apps-crc.testing` 域的请求转发到 **192.168.130.11** DNS 服务器。

## 5.2. 保留的 IP 子网

由 Red Hat OpenShift Local 管理的 OpenShift Container Platform 集群为内部使用保留 IP 子网，这不应与您的主机网络冲突。确保以下 IP 子网可供使用：

### 保留的 IP 子网

- **10.217.0.0/22**
- **10.217.4.0/23**
- **192.168.126.0/24**

此外，主机虚拟机监控程序可能会保留另一个 IP 子网，具体取决于主机操作系统。macOS 和 Microsoft Windows 上没有保留任何额外的子网。Linux 的额外保留子网是 **192.168.130.0/24**。

## 5.3. 在代理后面启动 RED HAT OPENSIFT LOCAL

您可以使用环境变量或可配置属性在定义的代理后面启动 Red Hat OpenShift Local。



### 注意

OpenShift Container Platform 不支持 SOCKS 代理。

### 先决条件

- 如果没有使用 `crc oc-env`，在与集群交互时，请将 `.testing` 域导出为 `no_proxy` 环境变量的一部分。嵌入的 `oc` 可执行文件不需要手动设置。有关使用嵌入的 `oc` 可执行文件的更多信息，请参阅使用 [OpenShift CLI](#) 访问 [OpenShift 集群](#)。

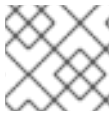
### 流程

1. 使用 `http_proxy` 和 `https_proxy` 环境变量或使用 `crc config set` 命令定义代理，如下所示：

```
$ crc config set http-proxy http://proxy.example.com:<port>
$ crc config set https-proxy http://proxy.example.com:<port>
$ crc config set no-proxy <comma-separated-no-proxy-entries>
```

2. 如果代理使用自定义 CA 证书文件，请按如下所示设置：

```
$ crc config set proxy-ca-file <path-to-custom-ca-file>
```



## 注意

Red Hat OpenShift Local 配置中设置的与代理相关的值优先于使用环境变量设置的值。

## 5.4. 在远程服务器中设置 RED HAT OPENSIFT LOCAL

配置远程服务器以运行由 Red Hat OpenShift Local 提供的 OpenShift Container Platform 集群。

这个过程假设使用 Red Hat Enterprise Linux、Fedora 或 CentOS 服务器。在远程服务器中运行此流程中的每个命令。



### 警告

仅在本地网络中执行此步骤。在互联网上公开不安全的服务器存在许多安全隐患。

### 先决条件

- Red Hat OpenShift Local 已安装并在远程服务器上设置。如需更多信息，请参阅[安装 Red Hat OpenShift Local](#) 和 [设置 Red Hat OpenShift Local](#)。
- Red Hat OpenShift Local 被配置为使用远程服务器上的 OpenShift 预设置。如需更多信息，请参阅 [更改所选预设置](#)。
- 您的用户帐户在远程服务器上具有 **sudo** 权限。

### 流程

1. 启动集群：

```
$ crc start
```

确保集群在此过程中保持运行。

2. 安装 **haproxy** 软件包和其他工具：

```
$ sudo dnf install haproxy /usr/sbin/semanage
```

3. 修改防火墙以允许与集群通信：

```
$ sudo systemctl enable --now firewalld
$ sudo firewall-cmd --add-service=http --permanent
$ sudo firewall-cmd --add-service=https --permanent
$ sudo firewall-cmd --add-service=kube-apiserver --permanent
$ sudo firewall-cmd --reload
```

4. 对于 SELinux，允许 HAProxy 侦听 TCP 端口 6443 在此端口上提供 **kube-apiserver**：

```
$ sudo semanage port -a -t http_port_t -p tcp 6443
```



5. 创建默认 **haproxy** 配置的备份：

```
$ sudo cp /etc/haproxy/haproxy.cfg{,.bak}
```

6. 配置 **haproxy** 以与集群搭配使用：

```
$ export CRC_IP=$(crc ip)
$ sudo tee /etc/haproxy/haproxy.cfg >/dev/null <<EOF
global
    log /dev/log local0

defaults
    balance roundrobin
    log global
    maxconn 100
    mode tcp
    timeout connect 5s
    timeout client 500s
    timeout server 500s

listen apps
    bind 0.0.0.0:80
    server crcvm $CRC_IP:80 check

listen apps_ssl
    bind 0.0.0.0:443
    server crcvm $CRC_IP:443 check

listen api
    bind 0.0.0.0:6443
    server crcvm $CRC_IP:6443 check
EOF
```

7. 启动 **haproxy** 服务：

```
$ sudo systemctl start haproxy
```

## 5.5. 连接到远程 RED HAT OPENSIFT LOCAL 实例

使用 **dnsmasq** 将客户端机器连接到运行由 Red Hat OpenShift Local 管理的 OpenShift Container Platform 集群的远程服务器。

这个过程假设使用 Red Hat Enterprise Linux、Fedora 或 CentOS 客户端。在客户端上运行此流程中的每个命令。



### 重要

连接到仅在本地网络上公开的服务器。

### 先决条件

- 为客户端设置要连接的远程服务器。如需更多信息，请参阅在 [远程服务器中设置 Red Hat OpenShift Local](#)。

- 您知道服务器的外部 IP 地址。
- 在客户端的 `$PATH` 中有 [最新的 OpenShift CLI \(oc\)](#)。

## 流程

1. 安装 `dnsmasq` 软件包：

```
$ sudo dnf install dnsmasq
```

2. 在 NetworkManager 中启用 `dnsmasq` 进行 DNS 解析：

```
$ sudo tee /etc/NetworkManager/conf.d/use-dnsmasq.conf &>/dev/null <<EOF
[main]
dns=dnsmasq
EOF
```

3. 将 Red Hat OpenShift Local 的 DNS 条目添加到 `dnsmasq` 配置中：

```
$ sudo tee /etc/NetworkManager/dnsmasq.d/external-crc.conf &>/dev/null <<EOF
address=/apps-crc.testing/SERVER_IP_ADDRESS
address=/api.crc.testing/SERVER_IP_ADDRESS
EOF
```



### 注意

注释掉 `/etc/NetworkManager/dnsmasq.d/crc.conf` 中的任何现有条目。这些条目通过运行 Red Hat OpenShift Local 的本地实例来创建，并将与远程集群的条目冲突。

4. 重新载入 NetworkManager 服务：

```
$ sudo systemctl reload NetworkManager
```

5. 使用 `oc` 以 `developer` 用户身份登录远程集群：

```
$ oc login -u developer -p developer https://api.crc.testing:6443
```

远程 OpenShift Container Platform Web 控制台位于 <https://console-openshift-console.apps-crc.testing>。

## 第 6 章 管理任务

### 6.1. 启动监控

Red Hat OpenShift Local 默认禁用集群监控，以确保 Red Hat OpenShift Local 可以在典型的笔记本中运行。监控负责在 [Red Hat Hybrid Cloud Console](#) 中列出您的集群。按照以下步骤为集群启用监控。

#### 先决条件

- 您必须为 Red Hat OpenShift Local 实例分配额外的内存。对于核心功能，建议使用 14 GiB 内存(14336)。增加的工作负载需要更多内存。如需更多信息，[请参阅配置实例](#)。

#### 流程

1. 将 `enable-cluster-monitoring` 可配置属性设置为 `true`：

```
$ crc config set enable-cluster-monitoring true
```

2. 启动实例：

```
$ crc start
```



#### 警告

无法禁用集群监控。要删除监控，请将 `enable-cluster-monitoring` 可配置属性设置为 `false`，并删除现有的 Red Hat OpenShift Local 实例。

### 6.2. 启用覆盖 OPERATOR

要验证 Red Hat OpenShift Local 是否可在典型的笔记本上运行，一些资源密集型服务会被默认禁用。可以通过从 Operator 覆盖列表中删除所需的 Operator 来启用这些服务。

#### 先决条件

- 正在运行的 Red Hat OpenShift Local 虚拟机和一个正常工作的 `oc` 命令。如需更多信息，[请参阅使用 oc 访问 OpenShift 集群](#)。
- 您必须以 `kubeadmin` 用户身份登录通过 `oc`。

#### 流程

1. 列出非受管 Operator，并记录所需 Operator 的数字索引：

- 在 Linux 或 macOS 中：

```
$ oc get clusterversion version -ojsonpath='{range .spec.overrides[*]}{.name}{"\n"}{end}' | nl -v 0
```

- 在使用 PowerShell 的 Microsoft Windows 上 :

```
PS> oc get clusterversion version -ojsonpath='{range .spec.overrides[*]}{.name}{"\n"}{end}' | % {$nl++; "t${$nl-1} `t $_" }; $nl=0
```

2. 使用标识的数字索引启动所需的 Operator :

```
$ oc patch clusterversion/version --type='json' -p ' [{"op": "remove",  
"path": "/spec/overrides/<unmanaged-operator-index>"} ]'  
clusterversion.config.openshift.io/version patched
```

## 第 7 章 RED HAT OPENSIFT LOCAL 故障排除



### 注意

Red Hat OpenShift Local 的目标是为开发和测试目的提供 OpenShift Container Platform 环境。在安装或使用特定 OpenShift 应用程序期间发生的问题超出了 Red Hat OpenShift Local 范围。向相关项目报告这些问题。

### 7.1. 获取 OPENSIFT 集群的 SHELL 访问权限

要访问集群以进行故障排除或调试目的，请按照以下步骤操作。



### 注意

常规使用不需要直接访问 OpenShift Container Platform 集群，强烈建议您这样做。

#### 先决条件

- 启用集群的 OpenShift CLI (**oc**)访问，并以 **kubeadmin** 用户身份登录。具体步骤请参阅使用 [OpenShift CLI 访问 OpenShift 集群](#)。

#### 流程

1. 运行 **oc get nodes** 命令以识别所需的节点。输出结果类似如下：

```
$ oc get nodes
NAME                STATUS ROLES      AGE  VERSION
crc-shdl4-master-0 Ready  master,worker  7d7h v1.14.6+7e13ab9a7
```

2. 运行 **oc debug nodes/<node> & gt;**，其中 <node> 是上一步中打印的节点的名称。

### 7.2. 过期的证书故障排除

每个发布的 **crc** 可执行文件中的 OpenShift Container Platform 系统捆绑包会在发布后 1 年过期。此过期是由于嵌入在 OpenShift Container Platform 集群中的证书造成的。**crc start** 命令在需要时触发自动证书续订过程。证书续订可最多添加五分钟到集群的开始时间。

要避免这个额外的启动时间，或者在证书续订过程中失败，请使用以下步骤：

#### 流程

要解决无法自动更新的过期的证书错误：

1. 下载最新的 [Red Hat OpenShift Local 版本](#)，并将 **crc** 可执行文件放在 **\$PATH** 中。
2. 使用 **crc delete** 命令删除带有证书错误的集群：

```
$ crc delete
```

**警告**

**crc delete** 命令会导致 Red Hat OpenShift Local 实例中存储的数据丢失。在运行此命令前，保存存储在实例中的需要的信息。

3. 设置新版本：

```
$ crc setup
```

4. 启动新实例：

```
$ crc start
```

### 7.3. 捆绑包版本不匹配故障排除

创建的 Red Hat OpenShift Local 实例包含捆绑包信息和实例数据。设置新的 Red Hat OpenShift Local 版本时，不会更新捆绑包信息和实例数据。由于之前实例数据中的自定义，此信息不会更新。这会在运行 **crc start** 命令时出现错误：

```
$ crc start
...
FATA Bundle 'crc_hyperkit_4.2.8.crcbundle' was requested, but the existing VM is using
'crc_hyperkit_4.2.2.crcbundle'
```

#### 流程

1. 在尝试启动实例前发出 **crc delete** 命令：

```
$ crc delete
```

**警告**

**crc delete** 命令会导致 Red Hat OpenShift Local 实例中存储的数据丢失。在运行此命令前，保存存储在实例中的需要的信息。

### 7.4. 未知问题故障排除

通过重启带有干净状态的 Red Hat OpenShift Local 来解决大多数问题。这涉及停止实例，删除它，恢复 **crc setup** 命令所做的更改，重新应用这些更改并重启实例。

#### 先决条件

- 您可以使用 **crc setup** 命令设置主机机器。如需更多信息，[请参阅设置 Red Hat OpenShift Local](#)。
- 您可以使用 **crc start** 命令启动 Red Hat OpenShift Local。如需更多信息，[请参阅 启动实例](#)。
- 您使用最新的 Red Hat OpenShift Local 版本。使用早于 Red Hat OpenShift Local 1.2.0 的版本可能会导致与过期的 x509 证书相关的错误。如需更多信息，[请参阅对过期的证书进行故障排除](#)。

## 流程

要排除 Red Hat OpenShift Local 的问题，请执行以下步骤：

1. 停止 Red Hat OpenShift Local 实例：

```
$ crc stop
```

2. 删除 Red Hat OpenShift Local 实例：

```
$ crc delete
```



### 警告

**crc delete** 命令会导致 Red Hat OpenShift Local 实例中存储的数据丢失。在运行此命令前，保存存储在实例中的需要的信息。

3. 从 **crc setup** 命令中清理剩余的更改：

```
$ crc cleanup
```



### 注意

**crc cleanup** 命令删除现有的 Red Hat OpenShift Local 实例，并将更改恢复到 **crc setup** 命令创建的 DNS 条目。

4. 设置主机机器以重新应用更改：

```
$ crc setup
```

5. 启动 Red Hat OpenShift Local 实例：

```
$ crc start
```



### 注意

在提供请求前，集群至少需要 4 分钟才能启动必要的容器和 Operator。

如果您的问题没有被这个过程解决，请执行以下步骤：

1. 为您遇到的问题 [搜索开放问题](#)。
2. 如果没有现有问题解决，[请创建一个](#) 问题，并将 `~/crc/crc.log` 文件附加到 创建的问题。`~/crc/crc.log` 文件的详细调试和故障排除信息，可帮助诊断您遇到的问题。