文章
jieliang liu · 十月 26, 2021 阅读大约需 21 分钟

IRIS 2021 技术文档 First Look 24 技术概要：容器中的 InterSystems 产品

本技术概要（First Look）通过重点概述和基本的实践示例，向您介绍在 InterSystems IRIS®数据平台上使用容器（Container）的基本原理。您将了解容器的目的、重要性和好处，以及 InterSystems 如何实现它们的具体细节。

有关 Docker 容器和 InterSystems IRIS 的完整文档，请参见 *Running InterSystems Products in Containers*（《*在容器中运行 InterSystems 产品*》），以及 *InterSystems Cloud Manager Guide*（《*InterSystems 云管理器指南*》）的 ICM Overview（ICM 概述）章节。Learn More About InterSystems IRIS in Containers（了解容器中的 InterSystems IRIS 的更多信息）部分提供了到其他参考资料的链接。

要浏览所有的技术概要（First Look），包括可以在 InterSystems IRIS 免费的评估实例上执行的那些，请参见 InterSystems First Looks（《InterSystems 技术概要》）。

1. 为什么是容器（Container）？

容器（Container）将应用程序打包成独立于平台、完全可移植的运行时解决方案（runtime solution），满足所有需求，但却隔离于任何依赖项（dependency）。具体来说，Docker 容器无处不在。因为所有主要的公共云基础设施即服务（Infrastructure as a Service，IaaS）供应商都支持 Docker，企业可以通过使用容器并让云供应商处理基础设施来降低成本和复杂性。

容器带来了以下所有的好处：

- 容器干净地划分代码和数据，提供完全分离的关注点，应用程序可轻松部署和升级。
- 容器是非常高效的；容器中的应用程序只包含运行它所需的元素（element），并使其能够访问所需的连接、服务和接口，而且容器作为单个进程运行，所占用的内存并不比任何其他可执行文件多。
- 容器支持应用程序在不同环境之间的清洁移动（clean movement）——例如，从开发到测试，再到生产——从而减少具有不同目标的部门之间的典型过程和管理冲突。开发人员可以专注于最新的代码和库（library），质量开发人员（quality developers）专注于测试和缺陷描述，运营工程师专注于整体解决方案的基础设施，包括网络、高可用性、数据持久性（data durability）等等。
- 容器提供了企业对业务和技术进行变革所需的敏捷性、灵活性和可重复性。容器将应用供应过程（包括构建阶段）与运行过程明确分开，企业可以采用统一的应用程序交付方法，包括更敏捷的交付方法（DevOps）和架构（微服务）。

这些优势使容器成为应用程序的天然构件块，推动应用程序交付和部署方法变得更简单、更快速、更可重复、更强大。

2. 容器中的 InterSystems IRIS

因为容器只打包运行容器化应用程序所需的元素，并在本地执行应用程序，所以它提供标准的、易于理解的应用程序配置、行为和访问。如果您对在 Linux 上运行的 InterSystems IRIS 有经验，那么基于 Linux 的 InterSystems IRIS 容器在什么物理、虚拟或云系统和操作系统平台上运行并不重要；您以相同的方式与它们进行交互，就像在 Linux 系统上运行传统的 InterSystems IRIS 实例一样。

下面介绍 InterSystems IRIS 在如何使用容器方面的不同。

- *InterSystems 提供的映像（InterSystems-provided images）*——*容器映像（container image）*是可执行包，而容器是映像（image）的运行时*实例（runtime instance）*——即映像（image）在执行时在内存中变成的实例。InterSystems 提供包含完整安装的 InterSystems IRIS 实例的映像（image），以及其他相关映像（image）。有关 InterSystems 映像（image）的更多信息，请参见 *Running InterSystems Products in Containers*（《*在容器中运行 InterSystems 产品*》）中的 Using InterSystems IRIS Images（使用 InterSystems IRIS 映像）。

在本文档的实践体验中，您将从 InterSystems 提供的 InterSystems IRIS 映像（image）中创建并启动一个容器。

- *Iris-main 程序（iris-main program）*——*iris-main* 程序使 InterSystems IRIS 和其他产品能够满足在 Docker 容器中运行的应用程序的要求。*入口点应用程序（entrypoint application）*，即容器启动时启动的主进程，需要阻塞（block）——即等待（wait），直到其工作完成，但启动 InterSystems IRIS 的命令并不作为一个阻塞进程（blocking process）运行。**iris-main** 程序通过启动 InterSystems IRIS 来解决这个问题，然后继续作为阻塞入口点应用程序（blocking entrypoint application）运行。有关 **iris-main** 的更多信息，请参见在 *Running InterSystems Products in Containers*（《*在容器中运行 InterSystems 产品*》）中的 The iris-main Program（iris-main 程序）。

该程序还提供了许多选项，以帮助调整 InterSystems IRIS 在容器中的行为；您将在本文档的实践体验中使用一些 **iris-main** 选项。

- *持久化 %SYS 功能（durable %SYS feature）*——因为容器化的应用程序与主机环境隔离，所以它不会写入持久化数据（persist data）；当容器被移除并被新的容器取代时，它在容器内写入的任何内容都会丢失。因此，容器化应用部署的一个重要方面是安排数据存储在容器之外，并提供给其他和未来的容器使用。

当 InterSystems IRIS 在容器中运行时，持久化 %SYS 功能（The durable %SYS feature）能够持久地存储特定于实例的数据——如用户定义、审计记录以及日志、日记和 WIJ 文件——允许一个实例跨越多个容器。例如，如果您使用持久化 %SYS 运行 InterSystems IRIS 容器，您可以通过停止原始容器（the original container）并运行使用旧容器创建的特定于实例的数据（instance-specific data）的新容器来升级实例。

您将在本文档的实践体验中探索持久化 %SYS （durable %SYS）功能。有关持久化 %SYS（durable %SYS）的详细信息，请参见 *Running InterSystems Products in Containers*（《*在容器中运行 InterSystems 产品*》）中的

Durable %SYS for Persistent Instance Data（持久化 %SYS 的持久化实例数据）。

提示： InterSystems 的容器映像符合开放容器标准（Open Container Initiative，OCI）规范，并在 Docker 企业版（Enterprise Edition）引擎上使用广泛流行的容器 Ubuntu 操作系统构建，该引擎完全支持 OCI 标准，并允许映像在 Docker Hub 注册表中认证和展示。因此，InterSystems 映像在任何基于 Linux 操作系统的符合 OCI 标准的运行时引擎上都得到支持，无论是在自有云还是在公共云中。

InterSystems 云管理器（InterSystems Cloud Manager，ICM）在其提供的云基础设施以及现有的虚拟和物理基础设施上提供 InterSystems IRIS 容器和其他容器的自动部署。有关使用 ICM 部署容器化的 InterSystems IRIS 实例的更多信息，请参见 *First Look：InterSystems Cloud Manager（《技术概要：InterSystems 云管理器》）* 和 *InterSystems Cloud Manager Guide*（*《InterSystems 云管理器指南》*）。

3. 下载 InterSystems IRIS 映像（Image）

要使来自 InterSystems 的 InterSystems IRIS 映像（image）可用于本实践，您必须将映像（image）下载到您正在使用的系统中。以下替代方案描述了您可以或可能可以使用的 InterSystems IRIS 映像（image）。

- 您可以使用来自 InterSystems 容器注册表（InterSystems Container Registry，ICR）的 InterSystems IRIS 社区版（Community Edition）映像（image），该映像（image）包含可从 InterSystems 获得的所有映像（image）的存储库，如 Using the InterSystems Container Registry（《使用 InterSystems 容器注册表》）中所述。您也可以从 Docker Store 的 InterSystems IRIS 数据平台页面下载社区版（Community Edition）映像（image）。

InterSystems IRIS 社区版（Community Edition）有一个免费的内置13个月的许可证（和一些功能限制）；如果您在这次实践中使用社区版（Community Edition），您就不需要提供下一步（Add the License Key to the External Storage Location（添加许可证密钥到外部存储位置））中所述的许可证密钥。更多信息，请参见 *Deploy and Explore InterSystems IRIS*（《*部署和探索 InterSystems IRIS*》）中的 Deploy InterSystems IRIS Community Edition on Your Own System（在您自己的系统上部署 InterSystems IRIS 社区版）。

注意： 另一个选择是在 GCP、AWS 或 Azure 上提供一个承载运行中的 InterSystems IRIS 社区版（Community Edition）容器的云节点；更多信息，请参见 *Deploy and Explore InterSystems IRIS*（《*部署和探索 InterSystems IRIS*》）中的 Deploy InterSystems IRIS Community Edition on a Cloud Node（在云节点上部署 InterSystems IRIS 社区版）。如果您在这个练习中使用社区版（Community Edition）的云节点，您可以跳过接下来的两个步骤，并跳转到 Change the Instance and Commit the Container as a New Image（更改实例并将容器作为新映像提交）。

- 如果您是 InterSystems 客户，您可以使用 InterSystems 容器注册表（InterSystems Container Registry，ICR）发布的 InterSystems IRIS 映像（image）。Using the InterSystems Container Registry（使用 InterSystems 容器注册表）列出了 IRC 提供的 InterSystems IRIS 映像（image），并解释了如何使用您的 WRC 凭证对注册表进行身份验证，以便您下载。
- 您的企业可能有一个私有映像（image）注册表，其中包括一个或多个 InterSystems IRIS 映像（image）。如果是这样，请获取您需要的映像（image）的注册表、存储库和标签的位置，以及访问所需的凭证。

当您确定了要下载的注册表和需要的凭证（如果有）后，请参见 *Running InterSystems Products in Containers*（《在*容器中运行 InterSystems 产品*》）中的 Downloadingthe InterSystems IRIS Image（下载 InterSystems IRIS 映像），了解下载 InterSystems IRIS 映像（image）的说明。

为了简单起见，这些说明假定您正在使用的映像（image）是 intersystems/iris:2021.1.0.205.0。

4. 添加许可证密钥到外部存储位置

与任何 InterSystems IRIS 实例一样，在容器中运行的实例需要一个许可证密钥（通常称为 iris.key）。

从 Docker Store 获得的 InterSystems IRIS 社区版（Community Edition）映像（在上一节中进行了描述）带有一个免费的内置临时许可证。然而，一般来说，许可证密钥没有也不能包含在 InterSystems IRIS 容器映像（container image）中，而是必须在容器启动后复制到容器中，以便激活运行在其中的 InterSystems IRIS 实例。**iris-main** 程序为此提供了一个选项，但它要求您把许可证密钥放在一个作为外部 volume 挂载的存储位置；下一节提供了使用说明。要了解有关 InterSystems IRIS 容器的许可证密钥的更多信息，请参见 *Running InterSystems Products in Containers*（《*在容器中运行 InterSystems 产品*》）中的 License Keys for InterSystems IRIS Containers（InterSystems IRIS 容器的许可证密钥）。

将您的 InterSystems IRIS 许可证密钥文件 iris.key 复制到外部存储位置。

5. 从 InterSystems IRIS 映像（Image）运行容器

一旦您在本地机器上使 InterSystems IRIS 映像（image
）可用，确定了外部存储位置并在其上放置了许可证密钥，您就可以使用 **docker
run** 命令来创建和启动容器了。**Docker run** 命令实际上结合了三个独立的命令，如下所示：

- **docker pull** — 如果本地不存在映像（image），则下载一个。
- **docker create** — 从映像（image）中创建一个容器。
- **docker start** — 启动容器。

这些命令中的每一个都是单独有用的，在不同的背景下有不同的用途。更多信息，请参见 Docker 文档中的Docker run reference（Docker 运行参考资料）。

下面是一个 **docker run** 命令的示例；所附文本对所有选项都作了解释。请注意，**docker run** 命令的选项出现在命令行中映像（image）规范之前，而 InterSystems **iris-main** 程序的选项（请参见 InterSystems IRIS in Containers（《容器中的 InterSystems IRIS》））出现在之后。在这种情况下，作为 **docker run** 一部分的 **pull** 命令不需要运行，因为您已经下载了您想使用的 **iris** 映像（image）。

```
docker run --name iris

--detach

--publish 52773:52773

--volume /nethome/pmartinez/irisexternal:/external intersystems/iris:2021.1.0.205.0

--key /external/iris.key
```

- **--name** *容器名称（container name）*

指定容器的名称，您可以在其他 Docker 命令中用它来引用该容器，例如，当您想停止该容器时，使用 **docker stop** *container name*。

- --detach

在后台运行容器（并显示容器的唯一 ID）。

- **--Publish** *主机端口：容器端口（hostport:containerport ）*

将容器内的端口发布到主机上的端口，这样容器外（在主机上或其他机器上）的实体就可以与容器内的程序联系。例如，InterSystems IRIS 实例的管理门户（Management Portal）是通过实例的 Web 服务器端口（默认为 52773）访问的。如果容器内的这个端口发布到主机上的端口，那么可以使用主机的端口将实例的管理门户（Management Portal）加载到浏览器中。

--volume *外部存储路径:内部 volume（externalstoragepath :internalvolume ）*

挂载容器可访问的外部存储位置作为容器内部的存储

volume。有关哪些存储位置可以以这种方式挂载以及可能需要的 Docker 配置的信息，请参见 Docker 文档中的 Use Volumes（使用 Volumes ）。

提示： InterSystems 不支持在 InterSystems IRIS 容器中把 NFS 位置作为外部 volume 挂载。

- *存储库|映像:标签（repository/image:tag）*

指定要提取并用于创建容器的映像（请参见 Download the InterSystems IRIS Image（下载 InterSystems IRIS 映像））。使用 **docker images** 命令来列出可用的映像（image），并确保您指定的是正确的映像（image）。

- **--Key** *许可证密钥路径（licensekeypath ）*

iris-main 选项，它标识了要安装在容器实例中的 InterSystems IRIS 许可证密钥；这个位置必须是在一个挂载的 volume 上。

使用前面的示例和说明来构建您自己的 **docker run** 命令，并在命令行上执行。当命令完成后，使用 **docker ps** 命令可以在列表中看到您的容器，状态为 **Up**。

试一试! 创建您自己的基于 InterSystems IRIS 的容器

```
$ docker run --name iris --detach --publish 52773:52773
--volume /nethome/pmartinez/irisexternal:/external intersystems/iris:2021.1.0.205.0
    --key /external/iris.key 426d4a511d6746d89ec2a24cf93b29aa546ea696b479a52210d37da4c6d04883
$ docker ps
CONTAINER ID  IMAGE                 COMMAND          CREATED      STATUS
426d4a511d67 intersystems/iris:2021.1.0.205.0 "/iris-main --key ..." 5 seconds ago Up 3 seconds PORTS   NAMES
0.0.0.0:52773->52773/tcp iris
```

注意： InterSystems IRIS 社区版（Community Edition）映像（请参见 Download the InterSystems IRIS Image（下载 InterSystems IRIS 映像））不需要**--key** 选项，该映像附带一个免费的内置许可证。

如果映像尚未在本地存在，但在您组织的存储库中——请参见 Download the InterSystems IRIS Image（下载 InterSystems IRIS 映像），Docker 会在创建和启动容器之前拉取（下载）该映像（image）。

如示例所示，创建容器后，Docker 输出 *UUID 长标识符（long identifier）*；前 12 个字符组成 *UUID 短标识符（short identifier）*，用于标识其他输出中的容器，例如来自 **docker ps** 命令的容器。

6.更改实例并将容器作为新映像（Image）提交

当您更改容器内运行的程序时，您可以使用 **docker commit** 命令从容器中创建一个新映像（image）。这个新映像（image）与您创建容器的原始映像（image）相同，但包括您对容器所作的更改。要了解其工作原理，请遵循以下步骤：

1. 打开容器中 InterSystems IRIS 实例的管理门户（Management Portal）。实例的管理门户（Management Portal）的 URL 包含了实例的主机标识符和 web 服务器端口号。
   - 主机标识符是运行容器的系统的主机名或 IP 地址；如果您的浏览器与容器运行在同一个系统上，则可以使用 localhost。
   - Web 服务器端口号是您在使用 **docker run** 启动容器时发布的实例的 web 服务器端口号，即 52773 的主机端口号。假设您包含了**--publish 52773:52773**，正如上一节末尾的示例命令中提供的那样，web 服务器端口号是 52773。

例如，在容器主机上，Web 服务器端口为 52773，管理门户（Management Portal）的 URL 将是 http://localhost:52773/csp/sys/%25CSP.Portal.Home.zen。

使用预定义用户帐户之一登录，例如 **SYSTEM**，其默认密码是 **SYS**（请参见 Predefined User

Accounts（预定义用户账户））。出于安全考虑，在第一次登录（使用管理门户（Management Portal）或 **iris terminal** 命令）到容器化的 InterSystems IRIS 实例上的任何预定义账户时，系统会立即提示您更改密码。

注意： 有关更改预定义帐户（强烈建议在生产中这样做）的默认密码（包括在脚本和自动部署中）的更多信息，请参见 *Running InterSystems Products in Containers*（《*在容器中运行 InterSystems 产品*》）中的 Authentication and Passwords （身份验证和密码）。

2. 从主页上，选择 **System Administration（系统管理）** > **Configuration（配置）** > **System Configuration（系统配置）** > **Namespaces（命名空间）** 来显示 **Namespaces（名称空间）** 页面，然后点击 **Create New Namespace（创建新的命名空间）** 按钮来显示 **New Namespace（新的命名空间）** 页面。
3. 创建名为 **USER2** 的命名空间，方法是在 **Name of the namespace box（命名空间框的名称）** 中输入 **USER2**，从 **Copy from** 下拉菜单中选择 **USER**，清除 **Enable namespace for interoperability productions（为互操作性产品启用命名空间）** 复选框并确认，点击 **Save（保存）** 按钮，最后确认您想要复制所有属性和映射。 然后点击 **Copy Namespace Mappings（复制命名空间映射）** 页面上的 **Close（关闭）** ，返回到 **Namespaces（命名空间）** 页面，在该页面上列出了 **USER2** 命名空间。 您现在已经改变了容器中的实例。
4. 接下来，停止该容器并将其作为名为 **iris2** 的新映像（image）提交，然后列出可用的映像。

```
$ docker stop iris
```

```
$ docker commit iris acme/iris2:test sha256:7b4adb9f7abf1490a39086665ccd3d255c05163c25cb9a3de8e9421f6ca16b40
```

```
$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
acme/iris2	test	421f6ca16b40	8 seconds ago	1.40GB
intersystems/iris	2020.3.0.221.0	15627fb5cb76	1 hour ago	1.39GB
centos	7.3.1611	262f7381844c	2 weeks ago	192MB
hello-world	latest	05a3bd381fc2	7 months ag	1.84kB

5. 最后，删除从原始 iris 映像创建的容器。

```
$ docker rm iris iris
```

7.运行和调查第二个基于 InterSystems IRIS 的容器

为了总结这一体验，您将使用 **docker run** 命令从您刚刚提交的基于 InterSystems-IRIS 的映像中创建并启动一个容器，包括用于持久化特定于实例的数据的持久化 %SYS 功能（durable %SYS feature）。持久化 %SYS （durable %SYS ）是保存特定于实例的数据以及您对它所做的任何更改的一种更有用的方法。因为这些数据被保存在容器之外，它可以成为新的 InterSystems IRIS 容器的数据，允许您通过运行后面的映像（image）的容器来升级 IRIS 实例，同时保留以前的容器的数据；当内部容器更改提交到新映像时，这是不可能的。有关持久化 %SYS （durable %SYS）的详细信息，请参见 *Running InterSystems Products in Containers*（《*在容器中运行 InterSystems 产品*》）中的 Durable %SYS for Persistent Instance Data（持久化实例数据的持久化 %SYS)。当您启动新容器后，将执行以下操作：

- 确认在作为一个新映像（image）提交的容器中创建的命名空间（请参见 Change the Instance and Commit the Container as a New Image（更改实例并将容器作为新映像提交） ）存在于新容器中的 InterSystems IRIS 实例中。
- 在容器中更改 InterSystems IRIS 实例的设置，并查看它反映在容器外持久化 %SYS （durable %SYS）数据中。
- 确认您添加的演示文件在容器内存在。

要做到这一点，请遵循以下步骤：

1. 为这个容器确定一个外部存储位置。您可以使用在 Add the License Key to the External Storage

Location（添加许可证密钥到外部存储位置）
中为前一个容器选择的许可证密钥，也可以选择一个新的。许可证密钥应该仍然在以前的位置。（如果您使用一个新的位置，请确保许可证密钥已经到位)。

2. 创建一个 **docker run** 命令，类似您在 Run a Container from the InterSystems IRIS Image（从 InterSystems IRIS 映像运行容器）中执行的命令， 基于那里的指令，但有两个更改。
    - 添加选项 **--env ISCDATADIRECTORY=** *pathname*

标识持久化 %SYS（durable %SYS） 目录，即写入 InterSystems IRIS 实例的持久化数据（persistent data）的位置。持久化 %SYS目录（durable %SYS directory）必须在一个挂载的 volume 上（请参见**--volume** 选项和 Add the License Key to the External Storage Location（添加许可证密钥到外部存储位置））。

注意： InterSystems 建议指定挂载 volume 的子目录为持久化 %SYS 的位置。下面的 **docker ps** 示例显示了这一点。

使用*映像：标签（image:tag）* 指定新映像

之前，您从 intersystems/iris:2021.1.0.205.0 （即 InterSystems 提供的映像）创建了容器；这次，您使用的是 acme/iris2:test，这是您通过提交更改过的 iris 容器创建的。

调用容器 iris2。当 **docker run** 命令完成后，使用 **docker ps** 命令来列出容器并查看其状态。例如：

```
$ docker run --name iris2 --detach --publish 52773:52773 --volume /nethome/pmartinez/irisexternal:/external
--env ISCDATADIRECTORY=/external/durable acme/iris2:test
    --key /external/iris.key bdfe214ef76a34290a8308cddce92162aae14df1ba1bc244e692af3c8d911a3e
$ docker ps
CONTAINER ID  IMAGE        COMMAND          CREATED      STATUS
af3c8d911a3e acme/iris2:test "/iris-main --key ..." 5 seconds ago Up 3 seconds PORTS NAMES
0.0.0.0:52773->52773/tcp  iris2
```

注意： InterSystems IRIS 社区版（Community Edition）映像（请参见 Download the InterSystems IRIS Image（下载 InterSystems IRIS 映像））不需要**--key 选项**，该映像附带一个免费的内置许可证。

Docker Compose 是一个用于定义和运行多容器应用程序的工具，为 Docker 命令行交互提供了替代方案。要使用 Compose，您需要创建 docker-compose.yml，其中包含您想要创建、启动和管理的容器的规范，然后使用 **docker-compose** 命令。更多信息，请参见 *Running InterSystems Products in Containers（《在容器中运行 InterSystems 产品》）*中的 Running an InterSystems IRIS Container:Docker Compose Example（运行 InterSystems IRIS 容器：Docker Compose 示例），和 Docker 文档中的 Overview of Docker Compose（Docker Compose 概述）。

1. 确认从已更改的 InterSystems 容器提交的更改

在 Change the Instance and Commit the Container as a New Image（更改实例并将容器作为新映像提交）中，您在从 InterSystems 提供的映像 intersystems/iris:2021.1.0.205.0 创建的容器中为 InterSystems IRIS 实例添加了一个命名空间，然后将该容器作为一个新映像 acme/iris2:test 提交。
因此，您添加的命名空间应该存在于运行在 iris2 容器内的 InterSystems IRIS 实例中，该容器是从 acme/iris2:test 创建的。

要确认这一点，请执行以下操作：

1. 打开管理门户（Management Portal）并登录，如 Change the Instance and Commit the Container as a New Image（更改实例并将容器作为新映像提交）中所述。
2. 从主页上，选择 **System Administration（系统管理）** > **Configuration（配置）** > **System Configuration（系统配置）** > **Namespaces（命名空间）**，来显示命名空间（Namespaces）页面；在 iris 容器中创建的 **USER2** 命名空间被列出。
    1. 探索和更改持久化 %SYS 目录

要探索 InterSystems IRIS 容器的持久化 %SYS 功能，请执行以下操作：

1. 要查看 InterSystems IRIS 在容器外写入的特定于实例的数据，因为您在 **docker run** 命令中包含了持久化 %SYS 环境变量，显示您在该变量中指定的目录的内容，该目录在您使用**--volume** 选项指定的存储位置，作为要挂载的外部 volume。例如，如果该目录被指定为 /nethome/pmartinez/irisexternal/durable ，如示例 **docker run** 中所示，您会做如下操作

```
$ cd /nethome/pmartinez/irisexternal

$ ls

durable iris.key

$ ls durable

csp dist httpd iris.cpf iris.cpf20180417 LastGood.cpf mgr

$ ls durable/mgr

alerts.log    irisaudit        iris.ids      irislocaldata iristemp IRIS.WIJ journal.log startup.last  SystemMonitor.log  user  ilock           IRIS.DAT
iris.lck iris.shid iris.use          journal messages.log  stream  Temp
```

2. 返回容器中 InterSystems IRIS 实例的管理门户（Management Portal），选择 **System Administration（系统管理）** > **Configuration（配置）** > **System Configuration（系统配置）** > **Journal Settings（日志设置）** 以显示日志设置（Journal Settings）页面。 **Secondary journal directory（二级日志目录）** 设置从 /external/durable/mgr/journal/ 更改为 /external/durable/mgr/journal2/，并点击 **Save（保存）** 。

3. 返回到命令行，再次列出持久化 %SYS 目录的 mgr 子目录：

```
$ ls /nethome/pmartinez/irisexternal/durable/mgr

alerts.log       irisaudit    iris.ids  iris.lck  iris.shid       iris.use      journal
 journal.log                  messages.log          stream          Temp          ilock                   IRIS.DAT
 irislocaldata                iristemp              IRIS.WIJ        journal2      licmanager.port         startup.last
 SystemMonitor.log            user
```

由于您对容器*内的* InterSystems IRIS 实例进行了更改，所以 journal2 子目录被添加到容器*外*。

这个示例显示了持久化 %SYS （durable %SYS）如何使您通过从新映像（image）创建容器来升级容器化的 InterSystems IRIS 实例。您对实例所做的所有持久化更改都存储在容器外的持久化 %SYS 目录（durable %SYS directory）；如果您使用所需的选项从任何 InterSystems IRIS 映像（image）创建并启动一个新容器 — 即 **--volume** 选项为持久化 %SYS （durable %SYS）挂载外部存储位置，以及--**env ISCDATADIRECTORY** 选项指定该挂载 volume 上持久化 %SYS（durable %SYS）位置，该位置必须存在，并且包含一个 /mgr 子目录 — 这些更改由实例继承，因为它使用与前一个容器中的实例相同的数据。

4.了解有关容器中的 InterSystems IRIS 的更多信息

此时，您已经准备好继续探索容器和 InterSystems IRIS 必须提供的东西。使用下面的文档和参考资料来深入了解容器和 InterSystems IRIS。

- Docker Containers and InterSystems IRIS（《Docker 容器和 InterSystems IRIS》） (视频)
- Running InterSystems Products in Containers（《在容器中运行 InterSystems 产品》）
- 来自 InterSystems 开发者社区（Developer Community）的文章：
  - What is a Container?（《什么是容器？》）
  - What is a Container Image?（《什么是容器映像？》）
  - Using InterSystems IRIS Containers with Docker for Windows（《使用 InterSystems IRIS 容器与 Docker for Windows》）

- Docker Documentation（《Docker 文档》）
- InterSystems Cloud Manager Guide（《InterSystems 云管理器指南》）——使用 InterSystems 云管理器（InterSystems Cloud Manager，ICM）可以轻松直观地配置基础设施，并以各种方式在上面部署容器。ICM 为 InterSystems IRIS 带来了基础设施即代码（Infrastructure as Code，IaC ）和容器化部署的好处，而不需要在新技术、培训、配置和管理方面进行重大投资。本指南包含有关 ICM 和使用 InterSystems IRIS 与 Docker 容器的文档。
- First Look:InterSystems Cloud Manager（《技术概要：InterSystems 云管理器》）

#新手 #InterSystems IRIS

源 URL: https://cn.community.intersystems.com/post/iris-2021-%E6%8A%80%E6%9C%AF%E6%96%87%E6%A1%A3-first-look-24-%E6%8A%80%E6%9C%AF%E6%A6%82%E8%A6%81%EF%BC%9A%E5%AE%B9%E5%99%A8%E4%B8%AD%E7%9A%84-intersystems-%E4%BA%A7%E5%93%81