
文章

[Nicky Zhu](#) · 十一月 2, 2021 阅读大约需 22 分钟

IRIS 2021 **技术文档** First Look 32 -- InterSystems **云管理器** (Cloud Manager)

目录

[技术概要：InterSystems 云管理器 \(Cloud Manager\)](#) 1

- 1. [ICM 能为您做什么？](#) 1
- 2. [ICM 是如何工作的？](#) 1
- 3. [试一试！使用 ICM 在云中部署 InterSystems IRIS](#) 2
 - 1. [安装 Docker](#) 3
 - 2. [下载 ICM 映像](#) 3
 - 3. [启动 ICM](#) 3
 - 4. [获取云供应商帐户和证书](#) 3
 - 5. [生成安全密钥](#) 4
 - 6. [自定义示例配置文件](#) 4
 - 7. [配置基础设施](#) 9
 - 8. [部署 InterSystems IRIS](#) 10
 - 9. [尝试 ICM 管理命令](#) 11
 - 10. [取消配置基础设施](#) 12
- 4. [ICM 能做的远不止这些！](#) 13
- 5. [了解有关 ICM 的更多信息](#) 13

插图目录

[图 1: ICM 让一切变得简单](#) 2

[图 2: 云供应商的 ICM 默认文件示例](#) 8

[图 3: 交互式 ICM 命令](#) 11

技术概要：InterSystems 云管理器 (Cloud Manager)

本文向您介绍了 InterSystems 云管理器 (InterSystems Cloud Manager, ICM)，这是基于 InterSystems IRIS®数据平台的应用程序的端到端云配置和部署解决方案。

作为本指南的一部分，您将使用 ICM 在公共云中配置基础设施，并在该基础设施上部署 InterSystems IRIS。

要浏览所有的技术概要 (First Looks)，包括那些可以在免费的[云实例](#)或[web 实例](#)上执行的技术概要 (First Looks)，请参见 InterSystems First Looks (《InterSystems 技术概要》)。

ICM 能为您做什么？

欢迎来到云时代 (Cloud Age) ! 您是否对它的机会虎视眈眈, 但又对它的挑战保持警惕? 具体来说,

- 您是否渴望利用云, 但又犹豫是否要投入资源进行复杂的迁移?
- 您是否已经在云中, 但还在努力寻找一种方法, 以便在各种软件环境中管理部署和版本化您的应用程序?
- 您是否希望将持续集成和交付引入您的软件工厂, 并将 DevOps 方法引入您的部署过程? 也就是说, 您想把自己从传统实践、库依赖、系统漂移、手动升级和其他开销的限制和风险中解放出来吗?

ICM 可以提供帮助! ICM

为您提供了一种简单、直观

的方法来配置云基础设施并在其上部署服务, 帮助您现在

就进入云, 而无需进行重大开发或重组。基础设施即代码 (infrastructure as code, IaC)

和容器化部署的优势使得在 Google、Amazon 和 Azure 等公共云平台或私有 VMware vSphere

云上部署基于 InterSystems IRIS 的应用程序变得容易。定义您想要的内容, 发布一些命令, 剩下的工作由 ICM 来完成。

即使您已经在使用云基础设施、容器, 或者同时使用两者, ICM

也可以通过自动化大量手动步骤, 极大地减少配置和部署应用程序所需的时间和精力。

ICM 是如何工作的?

在您输入的纯文本配置文件的指导下, ICM 使用 Hashicorp 流行的 Terraform IaC

工具配置您的基础设施, 并根据需要配置的主机节点。在下一阶段, ICM 将 InterSystems IRIS

和您的应用程序部署在 Docker 容器中, 如果需要, 还可以部署其他服务。您想要的部署所需的所有

InterSystems IRIS 配置都是自动执行的。ICM 还可以在现有的虚拟和物理集群上部署容器化服务。

ICM 本身带有一个容器映像, 其中包含您所需要的一切。从 InterSystems 的 ICM

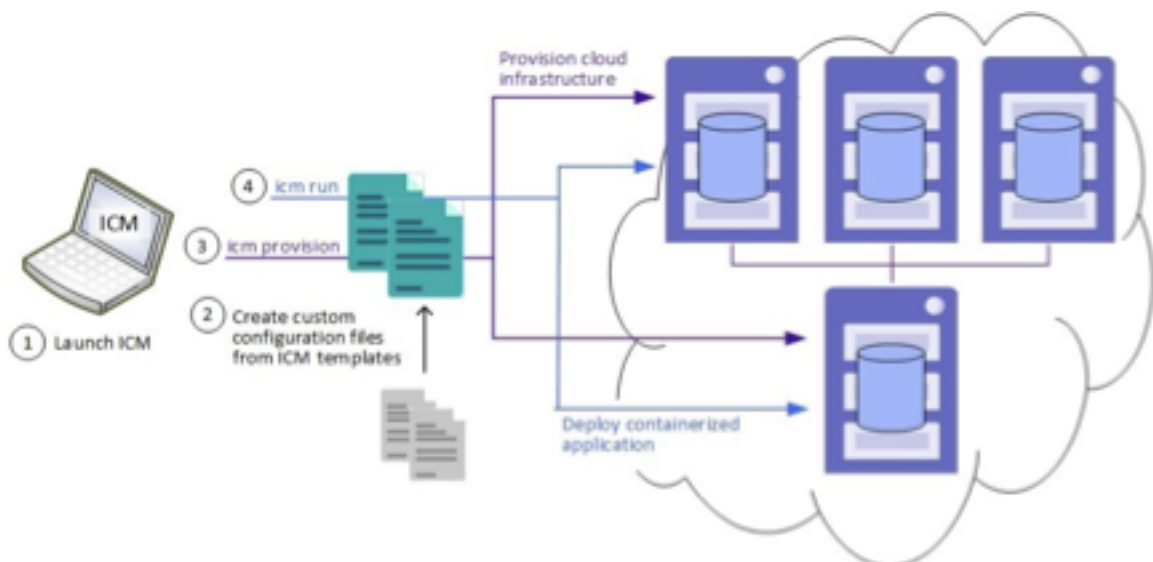
映像下载并运行容器, 打开命令行, 您就可以开始了。ICM 通过组合这些元素, 使一切变得简单:

- 可以用作模板以快速定义所需部署的示例配置文件。
- 可以向其中添加应用程序的 InterSystems IRIS 映像。

试一试! 使用 ICM 在云中部署 InterSystems IRIS

- 每个任务的用户友好命令。
- 管理已配置节点和部署在其上的服务并与之交互的多种方法。

图 1: ICM 让一切变得简单



试一试！使用 ICM 在云中部署 InterSystems IRIS

ICM 为您执行许多任务，并为您提供许多选项，以帮助您准确地部署您所需要的内容，所以在生产中使用它需要一定的计划和准备（尽管比手动方法要少得多！）。但配置和部署过程很简单，ICM 可以为您做出许多决定。这一探索旨在让您亲眼看到 ICM 是如何工作的，以及使用 ICM 在亚马逊网络服务（Amazon Web Services, AWS）上部署 InterSystems IRIS 配置是多么容易。虽然这不是一时的工作，但这种探索不应该占用您太多的时间，您可以在机会出现时分阶段进行。

为了让您体验 ICM 而又不致于在细节上陷入困境，我们保持了简单的探索；例如，我们让您尽可能多地使用默认设置。但是，当您把 ICM 引入您的生产系统时，您需要做很多不同的事情，特别是（但不限于）安全方面。所以请确保不要把这种对 ICM 的探索与真实的情况相混淆！本文档末尾提供的参考资料将使您对在生产中使用 ICM 的情况有一个很好的了解。ICM Guide（《ICM 指南》）提供了使用 ICM 的完整信息和程序，并在适当的地方提供了链接。

这些说明假定您具备以下条件：

- 特定于容器的 InterSystems IRIS 分片（sharding）许可证并可以访问 InterSystems 软件下载。
- 亚马逊网络服务（Amazon Web Services, AWS）、谷歌云平台（Google Cloud Platform, GCP）、微软云（Azure）或腾讯云（Tencent）上的帐户。

在配置文件中指定的许多属性在这些云平台上是通用的，但其他属性是特定于平台的。有关这些差异的详细信息可在 ICM Guide（《ICM 指南》）的"ICM Reference（《ICM 参考资料》）"一章中的"Provider-Specific Parameters（特定于供应商的参数）"部分找到。

试一试！使用 ICM 在云中部署 InterSystems IRIS

安装 Docker

ICM 是作为容器映像提供的，它包含您需要的所有内容。因此，对启动 ICM 的 Linux、macOS 或 Microsoft Windows 系统的唯一要求是安装 Docker，并运行 Docker 守护进程，以及系统连接到 Internet。有关在您的平台上安装 Docker 的信息，请参见 Docker 文档中的 [Install Docker（《安装 Docker》）](#)。

重要提示： Docker 企业版（Enterprise Edition）和社区版（Community Edition）18.09 及更高版本支持 ICM；企业版（Enterprise Edition）仅支持生产环境。

下载 ICM 映像（Image）

要使用 ICM，您需要将 ICM 映像下载到您正在工作的系统中；这要求您识别将从其中下载它的注册表和访问所需的凭证。同样，对于 ICM 部署 InterSystems IRIS 和其他 InterSystems 组件，它需要相关映像的此信息。ICM 下载映像的注册表必须能被您使用的云供应商访问（也就是说，不能在防火墙后面），而且为了安全起见，必须要求 ICM 使用您提供给它的凭证进行身份验证。有关识别所涉及的注册表和下载 ICM 映像的详细信息，请参见 InterSystems Cloud Manager Guide（《InterSystems 云管理器指南》）中的 Downloading the ICM Image（下载 ICM 映像）。

注意： 启动 ICM 的映像的主要版本和部署的 InterSystems 映像必须匹配。例如，您不能使用 2019.3 版本的 ICM 部署 2019.4 版本的 InterSystems IRIS。

有关在容器中使用 InterSystems IRIS 的简要介绍，包括实际操作经验，请参见 First Look: InterSystems Products in Containers（《技术概要：容器中的 InterSystems 产品》）；有关使用 ICM 以外的方法在容器中部署 InterSystems IRIS 和基于 InterSystems IRIS 的应用程序的详细信息，请参见

Running InterSystems IRIS in Containers (《在容器中运行 InterSystems IRIS》)。

启动 ICM

要在命令行上启动 ICM，使用下面的 `docker run` 命令，从注册表中拉出（下载）ICM 映像，从中创建一个容器，并启动该容器，例如：

```
docker run --name icm -it --cap-add SYS_TIME  
containers.intersystems.com/intersystems/icm:2021.1.0.205.0
```

ICM 容器中的 `/Samples`

目录，每个供应商都有一个（`/AWS`，`/GCP`，等等），旨在使您能够轻松地使用开箱即用的 ICM 进行配置和部署；您可以使用这些目录之一的示例配置文件

，也可以从该目录进行配置和部署。然而，由于这些目录在 ICM

容器内，当容器被删除时，其中的数据就会丢失，包括您的配置文件和 ICM

创建并用于管理配置的基础设施的状态目录。在生产中，最好的做法是在启动 ICM

时通过挂载外部卷使用容器外的位置来存储这些数据（请参见 Docker 文档中的 [Manage data in Docker \(在 Docker 中管理数据\)](#))。

获取云供应商帐户和证书

ICM 可以在四个公共云平台上进行配置和部署：亚马逊网络服务（Amazon Web Services，AWS）、谷歌云平台（Google Cloud Platform，GCP）、微软云（Azure）和腾讯云（Tencent）。

登录您的 AWS、GCP、Azure 或 Tencent

帐户。如果您和您的雇主都还没有帐户，您可以进入[AWS](#)、GCP、Azure或Tencent 门户页面，快速创建一个免费帐户。

有关获取 ICM 需要向每个云供应商进行身份验证的帐户凭证的信息，请参见 ICM Guide (《ICM 指南》) 的 "ICM Reference (《ICM 参考资料》)" 一章中的 Security- Related Parameters (安全相关的参数)。

试一试! 使用 ICM 在云中部署 InterSystems IRIS

生成安全密钥

ICM 与配置基础设施的云供应商、配置的节点上的 Docker 以及容器部署后的若干 InterSystems IRIS 服务进行安全的通信。您已经下载或识别了您的云证书；您还需要其他文件来启用安全的 SSH 和 TLS 通信。

您可以使用 ICM 提供的两个脚本来创建您需要的文件，`keygenSSH.sh` 脚本创建所需的 SSH 文件并将它们放在 ICM 容器的 `/Samples/ssh` 目录中。`keygenTLS.sh` 脚本创建所需的 TLS 文件，并将它们放在 `/Samples/tls` 中。

要在 ICM 命令行上运行这些脚本，请输入以下内容：

```
# keygenSSH.sh
```

```
Generating keys for SSH authentication.
```

```
...
```

```
# keygenTLS.sh
```

```
Generating keys for TLS authentication.
```

...

有关这些脚本生成的文件的更多信息，请参见 ICM Guide (《ICM指南》) 中的 Obtain Security-Related Files (获取安全相关的文件)、ICM Security (ICM 安全) 和 Security-Related Parameters (安全相关的参数)。

重要提示： 这些生成的密钥是为了方便您在本文体验和其他测试中使用；在生产中，您应该根据您的安全策略来生成或获取所需的密钥。这些脚本生成的密钥以及您的云供应商凭证必须完全安全，因为它们提供了对使用它们的任何 ICM 部署的完全访问权限。

自定义示例配置文件

要定义您希望 ICM 部署的配置，您在两个 JSON 格式的配置文件中包含所需的设置：默认文件和定义文件。前者 (defaults.json) 包含应用于整个部署的设置 --- 例如，您对云供应商的选择 -----而后者 (definitions.json) 定义了您想要的节点类型以及每种节点的数量，从而决定了您是部署一个分片集群、一个独立的 IRIS 实例，还是其他配置。

ICM 容器中的 /Samples 目录为所有四个云供应商提供示例配置文件。例如，AWS 的示例配置文件位于 /Samples/AWS 目录中。要自定义这些文件，请选择您要使用的供应商，然后按照下面的表格描述，为该供应商修改示例默认值和定义文件。要做到这一点，您可以使用 vi 等编辑器直接在容器内编辑它们，或者在本地命令行上使用 docker cp 命令将它们从容器复制到本地文件系统，然后在您编辑它们后再次返回，例如：

```
docker cp icm:/Samples/AWS/defaults.json . docker cp icm:/Samples/AWS/definitions.json .  
  
...  
  
docker cp defaults.json icm:/Samples/AWS docker cp definitions.json icm:/Samples/AWS
```

重要提示： 字段名和值都是区分大小写的；例如，要选择 AWS 作为云供应商，您必须在默认文件中包含 "Provider": "AWS"，而不是 "provider": "AWS"、"Provider": "aws"等等。

自定义 defaults.json

下表详细说明了 /Samples 文件中提供的默认文件所需的最小自定义，以及对可选更改的建议。这里没有涉及到的任何设置都可以在示例文件中保持原样。

试一试! 使用 ICM 在云中部署 InterSystems IRIS

每个设置都连接到 ICM Guide (《ICM 指南》) 的 "ICM Reference (《ICM 参考资料》)" 一章中的 ICM Configuration Parameters (ICM 配置参数) 部分中的相关表格，在那里您可以找到更详细的说明；在那里，搜索您想要的参数。最右边的列显示了在示例默认文件中出现的每个参数或参数集。

要查看所有供应商通用的所有设置，请参见该部分的 General Parameters (常规参数)；要查看特定于特定供应商的所有设置，请参见 Provider-Specific Parameters (特定于供应商的参数)。

下图显示了 AWS、GCP、Azure 和 Tencent 的示例 defaults.json 文件的内容：

设置	说明	示例 defaults.json 文件中的条目
Provider	识别云基础设施供应商；保留示例 defaults.json 中的值。	" Provider " :[" AWS " "GCP" "Azure" "Tencent"] ,
Label	配置节点的命名方案中的字段，Label-Role-Tag-NNNN (请参见下面的 Role)；更新以识别部署的所有者和目的，例如，使用您的公司名称和 " TES T " 来创建节点名称，如Acme-DATA-	"Label": "Sample",

设置	说明	示例 defaults.json 文件中的条目
Tag	TEST-0001。 (请参见上面的 Label)	"Tag": "TEST",
DataVolumeSize	为每个节点配置的持久化数据卷的大小 ，这可以在 definitions.json 文件中的 各个节点定义中被覆盖；接受示例 defaults.json 中的值，除非您在 Tencent 上配置，在这种情况下，将其改为 60。	"DataVolumeSize": "10",
SSHUser	在配置节点上具有 sudo 访问权限的非根帐户，用于 ICM 访问；您可以保留示例 defaults.json 中的默认值，但如果您在 AWS 或 Tencent 上更改了机器映像(如下)，则 可能需要更新此条目。	"SSHUser": "ubuntu", (AWS & Tencent) "SSHUser": "sample", (GCP & Azure)
SSHPublicKey	SSH 公钥的位置。如果您使用了 Generate Security Keys (生成安全密钥) 中讨论的密钥生成脚本，则密钥位于 示例文件指定的目录中，因此无需更改 ；如果您提供了自己的密钥，请使用 docker cp 将它们从本地文件系统复制 到这些位置。	"SSHPublicKey": "/Samples/ssh/insecure-ssh2.pub",
SSHPrivateKey	SSH 私钥的位置；请参见上面的 SSHPublicKey。	"SSHPrivateKey": "/Samples/ssh/insecure",
TLSKeyDir	TILS 文件的位置；请参见上面的 SSHPublicKey。	"TLSKeyDir": "/Samples/tls/",
设置	说明	示例 defaults.json 文件中的条目
DockerVersion	要安装在已配置节点上的 Docker 版本；保持默认值。	"DockerVersion": "5:19.03.83-0ubuntu-bionic",
DockerImage	要部署在已配置节点上的映像；更新以 反映您在 Identify Docker Repository and Credentials (识别 Docker 存储库和凭证) 中识别的存储库和映像信息。	"DockerImage": "containers.intersyste ms.com/intersystems/iris:2021.1.0.205 .0",
DockerUser- name Docker- Password	如果在私有存储库中，则下载 DockerImage 指定的映像所需的凭证 ；请更新以反映您在 Identify Docker Repository and Credentials (识别 Docker 存储库和凭证) 中识别的存储库信息和凭证。	"DockerUsername": "xxxxxxxxxxxxx", "DockerPassword": "xxxxxxxxxxxxx",
LicenseDir	InterSystems IRIS 许可证的暂存目录 ；将您的特定于容器的 InterSystems IRIS 分片 (sharding) 许可证放在这 个目录中。	"LicenseDir": " /Samples/Licenses ",
Region, Location (Azure)	供应商用于配置基础设施的计算资源的 地理区域；接受示例 defaults.json 中的默认值，或者从供应商那里选择 region 和 zone 的另一个组合 (如下)。	"Region": "us-west-1", (AWS) "Region": "us-east1", (GCP) "Location": "Central US", (Azure) "Region": "na-siliconvalley", (Tencent)
Zone	所选区域内的可用区 (如上)；接受示 例 defaults.json 中的默认值，或者从供应商那里选择 region 和 zone 的另一个组合。	"Zone": "us-west-1c", (AWS) "Zone": "us-east1-b", (GCP) "Zone": "1", (Azure) "Zone": "na- siliconvalley-1", (Tencent)
Machine image (特定于供应商)	配置节点的平台和 OS 模板；接受示例 defaults.json 中的默认值，或者从供应商那里选择 machine image 和 instance type	"AMI": "ami-c509eda6", (AWS) "Image": "ubuntu-os-cloud/ubuntu-180 4-bionic-v20180617", (GCP) "PublisherName": "Canonical", (Azure)

设置	说明	示例 defaults.json 文件中的条目
	的不同组合 (如下)。	"Offer": "UbuntuServer", (Azure) "Sku": "18.04-LTS", (Azure) "Version": "18.04.201804262", (Azure) "ImageID": "img-pi0ii46r", (Tencent)
设置 Instance type (特定于供应商)	说明 配置节点的计算资源模板；接受示例 defaults.json 中的值，或者从供应商那里选择 machine image 和 instance type 的不同组合 (如上)。	示例 defaults.json 文件中的条目 "InstanceType": "m4.large", (AWS) "MachineType": "n1-standard-1", (GCP) "Size": "Standard_DS2v2", (Azure) "InstanceType": "S2.MEDIUM4", (Tencent)
Credentials and account settings (特定于供应商)	ICM 向供应商进行身份验证所需的文件或 ID，因供应商而异；更新以指定您的帐户所需的文件位置或 ID (有关说明，请点击供应商链接)。	"Credentials": "/Samples/AWS/credentials", (AWS) "Credentials": "/Samples/GCP/sample.credentials", (GCP) "Project": "dp-icmdevelopment", (GCP) "SubscriptionId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx", (Azure) "ClientId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx", (Azure) "ClientSecret": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx", (Azure) "TenantId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx", (Azure) "SecretID": "xxxxxxxxxxxx", (Tencent) "SecretKey": "xxxxxxxxxxxx", (Tencent)
ISCPassword	已部署的 InterSystems IRIS 映像中预定义帐户的密码；要在部署阶段以交互方式提供带有屏蔽输入的密码 (为安全起见建议这样做)，请删除此字段，否则更改为您的首选密码。	"ISCPassword": "",
Mirror	确定部署在 DATA、DM 和 DS 节点上的 InterSystems IRIS 实例是否被配置为镜像；保留默认值。	"Mirror": "false"
设置 UserCPF	说明 指定用于覆盖已部署实例的初始 CPF 设置的配置合并文件。(如果您不熟悉配置合并功能或 CPF，请删除此条目；有关配置合并的信息，请参见 ICM Guide (《ICM**指南》) 的 "ICM Reference (《ICM 参考资料》)" 一章中的 Deploying with Customized InterSystems IRIS Configurations (使用自定义的 InterSystems IRIS 配置进行部署)。	示例 defaults.json 文件中的条目 "UserCPF": "/Samples/cpf/iris.cpf"

下图显示了 AWS、GCP、Azure 和 Tencent 的示例 defaults.json 文件的内容：

图 2: 云供应商的 ICM 默认文件示例

<pre>{ "Provider": "AWS", "Label": "Sample", "Tag": "TEST", "DataVolumeSize": "10", "SSHUser": "ubuntu", "SSHPublicKey": "/Samples/ssh/insecure-ssh2.pub", "SSHPrivateKey": "/Samples/ssh/insecure", "DockerImage": "intersystems/iris:2020.3.0.N.0", "DockerUsername": "xxxxxxxxxxxx", "DockerPassword": "xxxxxxxxxxxx", "TLSKeyDir": "/Samples/tls/", "LicenseDir": "/Samples/license/", "Region": "us-west-1", "Zone": "us-west-1c", "DockerVersion": "5:19.03.8~3-0~ubuntu-bionic", "AMI": "ami-08fd8ae3806f09a08", "InstanceType": "m4.large", "Credentials": "/Samples/AWS/sample.credentials", "ISCPasswd": "", "Mirror": "false", "UserCPF": "/Samples/cpf/iris.cpf" }</pre>	<pre>{ "Provider": "GCP", "Label": "Sample", "Tag": "TEST", "DataVolumeSize": "10", "SSHUser": "sample", "SSHPublicKey": "/Samples/ssh/insecure.pub", "SSHPrivateKey": "/Samples/ssh/insecure", "DockerImage": "intersystems/iris:2020.3.0.N.0", "DockerUsername": "xxxxxxxxxxxx", "DockerPassword": "xxxxxxxxxxxx", "TLSKeyDir": "/Samples/tls/", "LicenseDir": "/Samples/license/", "Credentials": "/Samples/GCP/sample.credentials", "Project": "dp-icmdevelopment", "MachineType": "n1-standard-1", "Region": "us-east1", "Zone": "us-east1-b", "Image": "ubuntu-os-cloud/ubuntu-1804-bionic-v20190911", "DockerVersion": "5:19.03.8~3-0~ubuntu-bionic", "ISCPasswd": "", "Mirror": "false", "UserCPF": "/Samples/cpf/iris.cpf" }</pre>
<pre>{ "Provider": "Azure", "Label": "Sample", "Tag": "TEST", "DataVolumeSize": "10", "SSHUser": "sample", "SSHPublicKey": "/Samples/ssh/insecure.pub", "SSHPrivateKey": "/Samples/ssh/insecure", "DockerImage": "intersystems/iris:2020.3.0.N.0", "DockerUsername": "xxxxxxxxxxxx", "DockerPassword": "xxxxxxxxxxxx", "TLSKeyDir": "/Samples/tls/", "LicenseDir": "/Samples/license/", "Location": "Central US", "Zone": "1", "PublisherName": "Canonical", "Offer": "UbuntuServer", "Sku": "18.04-LTS", "Version": "18.04.201909030", "DockerVersion": "5:19.03.8~3-0~ubuntu-bionic", "Size": "Standard_DS2_v2", "AccountTier": "Standard", "AccountReplicationType": "LRS", "SubscriptionId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx", "ClientId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx", "ClientSecret": "xx", "TenantId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx", "ISCPasswd": "", "Mirror": "false", "UserCPF": "/Samples/cpf/iris.cpf" }</pre>	<pre>{ "Provider": "Tencent", "Label": "Sample", "Tag": "TEST", "DataVolumeSize": "60", "SSHUser": "ubuntu", "SSHPublicKey": "/Samples/ssh/insecure.pub", "SSHPrivateKey": "/Samples/ssh/insecure", "DockerImage": "intersystems/iris:2020.3.0.N.0", "DockerUsername": "xxxxxxxxxxxx", "DockerPassword": "xxxxxxxxxxxx", "TLSKeyDir": "/Samples/tls/", "LicenseDir": "/Samples/license/", "SecretID": "xxxxxxxxxxxx", "SecretKey": "xxxxxxxxxxxx", "InstanceType": "S2.MEDIUM4", "Region": "na-siliconvalley", "Zone": "na-siliconvalley-1", "ImageId": "img-pi0i146n", "DockerVersion": "5:19.03.8~3-0~ubuntu-bionic", "ISCPasswd": "", "Mirror": "false", "UserCPF": "/Samples/cpf/iris.cpf" }</pre>

自定义 definitions.json

在 /Samples 目录中的示例 definitions.json 文件（对所有供应商来说都是一样的），定义了一个有两个数据节点和两个计算节点的分片集群，如下所示：

试一试! 使用 ICM 在云中部署 InterSystems IRIS

```
[
{
  "Role": "DATA",

  "Count": "2",

  "LicenseKey": "ubuntu-sharding-iris.key"
},

{
  "Role": "COMPUTE",
```

```
"Count": "2",  
  
"StartCount": "3",  
  
"LicenseKey": "ubuntu-sharding-iris.key"  
  
}  
  
]
```

Role 字段标识正在配置的节点类型，在这种情况下是 DATA 和 COMPUTE。Count 字段表示要配置多少个该类型的节点；对于 COMPUTE 节点，StartCount 从 0003 开始编号。LicenseKey 字段表示 InterSystems IRIS 许可证文件的名称，该文件位于默认文件中由 LicenseDir 字段指定的目录中。

在这个练习中，删除 COMPUTE 定义，只保留 DATA 定义，如下所示：

```
[  
  
{  
  
  "Role": "DATA",  
  
  "Count": "2",  
  
  "LicenseKey": "ubuntu-sharding-iris.key"  
  
}  
  
]
```

部署分片集群时，所有节点必须有分片（sharding）许可证。使用 docker cp 将分片（sharding）许可证复制到 LicenseDir 指定的容器内的位置，如 /Samples/Licenses，并更新 DATA 节点定义中的 LicenseKey 设置，以指定要使用的许可证密钥。

这些是配置基本分片集群所需的唯一 definitions.json 更改。要配置一个独立的 InterSystems IRIS 实例，请使用此定义：

```
[  
  
{  
  
  "Role": "DM",  
  
  "Count": "1",  
  
  "LicenseKey": "standard-iris.key"  
  
},  
  
]
```

配置基础设施

默认情况下，ICM

从当前目录中的配置文件中获取输入，因此您需要做的就是将配置基础设施更改为您选择的供应商的 /Samples 目录，例如 /Samples/AWS，并发出这个命令：

icm provision

icm provision 命令在您选择的平台上分配和配置主机节点。在配置操作期间，ICM 在状态子目录中创建或更新状态和日志文件，完成后创建 instances.json 文件，作为后续部署和管理命令的输入。

因为 ICM 同时运行多个任务，所以这些步骤（Terraform 计划和应用、复制文件、挂载卷等等）可能不会以相同的顺序在节点上启动和完成。完成后，ICM 提供了一个已经配置好的主机节点的摘要，并输出一个命令行，可以在以后用来删除基础设施，如下所示：

Machine IP Address DNS Name Region Zone

```
Acme-DATA-TEST-0001 00.53.183.209 ec2-00-53-183-209.us-west-1.compute.amazonaws.com us-west-1  
c Acme-DATA-TEST-0002 00.53.183.185 ec2-00-53-183-185.us-west-1.compute.amazonaws.com us-  
west-1 c To destroy: icm unprovision [-cleanUp] [-force]
```

试一试! 使用 ICM 在云中部署 InterSystems IRIS

重要提示： 复制输出中提供的 icm unprovision 命令行，并保存此信息，这样您就可以在取消配置时轻松地复制它。这个输出也出现在 icm.log 文件中。

管理和取消配置您的基础设施所需的文件和目录都在 ICM 容器中，因此当您删除它时就会丢失。因此，在完成基础设施并成功地取消配置之前，您不能删除 ICM 容器。（在生产中，最好的做法是在启动 ICM 时通过挂载外部卷使用容器外的位置来存储这些数据（请参见 Docker 文档中的 Manage data in Docker（在 Docker 中管理数据））。

在配置和部署阶段发生的错误的详细信息写入状态目录子目录中的 terraform.err 文件；当发生错误时，ICM 会将您指向适当的文件，该文件可以帮助您识别问题的原因。

如果 icm provision 由于超时和供应商方面的内部错误，或者配置文件中的错误而没有成功完成，您可以根据需要再次发出该命令，直到 ICM 为所有指定的节点完成所有需要的任务而没有错误。更多信息，请参见 ICM Guide（《ICM 指南》）的 "Using ICM（《使用 ICM》）" 一章中的 "Reprovisioning the Infrastructure（重新配置基础设施）"。

有关 icm provision 的更多信息，请参见 ICM Guide（《ICM 指南》）中的 The icm provision Command（icm 配置命令）。

部署 InterSystems IRIS

除了在提供的主机节点上下载映像（Image）并将它们作为容器运行外，ICM 还执行特定于 InterSystems IRIS 的配置和其他任务。要在您已配置的节点上部署 InterSystems IRIS，请留在 /Samples 目录，您在其中自定义了配置文件并配置了基础设施，然后发出以下命令：

icm run

默认情况下，icm run 下载并运行配置文件中 DockerImage 字段指定的映像，在这种情况下，是来自 InterSystems 存储库的 InterSystems IRIS 映像；每个容器都被命名为 iris。在实践中，在定义文件中的节点定义中使用不同的 DockerImage 字段（而不是对默认文件中的所有节点使用一次，如此处），可以让您在不同的节点类型上运行不同的映像，除此之外，您还可以使用某些选项多次执行 icm run 命令，以便在每个配置的节点上或只在指定节点上部署具有唯一名称的多个容器。

InterSystems IRIS 在每个节点上启动后，ICM 会执行实例所需的任何配置 --- 例如，重置密码、配置分片（sharding）等。因为 ICM 同时运行多个任务，因此部署过程中的步骤可能不会以相同的顺序在节点上启动和完成。

完成后，ICM 输出一个链接到相应的 InterSystems IRIS 实例的管理门户（Management Portal）：

```
$ icm run -definitions definitionscluster.json
```

Executing command 'docker login' on ACME-DATA-TEST-0001...

...output in /Samples/AWS/state/ACME-DATA-TEST/ACME-DATA-TEST-0001/docker.out

...

Pulling image intersystems/iris:2021.1.0.205.0 on ACME-DATA-TEST-0001...

...pulled ACME-DATA-TEST-0001 image intersystems/iris:2021.1.0.205.0

...

Creating container iris on ACME-DATA-TEST-0002...

...

Management Portal available at:

<http://ec2-00-53-183-209.us-west-1.compute.amazonaws.com:52773/csp/sys/UtilHome.csp>

在这种情况下，所提供的链接用于数据节点

1，或者用于独立的实例（如果这是您选择的配置）。在浏览器中打开链接，并使用管理门户（Management Portal）探索 InterSystems IRIS。

与 icm provision 一样，如果 icm run 由于超时和供应商方面的内部错误而没有成功完成，您可以再次发出命令；在大多数情况下，反复尝试，部署就会成功。但是，如果错误仍然存在，并且需要手动干预 ---

例如，如果它是由某个配置文件中的错误引起的 ---

在修复问题后，您可能需要删除受影响的一个或多个节点上的持久化 %SYS 数据，

试一试! 使用 ICM 在云中部署 InterSystems IRIS

在重新发布 icm run 命令之前，如 ICM Guide（《ICM 指南》）的 "Using ICM（《使用 ICM》）" 一章中的 Redeploying Services（重新部署服务）所述。在出现错误的情况下，请参见 ICM 指向您访问的日志文件，以获得有助于您确定问题原因的信息。

有关 icm run 命令及其选项的更多信息，请参见 ICM Guide（《ICM 指南》）中的 The icm run Command（icm run 命令）。

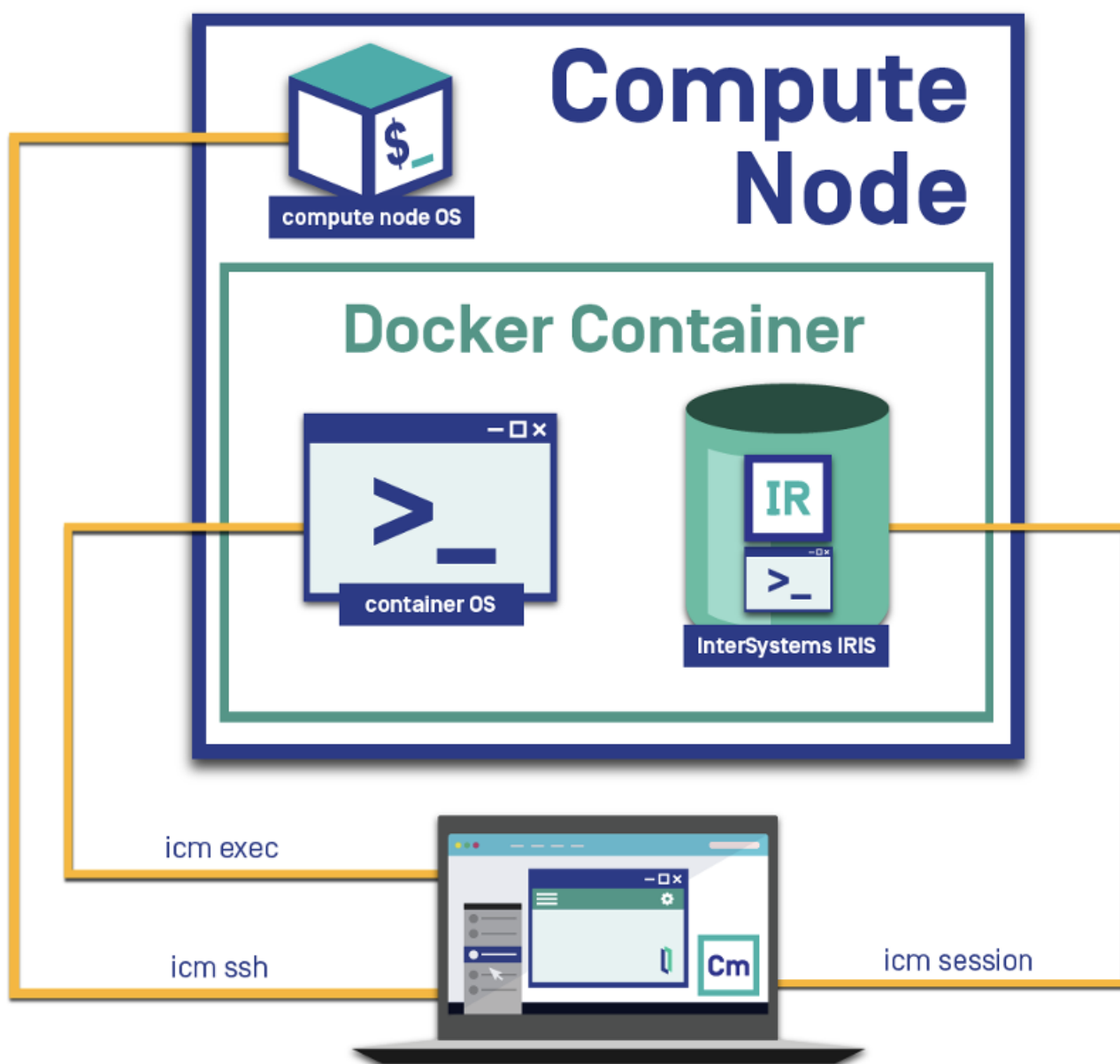
尝试 ICM 管理命令

ICM 提供了一系列的命令用于

- 管理配置的基础设施。
- 管理部署的容器。
- 与运行在部署的容器中的服务交互，包括 InterSystems IRIS。

这些命令在 ICM Guide（《ICM 指南》）的 Infrastructure Management Commands（基础设施管理命令）、Container Management Commands（容器管理命令）和 Service Management Commands（服务管理命令）小节中有详细的讨论，并在 ICM Commands and Options（ICM 命令和选项）中全面列出。这里提供了一些示例，供您在部署时试用。特别要注意的是，有三个命令（icm ssh、icm exec 和 icm session）可以让您在多个层次上与您部署的节点交互 --- 与节点本身、与部署在它上面的容器以及与容器内运行的 InterSystems IRIS 实例，如下所示：

图3: 交互式 ICM 命令



在您新部署的 InterSystems IRIS 配置上试试这些命令吧!

1. 列出配置的主机节点：

试一试! 使用 ICM 在云中部署 InterSystems IRIS

```
icm inventory
```

2. 在每个主机节点上运行 shell 命令：

```
icm ssh -command "df -k"
```

当一个命令在多个节点上执行时，输出被写入文件，并提供一个输出文件的列表。例如，在这种情况下，如果您部署了分片集群，三个节点各有一个输出文件。

3. 在特定的主机节点上打开交互式 shell：

```
icm ssh -role DATA -interactive
```

要做到交互，命令必须使用 `-interactive` 选项并指定单个节点。例如，虽然 `icm ssh` 可以在多个节点上执行命令，如前面的示例所示，但它只能使用 `-interactive` 选项在一个节点上打开交互式 shell，如本例所示。代替 `-role DM` 选项将命令限制在该类型的节点上（在您的部署中只有一个节点），您也可以使用 `-machine` 选项来指定一个特定节点的名称，例如：

```
icm ssh -machine Acme-DATA-TEST-0001 -interactive
```

4. 显示部署的 InterSystems IRIS 容器的状态：

```
icm ps
```

当在节点上部署了多个容器时，这个命令会将它们全部列出。

5. 将本地文件复制到 InterSystems IRIS 容器或其中一个容器：

```
icm cp -localPath /Samples/ssh/sshnotes -remotePath /home/sshuser
```

```
icm cp -localPath /Samples/ssh/sshnotes -remotePath /home/sshuser -role DM
```

6. 在每个 InterSystems IRIS 容器中运行 shell 命令：

```
icm exec -command "ls /irissys/"
```

7. 在特定的 InterSystems IRIS 容器中打开一个交互式 shell（或运行任何 shell 命令）：

```
icm exec -command "bash" -role DATA -interactive
```

8. 为 InterSystems IRIS 实例打开一个终端（Terminal）窗口（对于单个实例，总是交互式的）：

```
icm session -machine Acme-DATA-TEST-0001
```

9. 针对 InterSystems IRIS 的每个实例，或针对特定的实例，运行 SQL 命令：

```
icm sql -command "SELECT Name FROM Security.Users"
```

```
icm sql -command "SELECT Name FROM Security.Users" -machine Acme-DATA-TEST-0002
```

取消配置基础设施

因为 AWS 和其他公共云平台实例会不断产生费用，所以在完成这一体验后，立即取消配置基础设施是很重要的。

要做到这一点，把您从 `icm provision` 的输出中保存的 `icm unprovision` 命令复制到 ICM 命令行，例如：

```
$ icm unprovision -cleanUp
```

ICM 能做的远不止这些!

如果您没有从配置输出中保存命令，您可以在工作目录的 icm.log 文件中找到它（例如，/Samples/AWS/icm.log）。-cleanUp 选项在取消配置后会删除状态目录；如果没有这个选项，状态目录会被保留下来。

ICM 能做的远不止这些!

尽管您刚刚完成的 ICM 探索是刻意简化的，但它仍包含了现实世界的基础设施配置和服务部署。添加到您刚刚完成的只是在配置文件中扩展您的部署定义并利用大量命令行选项的问题。例如，您可以

- 通过定义 AM 节点和 DM 节点，部署应用服务器和数据服务器的分布式缓存集群，或通过仅定义 DM 节点部署独立的 InterSystems IRIS 实例。
- 部署镜像（mirror）DATA 节点或镜像 DM 节点，只需在默认文件中包含 "Mirror": "True"，并在定义文件中定义偶数个 DATA 节点或两个 DM 节点和一个 AR（仲裁器）节点。
- 部署一个包含 COMPUTE 节点的分片集群，以分离查询和数据摄取工作量，同时保持并行处理和分布式缓存的优势，提高两者的性能。
- 在定义文件中添加 web 服务器（WS）和负载均衡器（LB 或自动）。
- 通过在定义文件中为不同的节点指定不同的 DockerImage 值，并多次执行 icm run 命令，在不同的节点上部署不同的服务。
- 通过多次执行 icm run，同时在命令中使用 -container 和 -image 选项指定不同的容器名称和映像（包括自定义和第三方映像），在一些或所有节点上部署多个服务。
- 通过使用第三方工具和内部脚本扩展 ICM 的功能，进一步提高自动化程度，减少工作量。
- 在已有的虚拟和物理集群上部署服务。
- 在不部署服务的情况下配置基础设施，只需使用 ICM provision 命令停止。

所有这些可能性以及更多的可能性都在 ICM Guide（《ICM 指南》）中有所涉及。

了解有关 ICM 的更多信息

要了解有关 ICM 和在容器中使用 InterSystems IRIS 的更多信息，请参见

- [InterSystems Cloud Manager Introduction \(《InterSystems 云管理器简介》\)](#) (视频)
- [The Benefits of InterSystems Cloud Manager \(《InterSystems 云管理器的优势》\)](#) (视频)
- [Experience InterSystems IRIS in the Cloud \(在云中体验 InterSystems IRIS\)](#) (在线体验)
- InterSystems Cloud Manager Guide (《InterSystems 云管理器指南》)
- First Look: InterSystems Products in Containers (《技术概要：容器中的 InterSystems 产品》)
- Running InterSystems Products in Containers (《在容器中运行 InterSystems 产品》)

[#AWS #平台 #InterSystems IRIS for Health](#)

源

URL:

<https://cn.community.intersystems.com/post/iris-2021-%E6%8A%80%E6%9C%AF%E6%96%87%E6%A1%A3-first-look-32-intersystems-%E4%BA%91%E7%AE%A1%E7%90%86%E5%99%A8%E5%BC%88cloud-manager%E5%BC%89>