

文件版本：v1.2

# 超算互联网服务商手册

## 应用软件适配

2024年01月17日

# 目 录

一、 说明.....	1
二、 应用软件定义.....	1
三、 开通平台账号和测试权限.....	1
3.1 开通平台账号.....	1
3.2 开通测试权限.....	1
四、 封装测试区及应用软件仓库.....	2
4.1 封装测试区.....	2
4.2 应用仓库.....	3
五、 应用软件封装测试流程.....	4
5.1 上传应用软件包.....	4
5.2 测试应用软件包.....	4
5.3 选择封装类型.....	5
5.4 封装应用软件包.....	5
5.5 测试封装后的包.....	5
5.6 安全检测.....	5
5.7 上传应用仓库.....	5
六、 操作规范.....	6
6.1 名词解释.....	6
6.2 命名规范.....	7
6.2.1 软件包文件命名规范.....	7
6.2.2 软件包目录命名规范.....	8
6.2.3 容器镜像命名规范.....	8
6.3 Tar 应用封装规范 .....	9
6.4 conda 应用封装规范 .....	10
6.5 虚拟机镜像应用封装规范.....	11
6.6 容器镜像应用封装规范.....	12
七、 应用软件发布运营 .....	12

## 一、说明

本文档为应用软件服务商（以下简称“服务商”）提供入驻超算互联网平台的流程指引。

## 二、应用软件定义

应用软件为满足用户不同领域、不同问题的应用需求而提供的直接用于解决实际问题软件。支持在云端进行部署及交付，能减少用户在本地采购硬件的费用和周期，利用云端算力资源快速求解、分析及优化等。

应用软件主要包括工业类、科学类及其他应用领域的软件，支持 Windows、Linux 系统部署。

超算互联网平台引入的应用软件的收费模式及定价由服务商自行确定。计费模式包括但不限于：

- 1) 免费软件：最终用户在平台免费调用软件进行仿真计算；
- 2) 收费软件：按月/按年收费、按核时收费等，用户在超算互联网平台采购并使用应用软件，计费均由平台进行，平台和服务商约定结算时间和模式。

## 三、开通平台账号和测试权限

### 3.1 开通平台账号

服务商入驻意向确认完毕后，可在超算互联网平台点击注册按钮，注册平台账号，用于商家入驻，开店及超算互联网平台应用软件测试及验证工作。账号为超算互联网平台账号，服务商需登录超算互联网平台验证账号的有效性。

### 3.2 开通测试权限

服务商应向超算互联网运营人员提供如下信息，并由运营人员开通应用软件的测试权限。

表 1 应用软件信息

软件	提供内容
应用软件 1	软件安装包（包括软件安装文件，license manager 安装文件（如有））
	软件典型测试算例
	软件使用手册

	软件测试 license（如需 MAC 地址申请 license，请联系运营人员）
应用软件 2	同上
应用软件 3	同上
...	...

应用软件由平台/服务商技术工程师进行封装测试后，将上架为商品并发布到 scnet.cn。服务商商品发布审核提交后，管理员进行审核，通过后的软件可在应用商城进行查看，驳回的商品申请需进行修改后再重新提交审核。

商品发布成功后，服务商即可登录 scnet.cn 采购对应的软件，并通过控制台中，点击账号名->我的商品->应用软件，查看采购的商品列表，并点击对应的命令行/模板提交方式，测试软件的可用性。

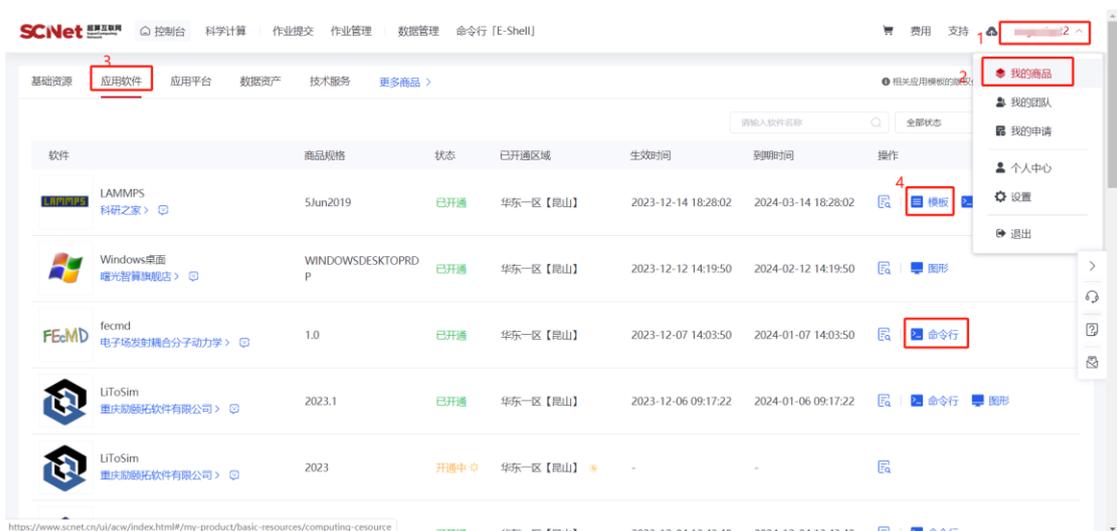


图 1 已购应用软件列表界面

## 四、封装测试区及应用软件仓库

鉴于不同的软件适配的硬件资源有一定的差异，平台为服务商提供了不同资源配置的封装测试区及应用软件仓库。封装测试区及应用软件仓库的要求如下：

### 4.1 封装测试区

- 1) 为应用软件的封装流程提供基础封装环境；
- 2) 针对不同的封装类型建立不同的权限受控的封装测试区；
- 3) 应用软件包可以上传到对应的封装测试区内，根据应用需要对应用软件进行编译，安装，配置；

- 4) 通过平台或者后台命令将不同封装类型的应用软件进行封装。
- 5) 为应用软件基础案例的运行提供运行环境，确保仓库内软件包的可用性，将可能出现的问题控制在最小范围；
- 6) 对应用软件进行可用性测试，如果发现问题及时回到封装测试区调整应用软件重新封装。

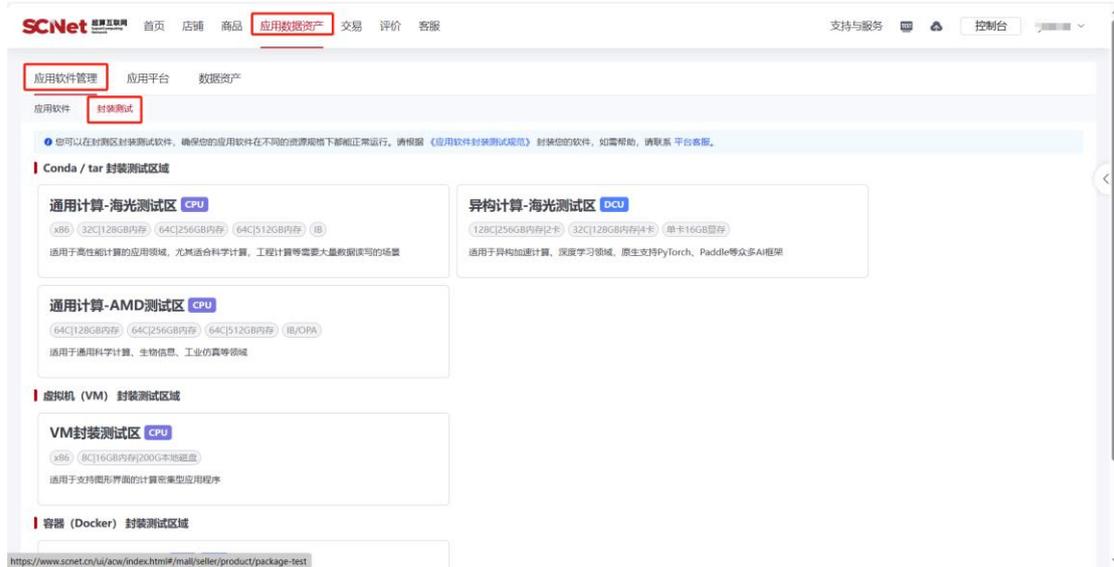


图 2 封装测试区

## 4.2 应用仓库

应用仓库为统一存储及管理超算互联网平台应用、数据、模型等文件。应用仓库提供以下能力：

- 1) 支持主流超算应用的管理与使用方式；
- 2) 支持对用户的权限管理，并与超算互联网平台用户管理系统对接；
- 3) 支持一定的安全管理策略，防范文件被非法窃取或破坏；
- 4) 满足 100 用户并发访问，秒级相应；
- 5) 满足 10G 大小镜像和软件的分钟级上传及下载；
- 6) 上下级仓库之间传输带宽不低于 100MB/S。

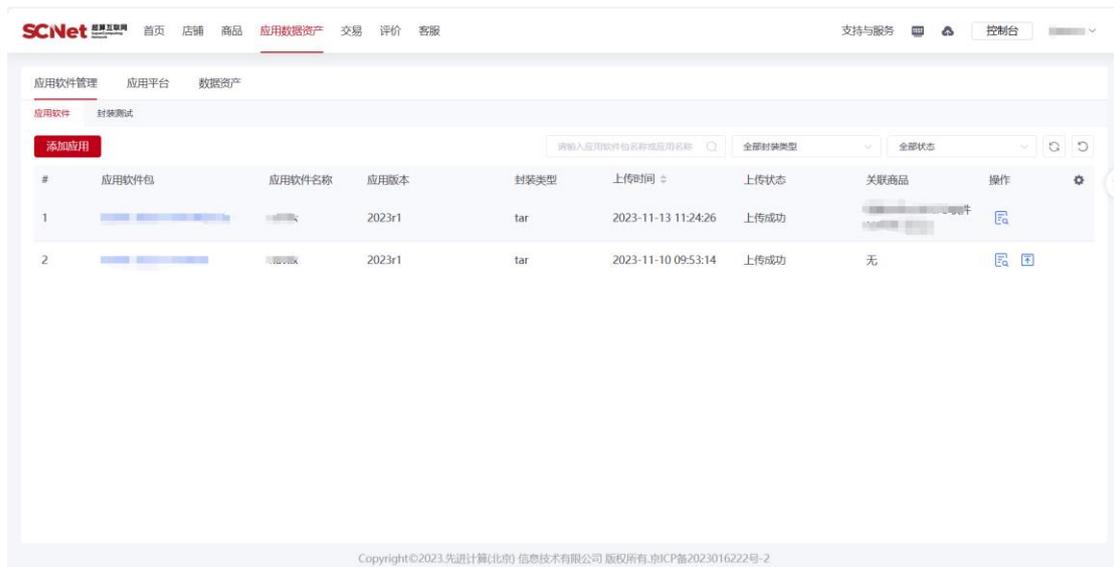


图 3 应用软件仓库

## 五、应用软件封装测试流程

服务商上架的商品为应用软件，平台支持 Linux 和 Windows 系统软件封装测试并上架。

- Linux: Centos7.6
- Windows: Windows server 2016、Windows 10

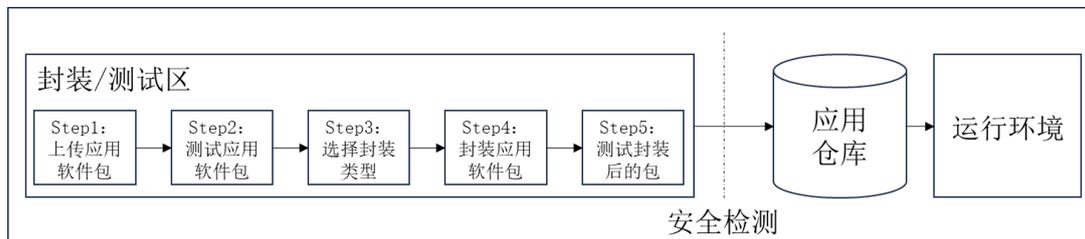


图 4 应用封装测试流程

### 5.1 上传应用软件包

平台/服务商技术工程师将应用软件包上传至封装测试区。应用软件包的命名和目录规范请参考 6.2 命名规范。

### 5.2 测试应用软件包

根据应用软件包中相关文档的提示，部署应用软件，并结合服务商提供的典型测试算例进行测试，如应用软件为免费模式，则直接运行；如应用软件需 license 启动，需服务商同时提供 license 文件（如 license 文件需通过 license manager 进行管理，需同时提供 license manager 的安装包；如 license 文件需绑定 mac 地址，请联系运营人员提供）。

## 5.3 选择封装类型

封装常用 HPC 应用软件环境和文件类型，包括：

- **Tar 包形式（推荐）**  
将应用及相关文件以 Tar 包的形式封装
- 符合 Conda 环境的封装形式  
将应用按照 Conda 环境要求文件进行封装
- 虚拟机镜像形式  
将应用及环境以虚拟机镜像的形式封装
- 容器镜像形式  
将应用及环境以容器镜像的形式封装

平台/服务商工程师根据应用软件的类型，选择适合的封装类型。

## 5.4 封装应用软件包

平台/服务商技术工程师对软件包执行封装操作。不同类型的应用软件封装规范可参考六、操作规范。

## 5.5 测试封装后的包

平台工程师对封装后的软件包需放到测试环境进行可用性测试（和 5.2 采用同一算例进行测试，计算结果一致，则表示可用性测试通过）。通过后才需放到超算互联网平台应用仓库。用于后续软件商品在店铺进行上架、销售及服务。

## 5.6 安全检测

- 1) 应对所有待发布或部署到应用仓库的软件包进行安全检测，发现其中漏洞、后门以及病毒等危险因素；
- 2) 应针对检测结果高危的软件进行及时处理；
- 3) 应针对检测发现漏洞及时进行修复；
- 4) 根据实际环境，确定安全检测阶段。

## 5.7 上传应用仓库

经过安全性检测与漏洞扫描的应用软件包，确保应用软件经过合法授权，没有版权纠纷

后上传至应用仓库，最后部署到用户运行环境。

## 六、操作规范

超算互联网平台应用仓库提供多种不同的应用软件封装类型供服务商选择(conda、tar、vm、docker)。平台/服务商技术工程师按照本规范封装应用软件，上传到应用仓库，以实现超算互联网平台自动化安装应用软件。具体详细操作规范可参考封装测试区界面的《应用软件封装测试规范》。



图 5 应用软件封装测试规范线上文档位置

### 6.1 名词解释

#### ➤ tar

tar 是 Linux 操作系统下的一种文件格式，平台支持技术工程师封装 tar 类型的应用软件。

#### ➤ conda

conda 是一种包、环境的管理工具，并且提供了包安装管理的功能。平台支持技术工程师封装 conda 类型的应用软件。

#### ➤ recipe

recipe 是指构建 conda 包时的依赖文件。主要包括：meta.yaml、build.sh 等文件。

#### ➤ 虚拟机镜像

虚拟机镜像为副本文件，安装有操作系统的一个磁盘分区。该副本文件包含了一块或多块云盘中的所有数据。安装虚拟机时，会运行镜像里的安装程序，这程序的主要任务是把镜像里的东西解压到磁盘文件里，不会修改镜像里的内容。镜像文件格式为 qcow2。平台/服务商支持技术工程师封装虚拟机镜像类型的应用软件。

#### ➤ 容器镜像

容器镜像支持封装应用软件以及依赖包到一个可移植的镜像中，然后发布到 Linux 或

Windows 操作系统的机器上，也可以实现虚拟化。常见的如 docker、singularity 镜像等。平台/服务商支持技术工程师封装容器镜像类型的应用软件。

## 6.2 命名规范

### 6.2.1 软件包文件命名规范

平台针对应用软件定义命名制定规范，以便更好的区分不同类型及作用的应用。软件包文件命名应符合下列要求：

a) 应以 X-Y-Z 形式命名，X-Y-Z 之间以中划线“-”分割。

其中：

1) X：应用名称

2) Y：应用版本号

3) Z：应用描述信息。应用描述信息中，如包含编译器、MPI、加速卡驱动等描述信息，以下划线“\_”分割。

示例：

软件名称-版本-[编译器]\_[MPI]\_[驱动]\_[其他描述信息]

b) 不能使用生僻字符和特殊符号命名，如：\$、%、\*等。

软件名称-版本-[编译器]\_[MPI]\_[驱动]\_[其他描述信息] 完整的规则如下：

表 2 软件包命名规范

	说明	规则描述
X	应用名	必须为小写英文，可以带下划线“_”，不允许中划线“-”
Y	版本	不允许下划线“-”
Z	描述信息	可分为 3 部分，以下划线“-”分隔 ① 编译器（选填） ② MPI（选填） ③ 驱动（选填） ④ 其他描述信息（选填）
tar	后缀	封装的 Tar 应用应为 tar 为后缀
tar.bz2/conda	后缀	conda 默认为 tar.bz2，conda 格式需要二次转换

qcow2	后缀	封装的虚拟机镜像应用应为 qcow2 或主流镜像后缀
-------	----	----------------------------

注意：

- 1) conda 构建过程中的 recipe 文件定义了上述各部分内容，按照规范填写各部分内容即可，请勿手动修改构建完成的应用文件名称；
- 2) 因为使用 tar 封装本身不限制规则，所以用户在封装时，应该严格遵守命名规范；

## 6.2.2 软件包目录命名规范

软件包目录包括软件内所有内容文件，命名符合下列要求：

- 1) 应用软件全部文件，放置到单独的文件夹，应命名为“app”；
- 2) 标准算例和运行脚本，放置到单独文件夹，应命名为“case”；
- 3) 应用软件的运行说明，写到单独的文件，应命名为“README”；
- 4) 应用软件的帮助文档/手册，放置到单独文件夹，宜命名为“man”；
- 5) 应用软件的源码，放置到单独文件夹，宜命名为“source”。

示例：

以 sapp 应用软件为例，封装应用之前，准备的工作目录结构示例如下：

```
sapp/
├── app
│   ├── bin
│   │   ├── sapp_tmpl
│   │   ├── sapp_cfg
│   │   └── sap_std
│   ├── include
│   └── lib
├── case
│   ├── input.txt
│   └── test.sh
├── man
├── readme.md
├── scripts
│   └── env.sh
└── source
```

## 6.2.3 容器镜像命名规范

应符合下列要求：

- a) docker 应用镜像命名

应以 X:Y-Z 形式命名， X:Y 之间以冒号 “:” 分割， Y-Z 之间以中划线 “-” 分割。

其中：

1) X: 应用名称

2) Y: 应用版本号

3) Z: 应用描述信息。应用描述信息中，如包含编译器、MPI、加速卡驱动等描述信息，以下划线 “\_” 分割。

示例：

软件名称:应用版本-[编译器]\_[MPI]\_[加速卡驱动]\_[Python 版本]\_[基础 OS]\_[其他]

tensorflow:1.15.1-dtk22.10.1\_py36\_centos7.6

pytorch:1.10.0-cuda10.0\_py36\_ubuntu20.04

b) Apptainer 应用镜像命名

应以 X-Y-Z 形式命名， X-Y-Z 之间以中划线 “-” 分割。

其中：

1) X: 应用名称

2) Y: 应用版本号

3) Z: 应用描述信息。应用描述信息中，如包含编译器、MPI、加速卡驱动等描述信息，以下划线 “\_” 分割。

示例：

软件名称-应用版本-[编译器]\_[MPI]\_[加速卡驱动]\_[Python 版本]\_[基础 OS]\_[其他]

hpcx-2.6.0-intel2017.5.239\_dtk22.10.1.sif

tensorflow-1.15.1-dtk22.10.1\_py36\_centos7.6.sif

### 6.3 Tar 应用封装规范

tar 用于封装商业类应用软件。商业类应用软件的依赖由自身解决，整体封装在一个单独完整的应用软件包内。打包应符合下列要求：

a) 完整性：封装的应用软件应具有完整性，包含应用软件及其全部依赖组件，整体封装在一个单独的包内；

b) 可移植性：封装的应用软件应当具有可移植性，在适配的操作系统环境上，解压即具备直接运行的条件；

c) 最小封装原则：封装的应用软件应当遵循最小封装原则，只封装应用软件相关的依

赖。

以 vasp 为例，准备构建应用之前，准备的工作目录结构示例如下：

```
vasp_root/  
├── app  
│   ├── bin  
│   │   ├── vasp_gam  
│   │   ├── vasp_ncl  
│   │   └── vasp_std  
│   └── include  
├── lib  
├── case  
│   └── INPUT  
├── test.sh  
├── man  
├── README.md  
├── scripts  
├── env.sh  
└── source
```

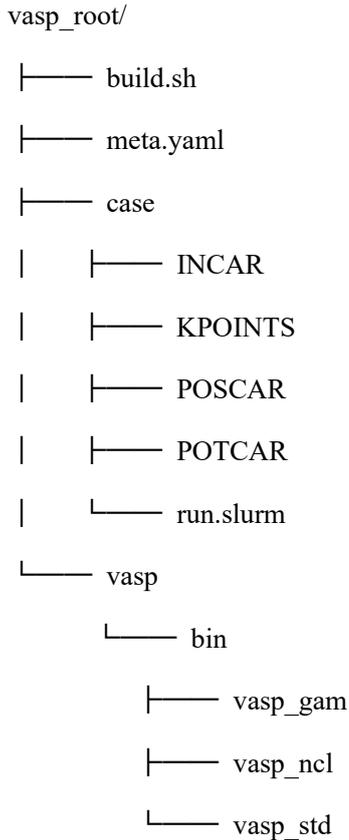
## 6.4 conda 应用封装规范

Conda 是超算中心常用的软件包管理工具，用于封装开源类应用软件。利用 conda 工具自身能够解决软件包依赖的能力，来解决高性能计算集群上适配开源应用软件的封装。基于 Conda 封装应符合下列要求：

- a) 编写 build.sh 和 meta.yaml 两个配置文件；
- b) 应在 build.sh 中编写对应的 modulefile 文件内容，保证安装完成之后 modulefile 可用；
- c) 应在 meta.yaml 中定义应用程序所需要的依赖包，确保应用软件安装时，可以检查运行环境，如果缺少依赖回将依赖同时进行安装。

Conda 封装参考案例如下：

以 vasp 为例，准备构建应用之前，准备的工作目录结构示例如下：



## 6.5 虚拟机镜像应用封装规范

虚拟机镜像可用于封装图形桌面类应用软件。图形桌面类应用软件对资源的需求具有较强的伸缩性和对图形渲染的需求特点，因此把图形桌面类应用软件封装为一个独立的封闭环境的虚拟机。封装应符合下列要求：

- a) 虚拟机镜像的制作可基于已有基础镜像环境，进行克隆；
- b) 应可封装 linux 与 windows 常见操作系统版本的镜像；
  - **linux:** 构建过程中，进入 VM 时使用 root 权限，验证以及 VM 应用运行时，linux 应为普通用户角色权限，linux 镜像启动时会进行初始化操作，检查配置镜像的可用性；
  - **Windows:** windows 镜像一般会封装一些图形界面相关的应用，在 windows 中进行前处理相关的操作，后续的计算过程放到计算节点上进行运算。
- c) 虚拟机镜像启动后能够挂载集群用户家目录；
- d) 虚拟机镜像启动后能够与集群用户打通。

## 6.6 容器镜像应用封装规范

容器技术是一种通过隔离和封装应用软件及其依赖项，以实现轻量级、可移植和可扩展的应用软件部署和管理的方法。

Docker 和 Apptainer 是容器技术的两种不同实现。Docker 可以实现封装、发布和运行分布式应用软件。Apptainer 为高性能计算领域应用设计，是一种轻量级的容器技术，使用 Apptainer 部署容器应用，容器内部可以访问外部相同的资源。封装应符合下列要求：

- a) 容器内应用环境尽量简单，减少对系统库及软件包的依赖关系；
  - b) 单个容器镜像，只启用和该封装应用相关的进程；
  - c) 宜选择官方提供镜像作为原始版本，根据实际环境，加入必要软件作为基础镜像。
- docker 应用镜像命名示例

软件名称:应用版本-[编译器]\_[MPI]\_[加速卡驱动]\_[Python 版本]\_[基础 OS]\_[其他]

Tensorflow	tensorflow:1.15.1-dtk22.10.1_py36_centos7.6
Pytorch	pytorch:1.10.0-cuda10.0_py36_ubuntu20.04

- Apptainer 应用镜像命名示例

软件名称-应用版本-[编译器]\_[MPI]\_[加速卡驱动]\_[Python 版本]\_[基础 OS]\_[其他]

hpcx	hpcx-2.6.0-intel2017.5.239_dtk22.10.1.sif
tensorflow	tensorflow-1.15.1-dtk22.10.1_py36_centos7.6.sif

## 七、应用软件发布运营

当服务商完成技术对接确认应用软件可在平台上正常部署、运行，并确认商务合作内容之后，服务商即可在超算互联网平台的应用商城注册成为应用软件服务商。之后，即可在店铺中发布应用软件商品。即超算互联网所有用户可以浏览、购买应用软件，超算互联网和服务商启动应用软件商品的运营工作。

应用软件商品发布流程请参考：<https://www.scnet.cn/help/docs/mainsite/seller-guide/publish-software/index.html>。