1. 下载镜像

docker pull trinityctat/infercnv:latest

* docker pull: 这是 Docker 命令的一部分，用于从 Docker 镜像仓库中拉取（即下载）一个镜像到本地计算机。
* trinityctat/infercnv: 这是镜像的名称。Docker 使用这个名称来唯一标识镜像。在这个例子中，trinityctat/infercnv是一个命名空间为trinityctat，镜像名称为infercnv的镜像。
* latest: 这是标签（tag）。标签用于标识镜像的特定版本或者变体。latest是一个常用的标签名称，表示拉取最新版本的镜像。如果不指定标签，默认会拉取latest标签的镜像。

1. 运行容器

docker run --rm -it -v `pwd`:`pwd` trinityctat/infercnv:latest bash

* docker run: 这是 Docker 命令的一部分，用于在新的容器中运行一个特定的镜像。
* --rm: 这个选项告诉 Docker 在容器退出时自动删除容器。这意味着，一旦你退出 Bash shell 并关闭容器，相关的容器实例会被自动清理和删除。
* -it: 这是两个选项的结合，分别是 -i 和 -t。-i 表示交互式操作，允许你在容器内部进行输入。-t 表示为容器分配一个伪终端（pseudo-TTY），使得你可以与 Bash shell 进行交互。
* -v `pwd`:`pwd`: 这是一个卷挂载（Volume Mount）的指令。它将当前工作目录挂载到容器内部的相同位置。反引号 `` `` 是用来执行 shell 命令的符号，pwd 会被替换为当前工作目录的路径。
* trinityctat/infercnv:latest: 这是要运行的 Docker 镜像的名称和标签。它告诉 Docker 从 trinityctat/infercnv 镜像的 latest 标签中创建一个新的容器。
* bash: 这是在容器中要启动的命令。在这种情况下，它是在容器内部启动一个交互式的 Bash shell，允许你与容器内的操作系统进行交互。

1. 后台运行容器

docker run --rm -d -v `pwd`:`pwd` trinityctat/infercnv:latest "cd inferCNV/example ; Rscript ./run.R"

* docker run: 这是 Docker 命令的一部分，用于在容器中运行一个镜像。
* --rm: 这是一个选项，它告诉 Docker 在容器退出时自动删除容器。即，一旦容器停止运行，相关的容器实例会被自动清理和删除。
* -d: 这是一个选项，表示在后台（detached mode）运行容器，使其在后台运行而不占据当前终端的控制权。
* -v ``pwd:pwd``: 这是一个卷挂载（Volume Mount）的指令，用反引号 `` `` 包裹的 pwd 命令，表示将当前工作目录挂载到容器内部的相同位置，以便容器内部可以访问宿主机的文件系统。
* trinityctat/infercnv:latest: 这是要运行的 Docker 镜像的名称和标签。它指定了 trinityctat/infercnv 镜像的 latest 标签。
* "cd inferCNV/example ; Rscript ./run.R": 这部分是要在容器中执行的命令。在这个例子中，使用了双引号将整个命令串起来。
  + cd inferCNV/example：这个命令将容器的当前工作目录更改为 inferCNV/example 目录。
  + Rscript ./run.R：这个命令使用 R 语言的解释器运行了 run.R 文件。假设在inferCNV/example 目录中存在名为 run.R 的文件，此命令将在容器中执行该脚本文件。

1. 导出镜像

docker save -o infercnvimg.tar trinityctat/infercnv:latest

* docker save: 这是 Docker 命令的一部分，用于将一个或多个 Docker 镜像保存到一个文件中。
* -o infercnvimg.tar: 这是一个选项，-o 指定了输出的目标文件名。在这个例子中，infercnvimg.tar 是要保存镜像的 tar 文件的名称。该文件会包含一个或多个镜像的数据。
* trinityctat/infercnv:latest: 这是要保存的 Docker 镜像的名称和标签。在这种情况下，它指定了 trinityctat/infercnv 镜像的 latest 标签。

1. 导入镜像

docker load -i infercnvimg.tar

* docker load: 这是 Docker 命令的一部分，用于加载一个或多个 Docker 镜像。
* -i infercnvimg.tar: 这是一个选项，-i 指定了输入的源文件。在这个例子中，infercnvimg.tar 是要加载镜像数据的 tar 归档文件。

1. 查看镜像

docker images

这个命令 docker images 是 Docker 命令之一，用于列出本地计算机上存储的 Docker 镜像列表。当你运行这个命令时，它会显示类似如下格式的输出：



每一列的含义如下：

* REPOSITORY：镜像的仓库名称，通常包括镜像的作者或来源。
* TAG：镜像的标签，标识着镜像的版本或者变体。有些镜像可能没有标签，那么默认会显示为 <none>。
* IMAGE ID：镜像的唯一标识符，用于标识镜像的内容。
* CREATED：镜像的创建时间或者最后更新时间。
* SIZE：镜像所占用的磁盘空间大小。

1. 查看容器

docker ps

docker ps是 Docker 命令之一，用于列出正在运行的容器。当你在终端中运行 docker ps 命令时，它会显示当前正在运行的容器列表。

默认情况下，docker ps 只会显示运行中的容器。如果没有正在运行的容器，它将不会输出任何内容。

如果你想显示所有的容器，包括运行中的和已经停止的容器，可以使用 -a 标志，即 docker ps -a。

这个命令会显示一系列容器相关的信息，例如：

* CONTAINER ID：容器的唯一标识符。
* IMAGE：容器所基于的镜像。
* COMMAND：容器启动时执行的命令。
* CREATED：容器的创建时间。
* STATUS：容器的状态，如运行中（running）、已停止（exited）等。
* PORTS：容器暴露的端口。
* NAMES：容器的名称。

通过运行 docker ps 命令，你可以查看当前正在运行的 Docker 容器的概要信息，以便监视它们的状态、端口绑定等。