



Packaging Holoscan Applications

Table of contents

Prerequisites

Package an application

Run a packaged application

The [Holoscan App Packager](#), included as part of the [Holoscan CLI](#) as the `package` command, allows you to package your Holoscan applications into a [HAP-compliant](#) container image for distribution and deployment.

Prerequisites

Dependencies

Ensure the following are installed in the environment where you want to run the [CLI](#):

- **[PIP dependencies](#)** (automatically installed with the holoscan python wheel)
- **[NVIDIA Container Toolkit with Docker](#)**
 - Developer Kits (aarch64): already included in IGX Software and JetPack
 - x86_64: tested with NVIDIA Container Toolkit 1.13.3 w/Docker v24.0.1
- **Docker BuildX plugin**

1. Check if it is installed:

```
$ docker buildx version github.com/docker/buildx v0.10.5 86bdced
```

2. If not, run the following commands based on the [official doc](#):

```
# Install Docker dependencies sudo apt-get update sudo apt-get install ca-  
certificates curl gnupg # Add Docker Official GPG Key sudo install -m 0755 -  
d /etc/apt/keyrings curl -fsSL  
https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o  
/etc/apt/keyrings/docker.gpg sudo chmod a+r  
/etc/apt/keyrings/docker.gpg # Configure Docker APT Repository echo \  
"deb [arch="$(dpkg --print-architecture)" signed-  
by=/etc/apt/keyrings/docker.gpg]  
https://download.docker.com/linux/ubuntu \ "$( . /etc/os-release && echo  
"$VERSION_CODENAME")" stable" | \ sudo tee
```

```
/etc/apt/sources.list.d/docker.list > /dev/null # Install Docker BuildX Plugin  
sudo apt-get update sudo apt-get install docker-buildx-plugin
```

- **QEMU** (*Optional*)
 - used for packaging container images of different architectures than the host (example: x86_64 -> arm64)

CLI Installation

The Holoscan CLI is installed as part of the Holoscan SDK and can be called with the following instructions depending on your installation:

Ingested Tab Module

Package an application

Tip

The packager feature is also illustrated in the [cli_packager](#) and [video_replayer_distributed](#) examples.

1. Ensure to use the [HAP environment variables](#) wherever possible when accessing data. For example:

Let's take a look at the distributed video replayer example ([examples/video_replayer_distributed](#)).

- **Using the Application Configuration File**

Ingested Tab Module

- **Using Environment Variable `HOLOSCAN_INPUT_PATH` for Data Input**

Ingested Tab Module

2. Include a YAML configuration file as described in the [Application Runner Configuration](#) page.
3. Use the `holoscan package` command to create a HAP container image. For example:

```
holoscan package --platform x64-workstation --tag my-awesome-app --config
/path/to/my/awesome/application/config.yaml
/path/to/my/awesome/application/
```

Run a packaged application

The packaged Holoscan application container image can run with the [Holoscan App Runner](#):

```
holoscan run -i /path/to/my/input -o /path/to/application/generated/output my-
application:1.0.1
```

Since the packaged Holoscan application container images are OCI-compliant, they're also compatible with [Docker](#), [Kubernetes](#), and [containerd](#).

Each packaged Holoscan application container image includes tools inside for extracting the embedded application, manifest files, models, etc. To access the tool and to view all available options, run the following:

```
docker run -it my-container-image[:tag] help
```

The command should print following:

```
USAGE: /var/holoscan/tools [command] [arguments]... Command List extract -----
----- Extract data based on mounted volume paths.
/var/run/holoscan/export/app extract the application
/var/run/holoscan/export/config extract app.json and pkg.json manifest files and
application YAML. /var/run/holoscan/export/models extract models
/var/run/holoscan/export/docs extract documentation files
/var/run/holoscan/export extract all of the above IMPORTANT: ensure the directory
```

to be mounted for data extraction is created first on the host system. and has the correct permissions. If the directory had been created by the container previously with the user and group being root, please delete it and manually create it again.

show ----- Print manifest file(s): [app | pkg] to the terminal. app print app.json pkg print pkg.json env ----- Print all environment variables to the terminal.

i Note

The tools can also be accessed inside the Docker container via `/var/holoscan/tools`.

For example, run the following commands to extract the manifest files and the application configuration file:

```
# create a directory on the host system first mkdir -p config-files # mount the directory created to /var/run/holoscan/export/config docker run -it --rm -v $(pwd)/config-files:/var/run/holoscan/export/config my-container-image[:tag] extract # include -u 1000 if the above command reports a permission error docker run -it --rm -u 1000 -v $(pwd)/config-files:/var/run/holoscan/export/config my-container-image[:tag] extract # If the permission error continues to occur, please check if the mounted directory has the correct permission. # If it doesn't, please recreate it or change the permissions as needed. # list files extracted ls config-files/ # output: # app.json app.yaml pkg.json
```

© Copyright 2022-2024, NVIDIA.. PDF Generated on 06/06/2024