

Apptainer

Apptainer is a container platform built for high-performance computing (HPC). A container is a self-contained "digital box" that packages software, its dependencies, and configuration files into a single portable file.

Why use Apptainers?

- **Reproducibility:** Locks the software environment, identical results across time and systems.
- **Isolation:** Each container runs its own dependencies, avoiding version conflicts.
- **Portability:** One file runs anywhere Apptainer is installed, on RCAC clusters or locally.
- **Security:** Runs without root privileges, making it safe for shared HPC environments.

Useful Sub-Commands

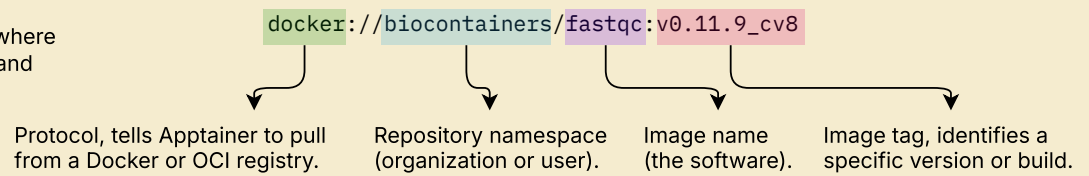
pull	Download prebuilt container images using the source URI
exec	Run a specific command inside a container
run	Execute the container's default command
shell	Open an interactive shell inside the container for debugging or exploring.
inspect	View metadata such as labels, environment variables, or help text from the container.
build	Build an image from a recipe
cache	Clear or check cached layers to free space or troubleshoot pulls.
verify	Check a container's signature or integrity
sif	Examine or extract information from sif files

Biocontainers are deployed as modules on RCAC clusters

```
module --force purge
module load biocontainers
module load fastqc/0.11.9
```

URI

A URI (Uniform Resource Identifier) tells Apptainer where to find a container image and how to retrieve it.



apptainer pull

```
apptainer pull fastqc.sif docker://biocontainers/fastqc:v0.11.9_cv8 Pulls FastQC image from BioContainers on Docker Hub.
apptainer pull hisat2.sif docker://quay.io/biocontainers/hisat2:2.2.1--h503566f_8 Pulls HISAT2 from Quay.io registry.
apptainer pull tensorflow.sif docker://tensorflow/tensorflow:latest Downloads latest version of TensorFlow image
apptainer pull --arch arm --arch-variant 6 alpine_arm.sif docker://alpine:latest Pulls ARM architecture image
apptainer pull --sandbox trinity/ docker://trinityrnaseq/trinityrnaseq:latest Pulls as a sandbox directory for inspection
apptainer pull https://example.com/rseqc.sif Downloads RSeQC image directly from an HTTPS link.
apptainer pull --disable-cache docker:// biocontainers/bwa:v0.7.17_cv1 Pulls BWA aligner image without using cache.
```

apptainer exec

```
apptainer exec multiqc.sif multiqc . Run multiqc on all output files in the current directory.
apptainer exec -B $SCRATCH:/data fastqc.sif fastqc /data/reads_1.fq Bind-mount a directory (-B) for input reads
apptainer exec --nv deepvariant.sif run_deepvariant ... Enable GPU acceleration for DeepVariant using --nv.
apptainer exec -e salmon.sif salmon quant -i index -l A -reads.fq -p 8 -o out Use -e to start with a clean environment.
apptainer exec -C bwa.sif bwa mem ref.fa reads.fq > aln.sam Use -C (--containall) to isolate (no shared /tmp or $HOME)
apptainer exec -W $SCRATCH trimgalore.sif trim_galore reads.fq Use -W to specify working directory
apptainer exec --overlay myoverlay.img maker.sif maker ... Run MAKER with a writable overlay to save intermediate data.
```

Custom Apptainer Image (from existing image)

```
apptainer pull --sandbox ubuntu_sandbox docker://ubuntu:22.04
```

Pull official Ubuntu image as a writable sandbox
(creates `ubuntu_sandbox/` instead of a `sif` file)

1

```
apptainer shell --writable ubuntu_sandbox
```

Enter the sandbox in writable mode with root-like
access limited to that sandbox environment

2

```
apt-get update && apt-get install -y wget unzip openjdk-11-jre perl
cd /opt
wget https://www.bioinformatics.babraham.ac.uk/projects/fastqc/fastqc_v0.12.1.zip
unzip fastqc_v0.12.1.zip && rm fastqc_v0.12.1.zip
chmod +x /opt/FastQC/fastqc
ln -s /opt/FastQC/fastqc /usr/local/bin/fastqc
exit
```

Install software inside the container
You can either use the system package
manager or manually install binaries. Place
executables in the `$PATH`

3

```
apptainer exec ubuntu_sandbox fastqc --version
```

Test inside the sandbox before building

4

```
apptainer build fastqc_custom.sif ubuntu_sandbox
```

Build a final `.sif` image from the sandbox.

This file can be shared or run safely on any RCAC cluster

5

```
apptainer exec fastqc_custom.sif fastqc sample_R1.fastq.gz
```

Run commands from your new image

6

Custom Apptainer Image (from a recipe)

filename: `fastqc.def`

1

```
Bootstrap: docker                # Base source for the container (Apptainer library)
From: ubuntu:22.04              # Starting image: Ubuntu 22.04 LTS

%labels                          # Metadata for documentation / version tracking
  Author Arun Seetharam
  Version v1.0
  Description "FastQC on Ubuntu 22.04"

%environment                    # Variables set every time the container runs
  export PATH=/opt/FastQC:$PATH # Add FastQC to system path
  export LC_ALL=C.UTF-8         # Ensure consistent locale handling

%post                           # Commands executed during build (inside container)
  export DEBIAN_FRONTEND=noninteractive
  apt-get update && \
  apt-get install -y wget unzip openjdk-11-jre perl && \
  apt-get clean && rm -rf /var/lib/apt/lists/*
  cd /opt && \
  wget https://www.bioinformatics.babraham.ac.uk/projects/fastqc/fastqc_v0.12.1.zip && \
  unzip fastqc_v0.12.1.zip && cd FastQC && chmod +x fastqc && \
  ln -s /opt/FastQC/fastqc /usr/local/bin/fastqc

%runscript                      # Defines default behavior for 'apptainer run'
  exec fastqc "$@"              # Passes user arguments to FastQC

%test                           # Optional: quick test after build
  fastqc --version

%help                           # Displayed with 'apptainer run-help fastqc.sif'
  Usage: apptainer exec fastqc.sif fastqc <input.fastq.gz>
```

```
apptainer build fastqc.sif fastqc.def
```

Build a `.sif` image using the recipe file.

This file can be shared or run safely on any RCAC cluster

2