New CLIMB-BIG-DATA VM Image

Dr. Anna Price

Research Software Engineer, Cardiff University

Microbiology Society Annual Conference 2022

New CLIMB VM image – What software is included?

- **Package manager:**
  - Miniconda
- **Container engines:**
  - Docker
  - Singularity
- **Workflow managers:**
  - Nextflow
  - Snakemake

Package Manager - Conda

- Cross-platform package manager that installs and manages software packages. The isolated environments that conda can build can be very useful
  - Using YAML files we can create reproducible environments
  - Software with conflicting underlying build dependencies can be isolated in different environments

Limitations of conda

- Environments are sometimes slow to resolve
- Non-deterministic dependency resolution; can be differences between platforms

Conda environments are not inherently cross-platform!

Solution: Use containers!

Container Engines – Docker and Singularity

Container engines can be used to deploy software packages in a lightweight, standalone, and reproducible environment

Software is packaged with its dependencies and is isolated from the host machine in a container

Software runs uniformly regardless of infrastructure

Easily integrated with Nextflow and Snakemake

What sort of container images are available?

- Programming languages such as Python and R
- Different Linux OS's, e.g. Ubuntu, Debian, Alpine
- Conda images are available
- Images for bioinformatics software

How to use Singularity?

How do I find Singularity containers?

- Pre-built container images for bioinformatics software can be found through resources such as biocontainers (https://biocontainers.pro/registry) hosted on Docker Hub and quay.io
- Over the coming months, CLIMB-BIG-DATA will start to release our own resource of container images hosted on quay.io
- You can build your own container images by writing a Singularity recipe/definition file

How do I run a Singularity container?

- To run a Singularity container, use the command **singularity exec**, e.g.

```
singularity exec $image.sif $command
```

Singularity – Technical Details

- Singularity was developed with HPC systems in mind (Linux only)
- More effective security than Docker
- Easier to make use of GPUs & MPI
- Singularity mounts the $HOME directory
- SIF container image files were created for easier distribution
- **Singularity can create a Singularity image from a Docker image**

Recipe -> Image -> Container

```
Bootstrap: docker
From: debian:buster

%environment
export kraken2_version=2.0.8
export PACKAGES="gcc g++ wget parallel build-essential rsync
unzip ncbi-blast+"

%post
apt-get update && apt-get install -y $PACKAGES \
&& wget
https://github.com/DerrickWood/kraken2/archive/v${kraken2_
version}-beta.tar.gz \
&& tar -xzf v${kraken2_version}-beta.tar.gz \
&& rm v${kraken2_version}-beta.tar.gz \
&& cd kraken2-${kraken2_version}-beta \
&& ./install_kraken2.sh /usr/local/bin

%runscript
exec /bin/bash "$@"
```

Singularity Recipe

Container Recipes are the instructions used to build container images, at runtime images become containers

Pre-built Images can be pulled from a container registry.

For Singularity images:
- sylabs

For Docker images:
- DockerHub
- quay.io

Singularity can pull Docker images and convert them to Singularity images

Workflow Managers – Nextflow and Snakemake

Workflow management systems can be used to create **reproducible** and **scalable** workflows

Workflows can use existing software and tools

A software analysis is defined by a process (Nextflow) or a rule (Snakemake). Processes and rules are used to build workflows

Can be used in conjunction with Conda or containers

nextflow

How to use Nextflow?

How do I find existing workflows?

- Pre-built workflows can be sourced from resources such as nf-core (https://github.com/nf-core)

How do I write my own workflows?

- Easiest to use the new syntax DSL2, which allows for a more modular way of building workflows
- Use resources such as nf-core modules (https://github.com/nf-core/modules)
- Over the coming months, CLIMB-BIG-DATA will start to release our own resource of reusable nextflow modules on GitHub which will have associated container images

Nextflow – Why DSL2 and how to organise the workflow?

- Nextflow DSL2 introduces a new syntax, allowing for **reusable sub-workflows** and the building of a library of **reusable processes**
- A library-based approach can be used where **processes are grouped into modules** by their function
- Each process can have an associated docker/singularity container

Sub-workflows and processes are reusable!

nextflow

Today's Demo

TODAY'S DEMO WILL BE FOCUSED ON SINGULARITY AND NEXTFLOW DSL2

THERE WILL BE AN INTRODUCTION TO THE BASICS OF USING SINGULARITY, AND I WILL ASSEMBLE A GENOME USING A CONTAINER FOR THE BACTERIAL ASSEMBLER SHOVILL

I WILL THEN SHOW A SHORT NEXTFLOW DSL2 WORKFLOW, CONTAINING TRIM_GALORE AND SHOVILL, WHICH WILL BE CONTAINERISED

https://github.com/annacprice/singularity-nextflow-demo

nextflow