# Clowdr Documentation

**Greg Kiar, Tristan Glatard**

**Jan 10, 2019**

# Contents:

CHAPTER 1

# Launching Local & Cluster Tasks

Manages local and cluster deployment. Ideal for development, testing, executing on local resources, or deployment on a computing cluster environment.

```
usage: clowdr local [-h] [--verbose] [--dev] [--workdir WORKDIR]
                    [--volumes VOLUMES] [--groupby GROUPBY] [--sweep SWEEP]
                    [--setup] [--cluster {slurm}] [--clusterargs CLUSTERARGS]
                    [--jobname JOBNAME] [--simg SIMG] [--user]
                    [--rerun {all,failed,incomplete}] [--run_id RUN_ID]
                    [--s3 S3] [--bids]
                    descriptor invocation provdir
```

## 1.1 Positional Arguments

**descriptor** — Local path to Boutiques descriptor for the tool you wish to run. To learn about descriptors and Boutiques, go to: https://boutiques.github.io.

**invocation** — Local path to Boutiques invocation (or directory containing multiple invocations) for the analysis you wish to run. To learn about invocations and Boutiques, go to: https://boutiques.github.io.

**provdir** — Local directory for Clowdr provenance records and other captured metadata to be stored. This directory needs to exist prior to running Clowdr.

## 1.2 Named Arguments

**--verbose, -V** — Toggles verbose output statements.

Default: False

**--dev, -d** — Launches only the first created task. This is intended for development purposes.

Default: False

| | |
|---|---|
| **--workdir, -w** | Specifies the working directory to be used by the tasks created. |
| **--volumes, -v** | Specifies any volumes to be mounted to the container. This is usually related to the path of any data files as specified in your invocation(s). |
| **--groupby, -g** | If you wish to run tasks in batches, specify the number of tasks to group here. For imperfect multiples, the last group will be the remainder. |
| **--sweep** | If you wish to perform a parameter sweep with Clowdr, you can use this flag and provide Boutiques parameter ID as the argument here. This requires: 1) the parameter exists in the provided invocation, and 2) that field contains a list of the parameter values to be used (if it is ordinarily a list, this means it must be a list of lists here). This option does not work with directories of invocations, but only single files. |
| **--setup** | If you wish to generate metadata but not launch tasks then you can use this mode.<br><br>Default: False |
| **--cluster, -c** | Possible choices: slurm<br><br>If you wish to submit your local tasks to a scheduler, you must specify it here. Currently this only supports SLURM clusters. |
| **--clusterargs, -a** | This allows users to supply arguments to the cluster, such as specifying RAM or requesting a certain amount of time on CPU. These are provided in the form of key:value pairs, and separated by commas. For example: –clusterargs time:4:00,mem:2048,account:ABC |
| **--jobname, -n** | If running on a cluster, and you wish to specify a unique identifier to appear in thesubmitted tasks, you can specify it with this flag. |
| **--simg, -s** | If the Boutiques descriptor summarizes a tool wrapped in Singularity, and the image has already been downloaded, this option allows you to specify that image file. |
| **--user, -u** | If the Boutiques descriptor summarizes a tool wrapped in Docker, toggles propagating the current user within the container.<br><br>Default: False |
| **--rerun, -R** | Possible choices: all, failed, incomplete<br><br>Allows user to re-run jobs in a previous execution that either failed or didn't finish, etc. This requires the –run_id argument to also be supplied. Three choices are: 'all' to re-run all tasks, 'failed' to re-run tasks which finished with a non-zero exit-code, 'incomplete' to re-run tasks which have not yet indicated job completion. While the descriptor and invocations will be adopted from the previous executions, other options such as clusterargs or volume can be set to different values, if they were the source or errors. Pairing the incomplete mode with the –dev flag allows you to walk through your dataset one group at a time. |
| **--run_id** | Pairs with –rerun. This ID is the directory within the supplied provdir which contains execution you wish to relaunch. These IDs/directories are in the form: year-month-day_hour-minute-second-8digitID. |
| **--s3** | Amazon S3 bucket and path for remote data. Accepted in the format: s3://{bucket}/{path} |
| **--bids, -b** | Indicates that the tool being launched is a BIDS app. BIDS is a data organization format in neuroimaging. For more information about this, go to https://bids.neuroimaging.io. |

Default: False

CHAPTER 2

# Launching Cloud Tasks

Manages cloud deployment. Ideal for running jobs at scale on data stored in Amazon Web Services S3 buckets (or similar object store).

```
usage: clowdr cloud [-h] [--verbose] [--dev] [--region REGION] [--sweep SWEEP]
                    [--bids]
                    descriptor invocation provdir s3 {aws} credentials
```

## 2.1 Positional Arguments

**descriptor** Local path to Boutiques descriptor for the tool you wish to run. To learn about descriptors and Boutiques, go to: https://boutiques.github.io.

**invocation** Local path to Boutiques invocation (or directory containing multiple invocations) for the analysis you wish to run. To learn about invocations and Boutiques, go to: https://boutiques.github.io.

**provdir** Local directory for Clowdr provenance records and other captured metadata to be stored. This directory needs to exist prior to running Clowdr.

**s3** Amazon S3 bucket and path for remote data. Accepted in the format: s3://{bucket}/{path}

**cloud** Possible choices: aws

Specifies which cloud endpoint you'd like to use. Currently, only AWS is supported.

**credentials** Your credentials file for the resource.

## 2.2 Named Arguments

**--verbose, -V** Toggles verbose output statements.

| | |
|---|---|
| | Default: False |
| **--dev, -d** | Launches only the first created task. This is intended for development purposes.<br><br>Default: False |
| **--region, -r** | The Amazon region to use for processing. |
| **--sweep** | If you wish to perform a parameter sweep with Clowdr, you can use this flag and provide Boutiques parameter ID as the argument here. This requires: 1) the parameter exists in the provided invocation, and 2) that field contains a list of the parameter values to be used (if it is ordinarily a list, this means it must be a list of lists here). This option does not work with directories of invocations, but only single files. |
| **--bids, -b** | Indicates that the tool being launched is a BIDS app. BIDS is a data organization format in neuroimaging. For more information about this, go to https://bids.neuroimaging.io.<br><br>Default: False |

CHAPTER 3

# Sharing Your Analysis

```
usage: clowdr share [-h] [--debug] [--verbose] provdir
```

## 3.1 Positional Arguments

**provdir** Local or S3 directory where Clowdr provenancerecords and metadata are stored. This path was returned by running either clowdr cloud or clowdr local. This can also be a clowdr-generated summary file.

## 3.2 Named Arguments

**--debug, -d** Toggles server messages and logging. This is intended for development purposes.

Default: False

**--verbose, -V** Toggles verbose output statements.

Default: False

CHAPTER 4

# Manually Running Tasks

```
usage: clowdr task [-h] [--verbose] [--provdir PROVDIR] [--local]
                   [--workdir WORKDIR] [--volumes VOLUMES]
                   tasklist [tasklist ...]
```

## 4.1 Positional Arguments

**tasklist** One or more Clowdr-created task.json files summarizing the jobs to be run. These task files are created by one of clowdr cloud or clowdr local.

## 4.2 Named Arguments

**--verbose, -V** Toggles verbose output statements.

Default: False

**--provdir, -p** Local or directory where Clowdr provenance records and metadata will be stored. This is optional here because it will be stored by default in a temporary location and moved, unless this is specified.

**--local, -l** Flag indicator to identify whether the task is being launched on a cloud or local resource. This is important to ensure data is transferred off clouds before shut down.

Default: False

**--workdir, -w** Specifies the working directory to be used by the tasks created.

**--volumes, -v** Specifies any volumes to be mounted to the container. This is usually related to the path of any data files as specified in your invocation(s).

CHAPTER 5

# Clowdr Python Interface

## 5.1 clowdr package

### 5.1.1 Subpackages

#### clowdr.controller package

#### Submodules

#### clowdr.controller.launcher module

clowdr.controller.launcher.**configureResource**(*endpoint*, *auth*, ***kwargs*)

#### clowdr.controller.metadata module

clowdr.controller.metadata.**bidsTasks**(*clowdrloc*, *taskdict*)

bidsTask Scans through BIDS app fields for creating more tasks than specified.

**clowdrloc** [str] Path for storing Clowdr intermediate files and outputs

**taskdict** [str] Dictionary of the tasks (pre-BIDS-ification)

**tuple: (list, list)** The task dictionary JSONs, and associated Boutiques invocation files.

clowdr.controller.metadata.**consolidateTask**(*tool*, *invocation*, *clowdrloc*, *dataloc*, ***kwargs*)

consolidate Creates Clowdr task JSON files which summarize all associated metadata

**tool** [str] Path to a boutiques descriptor for the tool to be run

**invocation** [str] Path to a boutiques invocation for the tool and parameters to be run

**clowdrloc** [str] Path for storing Clowdr intermediate files and outputs

**dataloc** [str] Path for accessing input data

****kwargs** [dict] Arbitrary keyword arguments (i.e. {'verbose': True})

**tuple: (list, list)** The task dictionary JSONs, and associated Boutiques invocation files.

clowdr.controller.metadata.**prepareForRemote**(*tasks*, *tmploc*, *clowdrloc*)
prepare Scans through BIDS app fields for creating more tasks than specified.

**clowdrloc** [str] Path for storing Clowdr intermediate files and outputs

**taskdict** [str] Dictionary of the tasks (pre-BIDS-ification)

**tuple: (list, list)** The task dictionary JSONs, and associated Boutiques invocation files.

clowdr.controller.metadata.**sweepTasks**(*taskdicts*, *invocations*, *sweep_param*)

#### Module contents

#### clowdr.endpoint package

#### Submodules

#### clowdr.endpoint.AWS module

**class** clowdr.endpoint.AWS.**AWS**(*auth*)
Bases: *clowdr.endpoint.remote.Endpoint*

**configureBatch**(***kwargs*)

**configureIAM**(***kwargs*)

**launchJob**(*taskloc*)

**setCredentials**(***kwargs*)

**startSession**()

#### clowdr.endpoint.remote module

**class** clowdr.endpoint.remote.**Endpoint**(*auth*)
Bases: object

#### Module contents

### 5.1.2 Submodules

### 5.1.3 clowdr.driver module

clowdr.driver.**cloud**(*descriptor*, *invocation*, *provdir*, *s3*, *cloud*, *credentials*, ***kwargs*)
Launches a pipeline locally at scale through Clowdr.

**descriptor** [str] Path to a boutiques descriptor for the tool to be run

**invocation** [str] Path to a boutiques invocation for the tool and parameters to be run

**provdir** [str] Path on S3 for storing Clowdr intermediate files and outputs

**s3** [str] Path on S3 for accessing input data

**cloud** [str] Which endpoint to use for deployment

**credentials** [str] Credentials for Amazon with access to dataloc, clowdrloc, and Batch

****kwargs** [dict] Arbitrary keyword arguments (i.e. {'verbose': True})

**int** The exit-code returned by the task being executed

clowdr.driver.**local**(*descriptor*, *invocation*, *provdir*, *backoff_time=36000*, *sweep=[]*, *verbose=False*, *workdir=None*, *simg=None*, *rerun=None*, *run_id=None*, *volumes=None*, *s3=None*, *cluster=None*, *jobname=None*, *clusterargs=None*, *dev=False*, *groupby=None*, *user=False*, *setup=False*, ***kwargs*)

cluster Launches a pipeline locally through the Clowdr wrappers.

**tool** [str] Path to a boutiques descriptor for the tool to be run

**invocation** [str] Path to a boutiques invocation for the tool and parameters to be run

**clowdrloc** [str] Path for storing Clowdr intermediate files and outputs

**dataloc** [str] Path for accessing input data. If local, provide the hostname and optionally a path. If on S3, provide an S3 path.

**cluster** [str] Scheduler on the cluster being used. Currently, the only supported mode is slurm.

****kwargs** [dict] Arbitrary keyword arguments. Currently supported arguments: - account : str

Account for the cluster scheduler

- **jobname** [str] Base-name for the jobs as they will appear in the scheduler
- **backoff_time: int** Time limit for wait times when resubmitting jobs to a scheduler
- **verbose** [bool] Toggle verbose output printing
- **dev** [bool] Toggle dev mode (only runs first execution in the specified set)

Additionally, transfers all keyword arguments accepted by both of "controller.metadata.consolidateTask" and "task.TaskHandler"

**int** The exit-code returned by the task being executed

clowdr.driver.**main**(*args=None*)

clowdr.driver.**makeparser**()

Command-line API wrapper for Clowdr as a CLI, not Python API. For information about the command-line wrapper and arguments it accepts, please try running "clowdr –help".

**args: list** List of all command-line arguments being passed.

**int** The exit-code returned by the driver.

clowdr.driver.**runtask**(*tasklist*, ***kwargs*)

clowdr.driver.**share**(*provdir*, ***kwargs*)

Launches a simple web server which showcases all runs at the clowdrloc.

**provdir** [str] Path with Clowdr metdata files (returned from "local" and "deploy")

****kwargs** [dict] Arbitrary keyword arguments (i.e. {'verbose': True})

None

### 5.1.4 clowdr.server module

### 5.1.5 clowdr.task module

**class** clowdr.task.**TaskHandler**(*taskfile*, ***kwargs*)

Bases: object

**execWrapper**(*sender*)

**manageTask**(*taskfile*, *provdir=None*, *verbose=False*, ***kwargs*)

**monitor**(*target*, ***kwargs*)

**provLaunch**(*options*, ***kwargs*)

### 5.1.6 clowdr.utils module

clowdr.utils.**backoff**(*function*, *posargs*, *optargs*, *backoff_time=36000*, ***kwargs*)

clowdr.utils.**get**(*remote*, *local*, ***kwargs*)

clowdr.utils.**getContainer**(*savedir*, *container*, ***kwargs*)

clowdr.utils.**post**(*local*, *remote*, ***kwargs*)

clowdr.utils.**randstring**(*k*)

clowdr.utils.**remove**(*local*)

clowdr.utils.**splitS3Path**(*path*)

clowdr.utils.**truepath**(*path*)

CHAPTER 6

# Indices and tables

- genindex
- search

# Python Module Index

## C

# Index