godocker Documentation

Release 1.0

Olivier Sallou

June 17, 2015

Contents

Contents:

CHAPTER 1

Go-d-docker IGoDockerPlugin reference

class godocker.IGoDockerPlugin.**IGoDockerPlugin**

Base plugin reference

get_group_usage(*group_id*)

Get cpu/ram/time usage for last period for a group

Parameters **group_id** (*int*) – group identifier

Returns list (cpu,ram,duration)

get_name()

Get name of plugin

get_running_tasks(*start=0*, *stop=-1*)

Get all tasks running

Parameters

- **start** (*int*) – first task index
- **stop** (*int*) – last task index (-1 = all)

get_users(*user_id_list*)

Get users matching ids in user_id_list

Parameters **user_id_list** (*list*) – list containing the id of users

Returns list of users

is_task_over(*task_id*)

Checks if task is over

Parameters **task_id** (*int*) – task identifier

Returns bool

is_task_running(*task_id*)

Checks if task is running

Parameters **task_id** (*int*) – task identifier

Returns bool

is_task_running_or_over(*task_id*)

Checks if task is running or over

Parameters **task_id** (*int*) – task identifier

Returns bool

kill_tasks(*task_list*)

Set input tasks in kill queue

Parameters **task_list** (*list*) – list of tasks

set_config(*cfg*)

Set configuration

set_logger(*logger*)
Set logger for logging

CHAPTER 2

Go-d-docker IExecutorPlugin reference

class godocker.iExecutorPlugin.**IExecutorPlugin**

Executor plugins interface

close()

Request end of executor if needed

features()

Get supported features

Returns list of features within ['kill', 'pause','resources.port']

get_mapping_port(*host*, *task*)

Get a port mapping for interactive tasks

Parameters

- **host** (*str*) – hostname of the container
- **task** (*int*) – task

Returns available port

kill_task(*task*)

Kills a running task

Parameters **tasks** (*Task*) – task to kill

Returns (Task, over) over is True if task could be killed

open(*proc_type=None*)

Request start of executor if needed

Parameters **proc_type** – type of process requesting open, 0 for scheduler, 1 for watcher

resume_task(*task*)

Resume/restart a task

Parameters **tasks** (*Task*) – task to resumed

Returns (Task, over) over is True if task could be resumed

run_all_tasks(*tasks*, *callback=None*)

Execute all task list on executor system, all tasks must be executed together

Parameters

- **tasks** (*list*) – list of tasks to run
- **callback** (*func(running list,rejected list)*) – callback function to update tasks status (running/rejected)

Returns tuple of submitted and rejected/errored tasks

run_tasks(*tasks*, *callback=None*)

Execute task list on executor system

Parameters

- **tasks** (*list*) – list of tasks to run
- **callback** (*func(running list,rejected list)*) – callback function to update tasks status (running/rejected)

Returns tuple of submitted and rejected/errored tasks

suspend_task(*task*)

Suspend/pause a task

Parameters **tasks** (*Task*) – task to suspend

Returns (Task, over) over is True if task could be suspended

watch_tasks(*task*)

Get task status

Parameters

- **task** (*Task*) – current task
- **over** (*bool*) – is task over

CHAPTER 3

Go-d-docker ISchedulerPlugin reference

class godocker.iSchedulerPlugin.**ISchedulerPlugin**

Scheduler plugins interface

get_bounds_usage(*usages*)

Gets min and max usage

Parameters **usages** (*list*) – input usages list of dict{ total_time, total__cpu, total_ram }

Returns

dict { max_time, min_time, max_cpu, min_cpu, max_ram, min_ram }

get_bounds_waiting_time(*tasks*)

Gets min and max waiting time for tasks

Parameters **tasks** (*list*) – list of pending tasks

Returns list (max_time, min_time)

get_project_prio(*project_id*)

Get project priority (between 0 and 1)

get_user_prio(*user_id*)

Get user priority (between 0 and 1)

get_user_usage(*identifier*, *key='user'*)

Parameters

- **identifier** (*str*) – user/group identifier
- **key** (*str*) – user or group

Returns dict { total_time, total__cpu, total_ram, }

schedule(*tasks*)

Schedule list of tasks to be ran according to user list

Returns list of sorted tasks

CHAPTER 4

Go-d-docker IStorage reference

class godocker.IStorage.**IStorage**(*cfg*)

Storage base interface

__weakref__

list of weak references to the object (if defined)

add_file(*task*, *name*, *content*)

Add content to a file with content in task directory

Parameters

- **task** (*dict*) – current task
- **name** (*str*) – name of the file
- **content** (*str*) – file content

Returns path to the file

clean(*task*)

Cleanup task directory

Parameters **task** (*dict*) – current task

get_task_dir(*task*)

Get directory where task files are written

CHAPTER 5

Go-d-docker IAuthPlugin reference

class godocker.iAuthPlugin.**IAuthPlugin**

ACL plugins interface

bind_api (*apikey*)

Check api key and return user info (same than bind_credentials)

bind_credentials (*login*, *password*)

Check user credentials and return user info

Returns a user dict:

{ 'id' : userId, 'uidNumber': systemUserid, 'gidNumber': systemGroupid, 'email': userEmail, 'homeDirectory': userHomeDirectory }

can_run (*task*)

Check if task can run (according to user etc...). If return False, then task is rejected

Parameters **task** (*Task*) – task to schedule

Returns bool

get_user (*login*)

Get user information

Returns a user dict:

{ 'id' : userId, 'uidNumber': systemUserid, 'gidNumber': systemGroupid, 'email': userEmail, 'homeDirectory': userHomeDirectory }

get_volumes (*user*, *requested_volumes*, *root_access=False*)

Returns a list of container volumes to mount, with acls, according to user requested volumes. Returned volumes should set real path to requested volumes, possibliy changed requested acl.

Parameters

- **user** (*dict*) – User returned by bind_credentials or bind_api
- **requested_volumes** (*list*) – list of volumes user expects to be mounted in container
- **root_access** (*bool*) – user request root access to the container

Returns list of volumes to mount

Volumes path are system specific and this method must be implemented according to each system.

If 'mount' is None, then mount path is the same than original directory.

requested_volumes looks like:

volumes: [

{ 'name': 'home', 'acl': 'rw'

}, { 'name': 'omaha',

'acl': 'rw'

}, { 'name': 'db',

```
        'acl': 'ro'
    },
]
```

Return volumes:

```
    volumes: [
      { 'name': 'home', 'acl': 'rw', 'path': '/home/mygroup/myuserid', 'mount':
        '/home/myuserid'
      }, { 'name': 'omaha',
        'acl': 'ro', 'path': '/mynfsshare/myuserid' 'mount': None
      }, { 'name': 'db',
        'acl': 'ro', 'path': '/db', 'mount': None
      },

    ]
```

CHAPTER 6

Go-d-docker Utils reference

godocker.utils.**is_array_child_task**(*task*)
Checks if input task is an array child task

Returns bool

godocker.utils.**is_array_task**(*task*)
Checks if input task is an array task eg a parent task

Returns bool

CHAPTER 7

Indices and tables

- genindex
- modindex
- search

g

Symbols

A

B

C

F

G

I

K

O

R

S

W