Technical document on ConPaaS services: ConPaaS MySQL Server

Aleš Černivec

November 18, 2011

Contents

1 Introduction

2 Architecture

3 ConPaaS MySQL Server Manager

Module `conpaas.mysql.server.manager.internals` contains internals of the ConPaaS MySQL Server. ConPaaS MySQL Server consists of several nodes with different roles.

- Manager node
- Agent node(s)
  - Master
  - Slave(s)

platform: Linux, Debian
synopsis: Internals of ConPaaS MySQL Servers.
moduleauthor: Ales Cernivec <ales.cernivec@xlab.si>

`conpaas.mysql.server.manager.internals.add_nodes(kwargs)`

Description:

HTTP POST method. Creates new node and adds it to the list of existing nodes in the manager. A role of new node can be one of: agent, manager. Currently only agent is supported. It makes internal call to `createServiceNodeThread()`.

Parameters:

kwargs – string describing a function (agent).

Returns:

HttpJsonResponse - JSON response with details about the node.

Raises:

ManagerException

`conpaas.mysql.server.manager.internals.remove_nodes(params)`

Description:

HTTP POST method. Deletes specific node from a pool of agent nodes. Node deleted is given by `{'serviceNodeId':id}`.

Parameters:

kwargs –string identifying a node.

Returns:

HttpJsonResponse - HttpJsonResponse - JSON response with details about the node. OK if everything went well.

Raises:

ManagerException if something went wrong. It contains a detailed description about the error.

`conpaas.mysql.server.manager.internals.list_nodes()`

Description:

HTTP GET method. Uses `IaaSClient.listVMs()` to get list of all service nodes. For each service node it checks if it is in servers list. If some of them are missing they are removed from the list. Returns list of all service nodes.

Parameters:

Returns:

HttpJsonResponse - JSON response with the list of services: `{ 'serviceNode': [<a list of ids>]})`

Raises:

HttpErrorResponse

`conpaas.mysql.server.manager.internals.get_node_info()`

Description:

HTTP GET method. Gets info of a specific node.

Parameters:

param (str) – serviceNodeId is a VMID of an existing service node.

Returns:

HttpJsonResponse - JSON response with details about the node: : `{'serviceNode':{'id': serviceNode.vmid,'ip': serviceNode.ip,'isRunningMySQL': serviceNode.isRunningMySQL}}`.

Raises:

ManagerException

`conpaas.mysql.server.manager.internals.get_service_info()`

Description:

HTTP GET method. Returns the current state of the manager.

Parameters:

param (str) – serviceNodeId is a VMID of an existing service node.

Returns:

HttpJsonResponse - JSON response with the description of the state.

Raises:

ManagerException

`conpaas.mysql.server.manager.internals.set_up_replica_master()`

Description:

HTTP POST method. Sets up a replica master node.

Parameters:

id – new replica master id.

Returns:

HttpJsonResponse - JSON response with details about the new node. ManagerException if something went wrong.

Raises:

ManagerException

`conpaas.mysql.server.manager.internals.set_up_replica_slave()`

Description:

HTTP POST method. Sets up a replica master node.

Parameters:

id – new replica slave id.

Returns:

HttpJsonResponse - JSON response with details about the new node. ManagerException if something went wrong.

Raises:

ManagerException

`conpaas.mysql.server.manager.internals.shutdown()`

Description:

HTTP POST method. Shuts down the manager service.

Parameters:

id – new replica slave id.

Returns:

HttpJsonResponse - JSON response with details about the status of a manager node: :py:attr‘S_EPILOGUE‘. ManagerException if something went wrong.

Raises:

ManagerException

`conpaas.mysql.server.manager.internals.get_service_performance()`

Description:

HTTP GET method. Placeholder for obtaining performance metrics.

Parameters:

kwargs (dict) – Additional parameters.

Returns:

HttpJsonResponse – returns metrics

Raises:

4 Conclusion

References