File ▾ Edit ▾ Insert ▾ Generate Server ▾ Generate Client ▾

API 1.0.0 OAS3

Interface for communicating with the Health-Monitor service. Its functionalities can be organized in the following set of operations:

- Operations for service management: permits to add and remove docker hosts to the system and to uninstall it
- Operations for container management: permits to see the managed containers, their status and to change the container manager configuration
- Operations for testing purpose: permits to test the service with configurable attacks to the containers

Servers

http://172.16.3.167:8080/healthmonitor - Unipi Virtual Machine offered by the course. A VPN connection

default

GET `/containers` Gives all the managed containers by the service

It doesn't require any parameter and gives back a set of lists each one describing the containers inside a particular managed docker host. The response will contain for each docker host its status and a list of all the containers with information about their status and the current packet loss measured

Parameters Try it out

No parameters

Responses

Code	Description	Links

Code	Description	Links

200	Correct response
	
	Media type
	application/json
	Controls `Accept` header.
	
	Example Value Schema
	
	{
	"command": "ok",
	"description": [
	{
	"address": "172.16.3.167",
	"content": {}
	}
	]
	}
		No links

400	Bad Request
	
	Media type
	application/json
	
	Example Value Schema
	
	{
	"command": "ok",
	"type": "INVALID_REQUEST",
	"description": "Request correctly executed"
	}
		No links

POST `/containers` Change the threshold configured into each manager.

Change the threshold configured into the docker managers. Each manager have a personal threshold used to identify if the packet loss percentage is too high and the container needs to be restarted.

Parameters Try it out

No parameters

Request body required application/json

The request requires the threshold value to be set into the managers. The value is a percentage so it needs to be between 0 and 100 to be accepted.

Example Value Schema

```
{
  "threshold": 60
}
```

Responses

Code	Description	Links
200	Correct Response Media type application/json Controls `Accept` header. **Example Value** Schema `{ "command": "ok", "content": [ {} ] }`	*No links*
400	Bad Request Media type application/json **Example Value** Schema `{ "command": "ok", "type": "INVALID_REQUEST", "description": "Request correctly executed" }`	*No links*

GET `/containers/{address}` Get the status of the selected docker host and a list of all the managed containers with their status and the current packet loss percentage measured

It doesn't require any parameter and gives back a set of lists each one describing the containers inside a particular managed docker host. The response will contain for each docker host its status and a list of all the containers with information about their status and the current packet loss measured

Parameters

Try it out

Name	Description
address * required `string` *`(path)`*	172.16.3.167

Responses

Code	Description	Links
200	Correct Response Media type application/json Controls `Accept` header. **Example Value** Schema `{ "command": "ok", "description": { "address": "172.16.3.167", "content": {} } }`	*No links*
400	Bad Request Media type application/json **Example Value** Schema `{ "command": "ok", "type": "INVALID_REQUEST", "description": "Request correctly executed" }`	*No links*

DELETE `/containers/{address}` Remove a registered docker host into the service

Uninstall the selected manager from the machine. The system has already stored the password but its required in order to secure the operation and prevent that users will disable the managers

Parameters Try it out

Name	Description
address * required `string` *`(path)`*	172.16.3.167

Responses

Code	Description	Links
200	Correct Response Media type application/json Controls `Accept` header. **Example Value** Schema `{ "command": "ok", "content": "Request correctly executed" }`	*No links*
400	Bad Request Media type application/json **Example Value** Schema `{ "command": "ok", "type": "INVALID_REQUEST", "description": "Request correctly executed" }`	*No links*

PUT `/containers/{address}` Add a new docker host into the service

Install a new manager into the the selected machine. The operations requires the root password of the destination machine and that a ssh connection is present on the default port 23

Parameters

Try it out

Name	Description
address * required `string` *(path)*	172.16.3.167

Request body required

application/json

The operation requires the password of the root user of the destination machine

Example Value Schema

```
{
  "password": "root"
}
```

Responses

Code	Description	Links
200	Correct Response Media type application/json Controls `Accept` header. **Example Value** Schema `{ "command": "ok", "content": "Request correctly executed" }`	*No links*

Code	Description	Links

400	Bad Request
	
	Media type
	application/json
	
	Example Value Schema
	
	{
	"command": "ok",
	"type": "INVALID_REQUEST",
	"description": "Request correctly executed"
	}
		No links

POST `/containers/{address}` Change the threshold configured into the selected manager.

Change the threshold configured into the selected docker manager. Each manager have a personal threshold used to identify if the packet loss percentage is too high and the container needs to be restarted.

Parameters Try it out

Name	Description
address * required `string` ***(path)***	172.16.3.167

Request body required application/json

The operation requires a threshold parameters to be set into the machine. The threshold must be between 0 and 100

Example Value Schema

```
{
  "threshold": 70
}
```

Responses

Code	Description	Links

200	Correct Response
	
	Media type
	application/json
	Controls `Accept` header.
	
	Example Value Schema
	
	{
	"command": "ok",
	"content": "Request correctly executed"
	}
		No links

400	Bad Request
	
	Media type
	application/json
	
	Example Value Schema
	
	{
	"command": "ok",
	"type": "INVALID_REQUEST",
	"description": "Request correctly executed"
	}
		No links

GET **/containers/{address}/{containerID}** Gives information about the selected container

The request returns the status and the packet loss of the selected container

Parameters Try it out

Name	Description
address * required **string** ***(path)***	172.16.3.167
containerID * required **string** ***(path)***	asd12pomfa1g

Responses

Code	Description	Links

200	Correct response
	
	Media type
	application/json
	Controls `Accept` header.
	
	Example Value Schema
	
	{
	"command": "ok",
	"description": {
	"address": "172.16.3.167",
	"content": {}
	}
	}
		No links

400	Bad Request
	
	Media type
	application/json
	
	Example Value Schema
	
	{
	"command": "ok",
	"type": "INVALID_REQUEST",
	"description": "Request correctly executed"
	}
		No links

PUT /containers/{address}/{containerID} Add a container to the service

Add a previously ignored container to the service. By default all the containers are added to the service

Parameters

Try it out

Name	Description
address * required **string** ***(path)***	172.16.3.167

Name	Description
containerID * required `string` *`(path)`*	asd12pomfa1g

Responses

Code	Description	Links
200	Correct response Media type application/json Controls `Accept` header. **Example Value** Schema `{ "command": "ok", "content": "Request correctly executed" }`	*No links*
400	Bad Request Media type application/json **Example Value** Schema `{ "command": "ok", "type": "INVALID_REQUEST", "description": "Request correctly executed" }`	*No links*

DELETE **`/containers/{address}/{containerID}`** Remove a container from the service

Ignore a container from the service management. By default all the containers are added to the service

Parameters

Try it out

Name	Description
address * required `string` *(path)*	172.16.3.167
containerID * required `string` *(path)*	asd12pomfa1g

Responses

Code	Description	Links
200	Correct response Media type application/json Controls `Accept` header. **Example Value** Schema `{ "command": "ok", "content": "Request correctly executed" }`	*No links*
400	Bad Request Media type application/json **Example Value** Schema `{ "command": "ok", "type": "INVALID_REQUEST", "description": "Request correctly executed" }`	*No links*

POST `/test` Change all the antagonists configuration

Change the configuration of all the antagonist. The configuration consists of five different parameters:

- heavy: defines the probability that the attacker will choose to attack a container

- balance: defines the probability that the attackers will perform a packet loss attack or a shutdown attack
- loss: defines the mean packet loss value introduced during an attack(the value will be the mean of normal distribution)
- duration: defines the duration of a packet loss attack to a container. The value will be the lambda of an exponential distribution
- frequency: defines the interleaving time of each attacker between the possibility to perform an attack

Parameters Try it out

No parameters

Request body required application/json

The request body can contain this fields:

- heavy: defines the probability that the attacker will choose to attack a container. It's a value between 0 and 100
- balance: defines the probability that the attackers will perform a packet loss attack or a shutdown attack. It's a value between 0 and 100, the more the value tends to 100 the more the probability is to perform a shutdown attack
- loss: defines the mean packet loss value introduced during an attack(the value will be the mean of normal distribution). It's a value between 0 and 100
- duration: defines the duration of a packet loss attack to a container. The value will be the lambda of an exponential distribution. The value is expressed in seconds
- frequency: defines the interleaving time of each attacker between the possibility to perform an attack . The value is expressed in seconds

If a field is missing simply it will not be changed, but no errors will be raised

Example Value Schema

```
{
  "heavy": 25,
  "balance": 50,
  "loss": 70,
  "duration": 5,
  "frequency": 3
}
```

Responses

Code	Description	Links

Code	Description	Links
200	Correct response Media type application/json Controls `Accept` header. **Example Value** Schema `{ "command": "ok", "content": [ {} ] }`	*No links*
400	Bad Request Media type application/json **Example Value** Schema `{ "command": "ok", "type": "INVALID_REQUEST", "description": "Request correctly executed" }`	*No links*

PUT `/test` Starts all the antagonists

Into each docker manager is linked an antagonist able to attack all the container into the local docker host. Starting the antagonist will perform an attack on all the containers with the last configuration provided. If no configuration has been provided than a default configuration will be used.

Parameters Try it out

No parameters

Responses

Code	Description	Links

Code	Description	Links
200	Correct Response Media type application/json Controls `Accept` header. **Example Value** Schema `{ "command": "ok", "content": [ {} ] }`	*No links*
400	Bad Request Media type application/json **Example Value** Schema `{ "command": "ok", "type": "INVALID_REQUEST", "description": "Request correctly executed" }`	*No links*

DELETE `/test` Stop all the antagonists

The function will stop all the attackers into all the docker host. A started attack needs to be stop otherwise it will continue forever.

Parameters Try it out

No parameters

Responses

Code	Description	Links

Code	Description	Links
200	Correct Response Media type application/json Controls `Accept` header. **Example Value** Schema `{ "command": "ok", "content": "Request correctly executed" }`	*No links*
400	Bad Request Media type application/json **Example Value** Schema `{ "command": "ok", "type": "INVALID_REQUEST", "description": "Request correctly executed" }`	*No links*

POST `/test/{address}` Change an antagonist configuration

Change the configuration of the selected antagonist. The configuration consists of five different parameters:

- heavy: defines the probability that the attacker will choose to attack a container
- balance: defines the probability that the attackers will perform a packet loss attack or a shutdown attack
- loss: defines the mean packet loss value introduced during an attack(the value will be the mean of normal distribution)
- duration: defines the duration of a packet loss attack to a container. The value will be the lambda of an exponential distribution
- frequency: defines the interleaving time of each attacker between the possibility to perform an attack

Parameters Try it out

Name	Description
address * required `string` *(path)*	172.16.3.167

Request body required

application/json

The request body can contain this fields:

- heavy: defines the probability that the attacker will choose to attack a container. It's a value between 0 and 100
- balance: defines the probability that the attackers will perform a packet loss attack or a shutdown attack. It's a value between 0 and 100, the more the value tends to 100 the more the probability is to perform a shutdown attack
- loss: defines the mean packet loss value introduced during an attack(the value will be the mean of normal distribution). It's a value between 0 and 100
- duration: defines the duration of a packet loss attack to a container. The value will be the lambda of an exponential distribution. The value is expressed in seconds
- frequency: defines the interleaving time of each attacker between the possibility to perform an attack . The value is expressed in seconds

If a field is missing simply it will not be changed, but no errors will be raised

Example Value Schema


```
{
  "heavy": 25,
  "balance": 50,
  "loss": 70,
  "duration": 5,
  "frequency": 3
}
```


Responses

Code	Description	Links
200	Correct response Media type application/json Controls `Accept` header. **Example Value** Schema `{ "command": "ok", "content": "Request correctly executed" }`	*No links*

Code	Description	Links

400	Bad Request
	
	Media type
	application/json
	
	Example Value Schema
	
	{
	"command": "ok",
	"type": "INVALID_REQUEST",
	"description": "Request correctly executed"
	}
		No links

PUT `/test/{address}` Start an antagonist

Into each docker manager is linked an antagonist able to attack all the container into the local docker host. Starting the antagonist will perform an attack on all the containers with the last configuration provided. If no configuration has been provided than a default configuration will be used.

Parameters Try it out

Name	Description
address * required `string` ***(path)***	172.16.3.167

Responses

Code	Description	Links

Code	Description	Links
200	Correct Response Media type application/json Controls `Accept` header. **Example Value** Schema `{ "command": "ok", "content": "Request correctly executed" }`	*No links*
400	Bad Request Media type application/json **Example Value** Schema `{ "command": "ok", "content": "Request correctly executed" }`	*No links*

DELETE `/test/{address}` Stop an antagonist

The function will stop the selected antagonist to attack the local docker host. A started attack needs to be stop otherwise it will continue forever.

Parameters Try it out

Name	Description

address * required
`string`
`(path)`
	172.16.3.167

Responses

Code	Description	Links
200	Correct Response Media type application/json Controls `Accept` header. **Example Value** Schema `{ "command": "ok", "content": "Request correctly executed" }`	*No links*
400	Bad Request Media type application/json **Example Value** Schema `{ "command": "ok", "content": "Request correctly executed" }`	*No links*

Schemas

CorrectResponse

```
{
  command    string
             example: ok

             type of results: ok or error

  content    {
             description:
                          Brief description of the operation
                          result

             }
             example: Request correctly executed
}
```

CorrectMultiResponse {

command string
example: ok

type of results: ok or error

content [{

}]

}

BadResponse {

command string
example: ok

type of results: ok or error

type string
example: INVALID_REQUEST

short description of the error

description string
example: Request correctly executed

Brief description of the operation result

}

AllContainersResponse {

command string
example: ok

type of results: ok or error

description [{
address string
example: 172.16.3.167

the IPv4 address of the selected docker host

content {...}

}]

}

ContainerResponse

```
{
  command          string
                   example: ok

                   type of results: ok or error

  description
                     {
                     address            string
                                        example: 172.16.3.167

                                        the IPv4 address of the selected
                                        docker host

                     content
                                          {

                                        }

                   }

}
```