Test Cloud API Design V0.0.1

# Introduction

The test cloud API is designed to provide a RESTful interface for managing proxmox-based virtualization cluster. It allows users to create/delete/start/stop virtual machines, manage networks, reserve/release IP and related issues.

# Base URL

## Test Cloud API:

https://{test.forticloud.com}/v1/

## DNS name:

(To be decided)

# Request IDs

Users can specify the global request ID in the request header. Users can also receive the request ID in the **response header**.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Direction | In | Type | Description |
| X-Request-Id  (Optional) | Request | header | string | The global request ID, which is a unique common ID for tracking each request in test cloud. The format of the global request ID must be req- + UUID (UUID4). If not in accordance with the format, it is ignored. It is associated with the request and appears in the log lines for that request.  Duplicated Message:  {      "code": 1002,      "message": "Sent duplicated request.",      "data": {      "vmid": "3dd756ee569946b5aa374eb836f2b0d9"      }  }  HTTP code  StatusInternalServerError = 500 |
| X-Cloud-Id | Response | header | string | The local request ID, which is a unique ID generated automatically for tracking each request to cloud-api. It is associated with the request and appears in the log lines for that request. |
| X-Request-Id | Response | header | string | Will be included in the response only if it is contained in the request header. |

# Authentication

Authentication for this API is performed using API token. Users must obtain an API token by asking promxmox cloud Administrator. To authenticate, include the API key in the X-API-token header of the HTTP request.

X-API-TOKEN: <API\_TOKEN>

# Endpoints

## 1 Spawn a virtual machine

### Brief Description

Create a virtual machine as per the requested template, selected flavor (CPU/Memory), networks and disks. The cloud will choose the best matched host to spawn it which we call its procedure as *Schedule*.

When you create a server, the response shows only the server ID. You can get additional attributes through subsequent GET requests on the server. You can get the progress of the operations if the spawn procedure is ongoing, and the detailed attributes once it is fully prepared.

### HTTP Method:

POST

### Endpoint:

/vm/

### Parameters:

#### Header parameter:

Content-Type: application/json

#### Body parameter:

Examples:

{

    "name": "KL-VM-WIN-2", //The length limit is 200 ascii character

"templateId": 9102,  //Leave as omit if the type is FortiCSP/FortiGate

    "osFamily": "windows" //Will open the serial port for the VM and execute the initial command remotely; Close the serial port after the execution.

    "osVersion": "Windows-XP-SP2-x86", //Like "1.0.0\_b0025" when the osFamily is " FortiCSP"

    "cpu": 4,

    "ram": 8192, //MiB

    "disks": [

        {

            "index": 0,

            "size": 2350  //MiB

        },

        {

            "index": 1,

            "size": 10240  //MiB

        }

],

    "nics": [

        {

            "index": 0,

            "netId": 1 //mgmt

        },

        {

            "index": 1,

            "netId": 3 //188

        }

    ],

    "metadata": {

        "user": "administrator",

        "password": "Forti1@#",

        "sshKey": "keys",

        "tags": {

            "env1": "testbed1",

            "env2": "testbed2"

        }

    }

}

### Response Body:

{

    "code": 200,

    "message": "OK",

    "data": {

        "vmid"："e9ef557798d84f0faae0663b7f1ba96e"

}

}

### Messages:

All the messages will be sent to the kafka topic “Cloud-Platform-FTC”

|  |  |  |
| --- | --- | --- |
| MessageType | Message Format | Comments |
| vm.create.start | {"type":"vm.create.start","timeStamp":"2024-05-12 12:12:12 333","proxmoxId":534,"vmid":"2a04db360c94434485813ba609a9e2bc"} |  |
| vm.create.end | {"type":"vm.create.end","timeStamp":"2024-05-12 12:15:32 644","proxmoxId":534,"vmid":"2a04db360c94434485813ba609a9e2bc", "taskStatus":”success”} |  |
| vm.create.end | {"type":"vm.create.end","timeStamp":"2024-05-12 12:15:32 644","proxmoxId":534,"vmid":"2a04db360c94434485813ba609a9e2bc", "taskStatus":”failed”, "failedReason":”XXX”} | Only send final status in case of retry |

|  |  |  |
| --- | --- | --- |
| Error code | Error Message | Comments |
| 1002 | Sent duplicated request |  |
| 2001 | Input contains invalid characters |  |
| 2002 | Input cannot end with a hyphen or period |  |
| 2003 | Hyphens and periods cannot be adjacent |  |
| 2004 | FortiCSP password must be at least 8 characters long |  |
| 2005 | Windows password must contain characters from three of the following four categories:  1. Uppercase letters (A-Z)  2. Lowercase letters (a-z)  3. Digits (0-9)  4. Non-alphanumeric characters (e.g., !, $, #, %) |  |
| 2009 | The image for specified version of FortiCSP/FortiCSPManager/FortiGate doesn't exist |  |

## 2 Get the detail for a virtual machine

### Brief Description

Once a virtual machine is created, you could get its detail any time.

If the virtual machine is under preparation, you could get its current stage and status;

If the virtual machine is ready, you could get its details configuration such as disk list and ip/mac list, and also the status.

### HTTP Method:

Get

### Endpoint:

/vm/{server\_id}

### Parameters:

#### Header parameter:

Content-Type: application/json

#### Body parameter:

None

Examples:

### Response Body:

**{**

**"code": 200,**

**"message": "ok",**

**"data": {**

**"vmId": "3dd756ee569946b5aa374eb836f2b0d9",**

**"name": "KL-VM-Linux-3",**

**"templateId": 9002,**

**"proxmoxId": 111,**

**"vmState": 70,**

**"osFamily": "linux",**

**"osVersion": "ubuntu",**

**"cpu": 4,**

**"ram": 8192,**

**"disks": [**

**{**

**"index": 0,**

**"size": 5555**

**},**

**{**

**"index": 1,**

**"size": 6666**

**}**

**],**

**"nics": [**

**{**

**"index": 0,**

**"ip": "10.65.185.158",**

**"mac": "BC:24:11:FD:18:98",**

**"maskBitCount": 22,**

**"netId": 1,**

**"netName": "mgmt",**

**"vlan": 1,**

**"gateway": "10.65.187.254",**

**"dnsServer": ""**

**}**

**],**

**"metadata": {**

**"user": "gkl",**

**"password": "Forti1@#",**

**"sshKey": "keys",**

**"tags": {**

**"env1": "testbed1",**

**"env2": "testbed2"**

**}**

**},**

**"steps": [**

**{**

**"id": 38,**

**"taskId": "5b18c692bd844d7bbbee471f1a9fa226",**

**"stepNum": 0,**

**"stepStatus": 3,**

**"stepName": "VM\_CLONE",**

**"continueOnErr": false**

**},**

**{**

**"id": 39,**

**"taskId": "5b18c692bd844d7bbbee471f1a9fa226",**

**"stepNum": 1,**

**"stepStatus": 3,**

**"stepName": "VM\_BASIC\_CONFIG",**

**"continueOnErr": false**

**},**

**{**

**"id": 40,**

**"taskId": "5b18c692bd844d7bbbee471f1a9fa226",**

**"stepNum": 2,**

**"stepStatus": 3,**

**"stepName": "VM\_NETWORK\_CONFIG",**

**"continueOnErr": false**

**},**

**{**

**"id": 41,**

**"taskId": "5b18c692bd844d7bbbee471f1a9fa226",**

**"stepNum": 3,**

**"stepStatus": 3,**

**"stepName": "VM\_BOOT",**

**"continueOnErr": false**

**},**

**{**

**"id": 42,**

**"taskId": "5b18c692bd844d7bbbee471f1a9fa226",**

**"stepNum": 4,**

**"stepStatus": 3,**

**"stepName": "VM\_BOOT\_CHECK",**

**"continueOnErr": false**

**},**

**{**

**"id": 43,**

**"taskId": "5b18c692bd844d7bbbee471f1a9fa226",**

**"stepNum": 5,**

**"stepStatus": 3,**

**"stepName": "VM\_DELIVERY",**

**"continueOnErr": false**

**}**

**],**

**"allowAccess": [**

**{**

**"method": "SSH",**

**"ip": "10.65.187.177",**

**"port": 22,**

**"user": "gkl",**

**"password": "Forti"**

**}**

**],**

**"CreatedAt": "2024-05-06T15:53:32.609-07:00",**

**"UpdatedAt": "2024-05-06T15:54:16.505-07:00",**

**"DeletedAt": null**

**}**

**}**

|  |  |  |
| --- | --- | --- |
| VM status code | Status Name | Comments |
| 10 | Create Submitted |  |
| 20 | Creating |  |
| 30 | Create Failed |  |
| 40 | Create Completed |  |
| 50 | Delete Submitted |  |
| 60 | Deleting |  |
| 70 | Delete Completed |  |
| 80 | Delete Failed |  |
| 90 | Unknown |  |

## 3 Destroy the VM by a VM ID

### Brief Description

Destroy the VM and all its used/owned volumes and Nics and IPs.

### HTTP Method:

DELETE

### Endpoint:

/vm/{id}

### Parameters:

id: (required) The id of VM

### Response Body:

{

    "code": 200,

    "message": "success",

    "data": {

       "vmid": "e9ef557798d84f0faae0663b7f1ba96e",

    }

}

|  |  |  |
| --- | --- | --- |
| MessageType | Message Format | Comments |
| vm.destroy.start | {"type":"vm.destroy.start","timeStamp":"2024-05-12 12:12:12 333","proxmoxId":534,"vmid":"2a04db360c94434485813ba609a9e2bc"} |  |
| vm.destroy.end | {"type":"vm.destroy.end","timeStamp":"2024-05-12 12:15:32 644","proxmoxId":534,"vmid":"2a04db360c94434485813ba609a9e2bc", "taskStatus":”success”} |  |
| vm.destroy.end | {"type":"vm.destroy.end","timeStamp":"2024-05-12 12:15:32 644","proxmoxId":534,"vmid":"2a04db360c94434485813ba609a9e2bc", "taskStatus":”failed”, "failedReason":”XXX”} |  |

### Exception:

|  |  |  |
| --- | --- | --- |
| Error code | Error Message | Comments |
| 404 | Not Found |  |
| 1004 | Duplicate deletion request for the same VM. |  |
| 1005 | The VM has already been deleted |  |

Example:

{

    "code": 404,

    "message": "Not Found",

    "data": {}

}

## 4 Shutdown the VM by a VM ID

### Brief Description

This is similar to pressing the power button on a physical machine.  
This will send an ACPI event for the guest OS, which should then proceed to a clean shutdown.

### HTTP Method:

PUT

### Endpoint:

/vm/{id}/shutdown

### Parameters:

id: (required) The id of VM

### Response Body:

{

    "code": 200,

    "message": "success",

    "data": {

       "vmid": "e9ef557798d84f0faae0663b7f1ba96e"

    }

}

|  |  |  |
| --- | --- | --- |
| MessageType | Message Format | Comments |
| vm.shutdown.start | {"type":"vm.shutdown.start","timeStamp":"2024-05-12 12:12:12 333","proxmoxId":534, "vmid":"2a04db360c94434485813ba609a9e2bc"} |  |
| vm.shutdown.end | {"type":"vm.shutdown.end","timeStamp":"2024-05-12 12:15:32 644","proxmoxId":534, "vmid":"2a04db360c94434485813ba609a9e2bc","taskStatus":”success”} |  |
| vm.shutdown.end | {"type":"vm.shutdown.end","timeStamp":"2024-05-12 12:15:32 644","proxmoxId":534, "vmid":"2a04db360c94434485813ba609a9e2bc","taskStatus":”failed”,"failedReason":”XXX”} |  |

## 5 Start the VM by a VM ID

### Brief Description

This is similar to pressing on the power button on a physical machine..

### HTTP Method:

PUT

### Endpoint:

/vm/{id}/start

### Parameters:

id: (required) The id of VM

### Response Body:

{

    "code": 200,

    "message": "success",

    "data": {

       "vmid": "e9ef557798d84f0faae0663b7f1ba96e"

    }

}

|  |  |  |
| --- | --- | --- |
| MessageType | Message Format | Comments |
| vm.boot.start | {"type":"vm.boot.start","timeStamp":"2024-05-12 12:12:12 333","proxmoxId":534,"vmid":"2a04db360c94434485813ba609a9e2bc"} |  |
| vm.boot.end | {"type":"vm.boot.end","timeStamp":"2024-05-12 12:15:32 644","proxmoxId":534,"vmid":"2a04db360c94434485813ba609a9e2bc", "taskStatus":”success”} |  |
| vm.boot.end | {"type":"vm.boot.end","timeStamp":"2024-05-12 12:15:32 644","proxmoxId":534,"vmid":"2a04db360c94434485813ba609a9e2bc",  "taskStatus":”failed”,"failedReason":”XXX”} |  |

## 6 Get the available network list

### Brief Description

The available network is the network with no VM or device attached and the status is enabled.

You could specify the expected count of network returned with the query string ‘topN’;

If topN is not set, it will return all the available networks.

If ctrlNet is set true, it will return only the management network; otherwise it will only return the data network; All the available networks will be return if ctrlNet is not set.

### HTTP Method:

Get

### Endpoint:

/network/available?topN=2**2&ctrlNet=true**

### Parameters:

#### Header parameter:

Content-Type: application/json

#### Body parameter:

None

Examples:

### Response Body:

{**{**

**"code": 200,**

**"message": "ok",**

**"data": [**

**{**

**"name": "mgmt",**

**"netSegment": "10.65.184.0",**

**"mask\_bit\_count": 22,**

**"vlan": 1,**

**"gateway": "10.65.187.254",**

**"dns\_serverServer": "",**

**"dhcp\_enabled": true,**

**"dhcp\_range": "10.65.184.1-10.65.185.255",**

**"dhcp\_lease\_time": 604800,**

**"enabled": true,**

**"id": 1,**

**"CreatedAt": "0001-01-01T00:00:00Z",**

**"UpdatedAt": "0001-01-01T00:00:00Z",**

**"DeletedAt": null**

**},**

**{**

**"name": "186",**

**"netSegment": "10.186.0.0",**

**"mask\_bit\_count": 24,**

**"vlan": 186,**

**"gateway": "10.186.0.1",**

**"dns\_serverServer": "",**

**"dhcp\_enabled": true,**

**"dhcp\_range": "10.186.0.2-10.186.0.250",**

**"dhcp\_lease\_time": 604800,**

**"enabled": true,**

**"id": 2,**

**"CreatedAt": "0001-01-01T00:00:00Z",**

**"UpdatedAt": "0001-01-01T00:00:00Z",**

**"DeletedAt": null**

**},**

**{**

**"name": "188",**

**"netSegment": "10.188.0.0",**

**"mask\_bit\_count": 24,**

**"vlan": 188,**

**"gateway": "10.188.0.254",**

**"dns\_serverServer": "",**

**"dhcp\_enabled": true,**

**"dhcp\_range": "10.188.0.2-10.188.0.200",**

**"dhcp\_lease\_time": 604800,**

**"enabled": true,**

**"id": 3,**

**"CreatedAt": "0001-01-01T00:00:00Z",**

**"UpdatedAt": "0001-01-01T00:00:00Z",**

**"DeletedAt": null**

**}**

**]**

**}**

## 7 Get the available image list

### Brief Description

The available image is the image which the status is enabled.

### HTTP Method:

Get

### Endpoint:

/images

### Parameters:

#### Header parameter:

Content-Type: application/json

#### Body parameter:

None

Examples:

### Response Body:

**{**

**"code": 200,**

**"message": "ok",**

**"data": [**

**{**

**"id": 1,**

**"templateId": 9102,**

**"templateName": "windows-10-tpl",**

**"osName": "windows",**

**"osVersion": "windows-10",**

**"osFamily": "windows",**

**"enabled": true,**

**"createdAt": "0001-01-01T00:00:00Z",**

**"updatedAt": "0001-01-01T00:00:00Z",**

**"deletedAt": null**

**},**

**{**

**"id": 2,**

**"templateId": 9001,**

**"templateName": "ubuntu-bionic-amd64",**

**"osName": "ubuntu",**

**"osVersion": "ubuntu 2023",**

**"osFamily": "linux",**

**"enabled": true,**

**"createdAt": "0001-01-01T00:00:00Z",**

**"updatedAt": "0001-01-01T00:00:00Z",**

**"deletedAt": null**

**},**

**{**

**"id": 3,**

**"templateId": 9007,**

**"templateName": "FortiCSP-Template",**

**"osName": "FortiCSP",**

**"osVersion": "1.0.0\_b0025",**

**"osFamily": "forticsp",**

**"enabled": true,**

**"createdAt": "0001-01-01T00:00:00Z",**

**"updatedAt": "0001-01-01T00:00:00Z",**

**"deletedAt": null**

**},**

**{**

**"id": 4,**

**"templateId": 9201,**

**"templateName": "FGT-7.4.3-tpl",**

**"osName": "FGT",**

**"osVersion": "7.4.3",**

**"osFamily": "fortigate",**

**"enabled": true,**

**"createdAt": "0001-01-01T00:00:00Z",**

**"updatedAt": "0001-01-01T00:00:00Z",**

**"deletedAt": null**

**}**

**]**

**}**