

Instances(WC)

Introduction

For information on how to access the API and API authorization see: [Obtaining API Keys](#).

The instance API is a low level interface designed to allocate, monitor, control and de-allocate instances of cloud resources.

All resources have a description as defined in [Cloud Resources](#). The instance object is an instantiation of one specific type of resource.

In the instance API, everything is considered an “instance” (not just a VM instance). We can have an instance of a key pair, an instance of a security group, an instance of a volume store, etc.

All **resources** are identified by means of a URI (Universal Resource Identifier) which uniquely identifies the resource in the ComputeNext catalog.

This URI has the form <resourceType>/<provider>/<region>/<providerResourceId>. For example: vm/hpcloud/nova/small

NOTE: By convention the URI is simply a string the ComputeNext catalog uses to uniquely identify a specific type of resource from a provider at a particular region.

The URI is always lower case to avoid case sensitivity issues.

All **instances** are identified by an **instanceId** which is a Guid.

The instance API is asynchronous. Most requests return a **requestId** which is a Guid. They will return the requestId when the request has been accepted and is in-progress, but has not completed. The user must poll with that requestId to find out the status of the request. Eventually the request will become completed or will have failed.

Possible resource types for instantiable instances (see list of resourceType in section Resources of [Cloud Resources](#)) are -

- kp - key pair
- sg - security group
- vm - virtual machine
- vs - volume storage
- image - image
- snap - volume snapshot
- ip - floating (elastic) IP address
- lb - loadBalancer for distributing compute load across various running instances using selectable algorithms

There are basically two types of records that are persisted by the instance API in the database - request and instance records.

requests. These are records which keep track of the user requests the API. Each request has a **requestId** (a Guid) and a **requestStatus**, plus all the parameters and results etc.

requestStatus can be -

- in-progress
- completed
- failed

If failed there will be some error information indicating the cause of the failure. Error information will include an error code, an error message, and an error ticket (a Guid) which we can use to track down the exact source of the error from the trace files.

instances: These are the records which keep track of the instances allocated at the provider side. Each instance has an **instanceId** and an **instanceStatus**. The instanceStatus is the last status we were able to retrieve from the provider side. Values for instanceStatus are similar to (creating, created, stating, deleted. etc.) and are all lower case.

metadata. The instance API uses a URI or Guids to identify everything. Names are not used. However, every instance can have metadata (tags) added. This can be used to add names, descriptions, etc. and it is also used by the workload API to tag specific instances as being part of a particular workload or transaction by adding a workloadId or a transactionId to the instance metadata.

(**Note:** if you want to add new metadata attributes to a VM instance then you must read the existing metadata, add your new attributes to it, and then update it.)

All request parameters and responses in the instance API are in JSON format.

Actions

In the instance API, the actions that can be performed vary by the resource type.

Actions are identified in the following format: <resourceType>.<CRUD action>.<action modifier>

- <resourceType> identifies the resource type (such as “vm”, “kp”, “sg” etc. as listed above)
- <CRUD action> is one of the create, retrieve, update or delete (CRUD) actions
- <action modifier> is an additional modifier for the CRUD action (such as “start”, “stop”, “reboot” etc.)

Actions names are lower case to avoid case sensitivity issues. Some examples are “vm.create”, “vs.update.attach”, “kp.delete”.

The <resourceType> is determined from the resource type of the resource or instance being acted upon - you do not have to specify this explicitly.

The CRUD action is determined from the type of REST action being performed -

- POST = create
- GET = retrieve
- PUT = update
- DELETE = delete

The <action modifier> is not required in many cases, but if it is required, it is specified as the *action* query parameter on the REST request URL “?action=start” for example.

A full list of the resource types, the actions they support, and the parameters for those actions can be found in [Resource Types, Actions and Parameters. Getting Started with Instances](#) gives samples of actions on instances using runcws commands with json file parameters with actions contained in them.

Instance Methods

Note that Basic authentication is required but is not shown here. See [Authentication](#) for details.

Create instance from resource

Create one instance from a resource uri and parameters. The parameters required will vary depending on the resource.

POST /api/resource/<uri>

Request Body

- metadata = some JSON to be added to the instance as metadata (tags)
- <parameters> = will vary depending on the resource type.

Example

POST /api/resource/kp/hpcloud/nova/standard (no parameters)

Body

```
{
  "metadata": {
    "name": "HP_KP",
    "description": "Keypair Response time"
  }
}
```

This will create a key pair in the HPCloud Nova region.

Returns

A request JSON object.

Retrieve request

Retrieve one request.

GET /api/request/<requestId>

Example

GET /api/request/aec5fb2d-9990-4cdf-bc24-6fbc1a13cbf4

Response

```
[
  {
    "requestId": "3780520d-9463-4273-8d52-aa492b185168",
    "instanceId": "b2b7c744-fe31-4db4-aab2-d56c118ae9ee",
    "created": "2013-07-03T07:45:56.001Z",
    "updated": "2013-07-03T07:46:00.674Z",
    "ownerId": "ael6300f-6eba-426f-b92b-dd5626f094bf",
    "requestStatus": "completed",
    "resourceUri": "kp/hpcloud/nova/standard",
    "resourceType": "kp",
    "provider": "hpcloud",
    "region": "nova",
    "connector": "openStack.compute",
    "parameters": {
      {
        "action": "kp.create",
        "instanceId": "b2b7c744-fe31-4db4-aab2-d56c118ae9ee",
        "kp_providerResourceId": "standard",
        "zone": "nova"
      },
      "metadata": {
        {
          "name": "KP2",
          "description": "my first key pair"
        },
        "results": {
          {
            "providerInstanceId": "b2b7c744-fe31-4db4-aab2-d56c118ae9ee",
            "keyFingerprint": "6a:c4:4f:ba:02:0b:d4:73:93:b7:08:23:d7:d2:17:40",
            "privateKey": "<HIDDEN>",
            "publicKey": "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgQDBvzE7rWC9G7+bpXwwluG08im... \n",
            "instanceStatus": "created"
          }
        }
      }
    }
  }
]
```

Returns

A request JSON object.

Retrieve multiple requests

Retrieve multiple requests. Query parameters can be provided to return just those requests that match.

GET /api/request

Query Parameters

- requestStatus = in-progress | completed | failed
- resourceType = kp | sg | vm | vs | image | snap | ip | lb
- provider = <provider name, lower case>
- region = <region name, lower case>
- cleanup = true - if this is set, then any old requests for this user that have "requestStatus = completed" or "requestStatus = failed" will be cleaned up.

If multiple query parameters are provided then you will get the logical AND effect from these parameters.

Example

GET /api/request?resourceType=vm

Returns

A JSON array of request objects.

Create an instance from an instance

Create one instance from another instance. At the moment this is only used for creating a new vm image instance from an existing vm instance.

POST /api/instance/<instanceId>

Example

Body

```
{
  "virtualMachineId": "456afec0-2c63-45d0-8436-df1815aabb5c",
  "metadata": {
    "name": "New IMAGE",
    "description": "my first image"
  }
}
```

Query Parameters

TODO:

Returns

A request JSON object.

Note: Section [Getting Started with Instances](#) has an example of creating an image from a vm instance using runcws in sub-section *Create a Private image from a Virtual Machine*.

Retrieve instance

Retrieve one instance.

GET /api/instance/<instanceId>

Query Parameters

- refresh = true - refresh the instance from the provider side.

If this is *not* set, then the instance is fetched from the database only, and no request is made to the provider side. An instance is returned.

If this *is* set, then a request is returned and a request is sent to the provider side to get the latest instance status.

Returns

If refresh is specified, a request JSON object is returned.

If refresh is not specified, an instance JSON object is returned.

Note: Section [Getting Started with Instances](#) has several examples of getting an instance using runcws. See sub-section *Create a Private image from a Virtual Machine*.

Retrieve multiple instances

Retrieve multiple instances. Query parameters can be provided to return just those instances that match.

GET /api/instance

Query Parameters

- instanceStatus = creating | created | starting | started... etc.
- resourceType = kp | sg | vm | vs | image | snap | ip | lb
- provider = <provider name, lower case>
- region = <region name, lower case>
- metadata.<parameter> = query by some metadata parameter
- workloadId = workload id (Guid)
- transactionId = transaction id (Guid)

Example

GET /api/instance?workloadId=ad5a66c3-8895-4d71-b786-1d129b33326e

Response

```
[
  {
    "instanceId": "de3ef325-7daa-4d47-ae8-4d0391b18d9b",
    "created": "2013-10-05T12:46:12.428Z",
    "updated": "2013-10-05T12:46:39.252Z",
    "ownerId": "0cfc0576-0088-486a-8416-7dbd79f2776e",
```

```

"resourceUri": "vm/hpcloud/nova/standard.small",
"resourceType": "vm",
"provider": "hpcloud",
"region": "nova",
"providerResourceId": "Standard.small",
"attributes":
{
  "providerInstanceId": 2205937,
  "password": "KAcqKeQX9XNagL5a",
  "instanceStatus": "running",
  "privateIpAddress": "10.2.225.213",
  "publicIpAddress": "15.185.233.238"
},
"attributeTimestamps":
{
  "password": "2013-10-05T12:46:39.244Z",
  "instanceStatus": "2013-10-05T12:46:39.245Z",
  "privateIpAddress": "2013-10-05T12:46:39.244Z",
  "publicIpAddress": "2013-10-05T12:46:39.244Z"
},
"metadata":
{
  "name": "Ubuntu",
  "description": "Virtual machine",
  "workloadId": "333617b7-5ade-43b4-88eb-45703ec1b0d4",
  "transactionId": "ab827e43-be23-4a33-8e01-11925cbe64d4"
},
"parameters":
{
  "imageUri": "image/hpcloud/nova/ami-00000075",
  "keyPairId": "15086ab9-bbe6-45d5-8d68-377bd4af58c4",
  "securityGroupIds":
  [
    "639a42d8-9d76-4422-919b-bcc49db0524c"
  ],
  "vm_providerResourceId": "Standard.small",
  "zone": "nova",
  "username": "ubuntu",
  "image_providerResourceId": "ami-00000075",
  "keyPairId_providerInstanceId": "15086ab9-bbe6-45d5-8d68-377bd4af58c4",
  "securityGroupIds_providerInstanceId": [398345]
}
},
{
  "instanceId": "888bd60a-e8e0-46d2-9fb9-60d0596d837a",
  "created": "2013-10-05T12:47:09.376Z",
  "updated": "2013-10-05T12:47:16.946Z",
  "ownerId": "0cfc0576-0088-486a-8416-7dbd79f2776e",
  "resourceUri": "vs/hpcloud/nova/standard.10",
  "resourceType": "vs",
  "provider": "hpcloud",
  "region": "nova",
  "providerResourceId": "standard",
  "attributes":
  {
    "providerInstanceId": 675435,
    "instanceStatus": "attached",
    "virtualMachineId": "de3ef325-7daa-4d47-ae8-4d0391b18d9b",
    "virtualMachineId_providerInstanceId": 2205937
  },
  "attributeTimestamps":
  {
    "instanceStatus": "2013-10-05T12:47:16.936Z",
    "virtualMachineId": "2013-10-05T12:47:16.936Z",
    "virtualMachineId_providerInstanceId": "2013-10-05T12:47:16.937Z"
  },
  "metadata":
  {
    "description": "My first volume storage",
    "name": "vsl",
    "workloadId": "333617b7-5ade-43b4-88eb-45703ec1b0d4",
    "transactionId": "ab827e43-be23-4a33-8e01-11925cbe64d4"
  }
}

```

```

    },
    "parameters":
    {
        "sizeInGB": 50,
        "attachTarget": "Ubuntu",
        "deviceName": "/dev/vdc",
        "virtualMachineId": "de3ef325-7daa-4d47-ae8-4d0391b18d9b",
        "vs_providerResourceId": "standard",
        "zone": "nova",
        "virtualMachineId_providerInstanceId": 2205937
    }
}
]

```

Returns

A JSON array of instance objects.

Note: Section [Getting Started with Instances](#) has an example of listing multiple instances using `runcws` in the sub-section *Getting Started With Instances*.

Update instance

PUT /api/instance/<instanceId>

Request Body

- metadata = some JSON to be added to the instance as metadata (tags)
- <parameters> = will vary depending on the resource type and the action.

Query Parameters

action = <some action>. Valid actions depend on the resource type.

If no action parameter is provided then default actions are as follows -

- POST = create
- PUT = update
- GET = retrieve
- DELETE = delete

Example

PUT /api/instance/ad5a66c3-8895-4d71-b786-1d129b33326e?action=stop

Response

```
[
  {
    "requestId": "27aac883-5444-4db0-acf6-795f3a9aec00",
    "instanceId": "456afec0-2c63-45d0-8436-df1815aabb5c",
    "created": "2013-10-07T09:30:13.854Z",
    "updated": "2013-10-07T09:30:14.588Z",
    "ownerId": "0cfc0576-0088-486a-8416-7dbd79f2776e",
    "requestStatus": "completed",
    "resourceUri": "vm/lunacloud/eu-west/xsmall",
    "resourceType": "vm",
    "provider": "lunacloud",
    "region": "eu-west",
    "connector": "lunaCloud.compute",
    "parameters":
    {
      "uri": "vm/lunacloud/eu-west/xsmall",
      "imageUri": "image/lunacloud/eu-west/ubuntu-12.04-x86_64",
      "providerInstanceId": "456afec0-2c63-45d0-8436-df1815aabb5c",
      "action": "vm.update.stop",
      "instanceId": "456afec0-2c63-45d0-8436-df1815aabb5c",
      "vm_providerResourceId": "{ \"cpuCount\": \"1\", \"cpupower\": \"1500\" ... }",
      "zone": "EU-West"
    },
    "metadata":
    {
      "name": "VM10",
      "description": "my first virtual machine"
    },
    "results":
    {
      "providerInstanceId": "456afec0-2c63-45d0-8436-df1815aabb5c",
      "instanceStatus": "stopping"
    }
  }
]
```

Returns

A request JSON object.

Update instance metadata

Updates the metadata field of an instance. Users will pass in a piece of JSON to update the instance's metadata with.

PUT /api/instance/<instanceId>

Request Body

- metadata = a JSON file to be added to the instance as metadata (tags)

Metadata Parameters

Common parameters of a instance in an active workload:

- name = The unique name that the system uses to identify a VM, should not be modified
- description = This attribute is used for VM name, as it appears in your workload. Can be updated by user.
- monitoringEnabled = Tells you if the ComputeNext monitoring service has been installed and enabled on the VM
- workloadId = The Id of an existing workload that the VM is attached to.
- transactionId = Do not modify this, include it in any json code you are trying to pass through.

Users can define their one attribute in the form of "newattribute" : "testnewattribute" but must pass all previous attributes to retain them.

Example

PUT /api/instance/ad5a66c3-8895-4d71-b786-1d129b33326e

Content

```
{
  "metadata": {
    "name": "CNWE734D_3504_E6F8_FA5A_41AA96E36E08",
    "description": "My new name",
    "monitoringEnabled": false,
    "workloadId": "b6d5ed5a-29e7-4455-a95c-296072c2be1d",
    "transactionId": "d40fe39c-855f-49a1-8341-5c0952b179c9",
    "newattribute": "testnewattribute"
  }
}
```

Returns

A request JSON object.

Delete instance

Delete one instance. Note that “deleting an instance” means deleting (de-provisioning) the instance from the provider side. The instance record is not actually deleted from our database. The instance will transition through a “deleting” state and hopefully into a “deleted” state if all goes well.

DELETE /api/instance/<instanceId>

Example

DELETE /api/instance/ad5a66c3-8895-4d71-b786-1d129b33326e

Returns

A request JSON object.