



# Jio Cloud API Reference

18<sup>th</sup> January, 2014

## Contents

Block Storage API .....	4
Volumes.....	6
Volume Types.....	6
Snapshots .....	6
Compute API.....	7
Versions .....	7
Extensions .....	8
Limits .....	8
Servers .....	8
Server Metadata .....	8
Server Addresses.....	9
Server Actions .....	9
Flavors.....	9
Images.....	10
Image Metadata.....	11
Server Admin Actions.....	11
Server Console Output.....	12
Server Console.....	12
Server Deferred Delete.....	12
Server Diagnostics.....	12
Flavors with FlavorDisabled Attribute.....	13
Flavor extra-specs.....	13
Flavors with rxtx_factor extended attribute.....	13
Flavors with extended attributes .....	13
Images with size attribute .....	13
Limits with project usage .....	14
Guest agents.....	14
Host aggregates.....	14
Attach interfaces .....	15
Root certificates.....	16
Cloudpipe.....	16
Coverage reports .....	16
Fixed IPs.....	16
Floating IP DNS Records.....	16
Floating IP pools.....	17

Floating IPs.....	17
Floating IPs bulk .....	17
Hosts.....	18
Server Actions.....	18
Keypairs .....	18
Networks .....	19
Server rescue and unrescue.....	19
Server start and stop .....	19
Security Groups.....	20
Rules for default security group .....	20
Server Password.....	20
Manage services .....	21
Usage Reports .....	21
Virtual interfaces.....	21
Volume extensions.....	21
Volume attachments.....	22
Volume attachments.....	23
Servers with Block device mapping format.....	23
Server OS-EXT-IPS-MAC:mac_addr extended attribute .....	23
Configuration drive.....	24
Servers and images with disk config.....	24
Server IP type.....	24
Server extended attributes .....	24
Server extended status .....	26
Servers multiple create.....	26
Image Service API.....	26
Images .....	26
Image schemas.....	27
Object Storage API .....	28

# API Documentation

---

## Block Storage API

Jio Cloud Block Storage provides volume management with the Jio Cloud Compute service.

This section describes the features available with the Block Storage API.

This section assists software developers who develop applications by using the Block Storage API. It assumes the reader has a general understanding of storage and is familiar with:

- ReSTful web services
- HTTP/1.1 conventions
- JSON and/or XML data serialization formats

Jio Cloud Block Storage Service is a block-level storage solution that enables you to:

- Mount drives to Jio Cloud servers to scale storage without paying for more compute resources.
- Use high performance storage to serve database or I/O-intensive applications.

You interact with Block Storage programmatically through the Block Storage API as described in this section.

- Block Storage allows you to create snapshots that you can save, list, and restore.
- Block Storage allows you to create backups of your volumes to Object Storage for archival and disaster recovery purposes. These backups can be subsequently restored to the same volume or new volumes.

To use the Block Storage API effectively, you must understand several key concepts:

- **Volume** - A detachable block storage device. You can think of it as a USB hard drive. It can only be attached to one instance at a time.
- **Volume type** - A type of a block storage volume. You can define whatever types work best for you, such as SATA, SCSCI, SSD, etc. These can be customized. You can also define extra\_specs associated with your volume types. For instance, you could have a VolumeType=SATA, with extra\_specs (RPM=10000, RAID-Level=5).
- **Snapshot** - A point in time copy of the data contained in a volume.
- **Instance** - A virtual machine (VM) that runs inside the cloud.
- **Backup** - A full copy of a volume stored in an external service. The service can be configured. The only supported service for now is Object Storage. A backup can subsequently be restored from the external service to either the same volume that the backup was originally taken from, or to a new volume.

The Block Storage API is implemented using a ReSTful web service interface. It shares a common token-based authentication system that allows access between products and services.

All requests to authenticate against and operate the service are performed using SSL over HTTP (HTTPS) on TCP port 443.

The Block Storage API supports both the JSON and XML data serialization formats. The request format is specified using the Content-Type header and is required for calls that have a request body. The response format can be specified in requests either by using the Accept header or by adding an .xml or .json extension to the request URI. Note that it is possible for a response to be serialized using a format different from the request. If no response format is specified, JSON is the default. If conflicting formats are specified using both an Accept header and a query extension, the query extension takes precedence.

All accounts, by default, have a preconfigured set of thresholds (or limits) to manage capacity and prevent abuse of the system. The system recognizes two kinds of limits: rate limits and absolute limits. Rate limits are thresholds that are reset after a certain amount of time passes. Absolute limits are fixed.

When an error occurs, the Block Storage Service returns a fault object containing an HTTP error response code that denotes the type of error. The response body returns additional information about the fault.

The following table lists possible fault types with their associated error codes and descriptions.

Fault Type	Associated Error Code	Description
badRequest	400	The user request contains one or more errors
Unauthorized	401	The supplied token is not authorized to access the resources, either its expired or invalid
Forbidden	403	Access to the requested resource was denied
itemNotFound	404	The back-end services did not find anything matching the Request URI
badMethod	405	The request method is not allowed for this resource
overLimit	413	Either the number of entities in the request is larger than allowed limits, or the user has exceeded allowable request rate limits
badMediaType	415	The requested content type is not supported by this service
unprocessableEntity	422	The requested resource could not be processed on at the moment
instanceFault	500	This is a generic server error and the message contains the reason for the error. This error could wrap several error messages and is a catch all.
notImplemented	501	The requested method or resource is not

		implemented
serviceUnavailable	503	The Block Storage Service is not available

## Volumes

Method	URI	Description
POST	v2/{tenant_id}/volumes	Creates a volume
GET	v2/{tenant_id}/volumes	Lists summary information for all Block Storage volumes that the tenant who submits the request can access
GET	v2/{tenant_id}/volumes/detail	Lists detailed information for all Block Storage volumes that the tenant who submits the request can access
GET	v2/{tenant_id}/volumes/{volume_id}	Shows information about a specified volume
PUT	v2/{tenant_id}/volumes/{volume_id}	Updates a volume
DELETE	v2/{tenant_id}/volumes/{volume_id}	Deletes a specified volume

## Volume Types

Method	URI	Description
GET	v2/{tenant_id}/types	Lists volume types
GET	v2/{tenant_id}/types/{volume_type_id}	Shows information about a specified volume type

## Snapshots

Method	URI	Description
POST	v2/{tenant_id}/snapshots	Creates a snapshot, which is a point-in-time copy of a volume. You can create a volume from the snapshot
GET	v2/{tenant_id}/snapshots	Lists summary information for all Block Storage snapshots that the tenant who submits the request can access
GET	v2/{tenant_id}/snapshots/detail	Lists detailed information for all Block Storage snapshots that the tenant who submits the request can access

GET	v2/{tenant_id}/snapshots/{snapshot_id}	Shows information for a specified snapshot
PUT	v2/{tenant_id}/snapshots/{snapshot_id}	Updates a specified snapshot
DELETE	v2/{tenant_id}/snapshots/{snapshot_id}	Deletes a specified snapshot

## Compute API

Jio Cloud Compute is a compute service that provides server capacity in the cloud. Compute Servers come in different flavors of memory, cores, disk space, and CPU, and can be provisioned in minutes. Interactions with Compute Servers can occur programmatically via the Jio Cloud Compute API.

The Jio Cloud Compute API is defined as a ReSTful HTTP service. The API takes advantage of all aspects of the HTTP protocol (methods, URLs, media types, response codes, etc.)

To use the Jio Cloud Compute API effectively, you should understand several key concepts:

- **Server** - A virtual machine (VM) instance in the compute system. Flavor and image are requisite elements when creating a server.
- **Flavor** - An available hardware configuration for a server. Each flavor has a unique combination of disk space, memory capacity and priority for CPU time.
- **Image** - A collection of files used to create or rebuild a server. Operators provide a number of prebuilt OS images by default. You may also create custom images from cloud servers you have launched. These custom images are useful for backup purposes or for producing “gold” server images if you plan to deploy a particular server configuration frequently.
- **Reboot** - Use this function to perform either a soft or hard reboot of a server. With a soft reboot, the operating system is signaled to restart, which allows for a graceful shutdown of all processes. A hard reboot is the equivalent of power cycling the server. The virtualization platform should ensure that the reboot action has completed successfully even in cases in which the underlying domain/VM is paused or halted/stopped.
- **Rebuild** - Use this function to remove all data on the server and replaces it with the specified image. Server ID and IP addresses remain the same.
- **Resize** - Use this function to convert an existing server to a different flavor, in essence, scaling the server up or down. The original server is saved for a period of time to allow rollback if there is a problem. All resizes should be tested and explicitly confirmed, at which time the original server is removed. All resizes are automatically confirmed after 24 hours if you do not confirm or revert them.

## Versions

Method	URI	Description
--------	-----	-------------

GET	v2	Gets details about this specific version of the API
-----	----	---

## Extensions

Method	URI	Description
GET	v2/{tenant_id}/extensions	Lists available extensions
GET	v2/{tenant_id}/extensions/{alias}	Gets details about the specified extension

## Limits

Method	URI	Description
GET	v2/{tenant_id}/limits	Lists the current limits for the account

## Servers

Method	URI	Description
GET	v2/{tenant_id}/servers	Lists IDs, names, and links for all servers
POST	v2/{tenant_id}/servers	Creates a server
GET	v2/{tenant_id}/servers/detail	Lists details for all servers
GET	v2/{tenant_id}/servers/{server_id}	Gets details for a specified server
PUT	v2/{tenant_id}/servers/{server_id}	Updates the editable attributes of the specified server
DELETE	v2/{tenant_id}/servers/{server_id}	Deletes a specified server

## Server Metadata

Method	URI	Description
GET	v2/{tenant_id}/servers/{server_id}/metadata	Lists metadata for the specified resource
POST	v2/{tenant_id}/servers/{server_id}/metadata	Updates metadata items by key for the specified resource
PUT	v2/{tenant_id}/servers/{server_id}/metadata	Sets metadata for the specified resource
GET	v2/{tenant_id}/servers/{server_id}/metadata/{key}	Gets a metadata item by key for the specified resource
PUT	v2/{tenant_id}/servers/{server_id}/metadata/{key}	Sets a metadata item by key for the specified resource



	{server_id}/metadata/{key}	resource
DELETE	v2/{tenant_id}/servers/{server_id}/metadata/{key}	Deletes a metadata item by key for the specified resource

## Server Addresses

Method	URI	Description
GET	v2/{tenant_id}/servers/{server_id}/ips	Lists networks and addresses for a specified tenant and server
GET	v2/{tenant_id}/servers/{server_id}/ips/{network_label}	Lists addresses for a specified tenant, server, and network

## Server Actions

Method	URI	Description
POST	v2/{tenant_id}/servers/{server_id}/action	Changes the password for a server. Specify the <b>changePassword</b> action in the request body
POST	v2/{tenant_id}/servers/{server_id}/action	Reboots the specified server. Specify the <b>reboot</b> action in the request body
POST	v2/{tenant_id}/servers/{server_id}/action	Rebuilds the specified server. Specify the <b>rebuild</b> action in the request body
POST	v2/{tenant_id}/servers/{server_id}/action	Resizes the specified server. Specify the <b>resize</b> action in the request body
POST	v2/{tenant_id}/servers/{server_id}/action	Confirms a pending resize action. Specify the <b>confirmResize</b> action in the request body
POST	v2/{tenant_id}/servers/{server_id}/action	Cancels and reverts a pending resize action. Specify the <b>revertResize</b> action in the request body
POST	v2/{tenant_id}/servers/{server_id}/action	Creates a new image. Specify the <b>createImage</b> action in the request body

## Flavors

Method	URI	Description
GET	v2/flavors	Lists IDs, names, and links for available flavors
GET	v2/flavors/detail	Lists all details for available flavors
GET	v2/flavors/{flavor_id}	Gets details for a specified flavour

## Images

Method	URI	Description
GET	v2/images	Lists IDs, names, and links for available images
GET	v2/images/detail	Lists all details for available images
GET	v2/images/{image_id}	Gets details for a specified image
DELETE	v2/images/{image_id}	Deletes a specified image

## Image Metadata

Method	URI	Description
GET	v2/images/{image_id}/metadata	Lists metadata for the specified resource
POST	v2/images/{image_id}/metadata	Updates metadata items by key for the specified resource
PUT	v2/images/{image_id}/metadata	Sets metadata for the specified resource
GET	v2/images/{image_id}/metadata/{key}	Gets a metadata item by key for the specified resource
PUT	v2/images/{image_id}/metadata/{key}	Sets a metadata item by key for the specified resource
DELETE	v2/images/{image_id}/metadata/{key}	Deletes a metadata item by key for the specified resource

## Server Admin Actions

Method	URI	Description
POST	v2/{tenant_id}/servers/{server_id}/action	Pauses a server. Changes its status to PAUSED
POST	v2/{tenant_id}/servers/{server_id}/action	Unpauses a PAUSED server and changes its status to ACTIVE
POST	v2/{tenant_id}/servers/{server_id}/action	Suspends a server and changes its status to SUSPENDED
POST	v2/{tenant_id}/servers/{server_id}/action	Resumes a SUSPENDED server and changes its status to ACTIVE
POST	v2/{tenant_id}/servers/{server_id}/action	Migrates a server to a host. The scheduler chooses the host
POST	v2/{tenant_id}/servers/{server_id}/action	Resets networking on a server
POST	v2/{tenant_id}/servers/{server_id}/action	Injects network information into a server
POST	v2/{tenant_id}/servers/{server_id}/action	Locks a server
POST	v2/{tenant_id}/servers/{server_id}/action	Unlocks a server
POST	v2/{tenant_id}/servers/{server_id}/action	Backs up a server instance

POST	v2/{tenant_id}/servers/{server_id}/action	Live-migrates a server to a new host without rebooting
POST	v2/{tenant_id}/servers/{server_id}/action	Resets the state of a server to a specified state
POST	v2/{tenant_id}/servers/{server_id}/action	Evacuates a server from failed host
POST	v2/{tenant_id}/servers/{server_id}/action	Assigns the specified security group to the server
POST	v2/{tenant_id}/servers/{server_id}/action	Removes the specified security group from the server
POST	v2/{tenant_id}/servers/{server_id}/action	Adds a floating IP address to an instance

## Server Console Output

Method	URI	Description
POST	v2/{tenant_id}/servers/{server_id}/action	Gets console output for a server instance

## Server Console

Method	URI	Description
POST	v2/{tenant_id}/servers/{server_id}/action	Gets a console for a server instance

## Server Deferred Delete

Method	URI	Description
POST	v2/{tenant_id}/servers/{server_id}/action	Force-deletes a server
POST	v2/{tenant_id}/servers/{server_id}/action	Restores a deleted server

## Server Diagnostics

Method	URI	Description
GET	v2/{tenant_id}/servers/{server_id}/diagnostics	Gets basic usage data for a specified server

## Flavors with FlavorDisabled Attribute

Method	URI	Description
GET	v2/{tenant_id}/flavors/ {flavor_id}	Gets details for a specified flavor. Includes the OS-FLV-DISABLED:disabled attribute
GET	v2/{tenant_id}/flavors/detail	Lists available flavors. Includes the OS-FLV-DISABLED:disabled attribute

## Flavor extra-specs

Method	URI	Description
GET	v2/{tenant_id}/flavors/ {flavor_id}/os-extra_specs	Lists the extra-specs or keys for the specified flavour
POST	v2/{tenant_id}/flavors/ {flavor_id}/os-extra_specs	Creates extra-specs or keys for the specified flavour
GET	v2/{tenant_id}/flavors/ {flavor_id}/os-extra_specs/ {key_id}	Gets the value of the specified key

## Flavors with rxtx\_factor extended attribute

Method	URI	Description
GET	v2/{tenant_id}/flavors/ {flavor_id}	Gets details for a specified flavor. Includes the rxtx_factor extended attribute
GET	v2/{tenant_id}/flavors/detail	Lists details for available flavors and includes the rxtx_factor extended attribute

## Flavors with extended attributes

Method	URI	Description
GET	v2/{tenant_id}/flavors/ {flavor_id}	Gets details for a specified flavor. Includes the rxtx_factor, OS-FLV-EXT-DATA:ephemeral, and swap extended attributes
GET	v2/{tenant_id}/flavors/detail	Lists available flavors. Includes the rxtx_factor, OS-FLV-EXT-DATA:ephemeral, and swap extended attributes

## Images with size attribute

Method	URI	Description
--------	-----	-------------

GET	v2/{tenant_id}/images/detail	Lists details for available images. Includes the image size
GET	v2/{tenant_id}/images/{image_id}	Gets details for a specified image. Includes the image size

## Limits with project usage

Method	URI	Description
GET	v2/{tenant_id}/limits	Gets absolute and rate limit information, including information on currently used absolute limits

## Guest agents

Method	URI	Description
GET	v2/{tenant_id}/os-agents	Lists all agent builds
POST	v2/{tenant_id}/os-agents	Creates an agent build
DELETE	v2/{tenant_id}/os-agents	Deletes an existing agent build
PUT	v2/{tenant_id}/os-agents/{id}	Updates an agent build

## Host aggregates

Method	URI	Description
GET	v2/{tenant_id}/os-aggregates	Lists all aggregates
POST	v2/{tenant_id}/os-aggregates	Creates an aggregate
DELETE	v2/{tenant_id}/os-aggregates/{aggregate_id}	Deletes an aggregate
GET	v2/{tenant_id}/os-aggregates/{aggregate_id}	Gets details about a specified aggregate
PUT	v2/{tenant_id}/os-aggregates/{aggregate_id}	Updates the name, and optionally the availability zone, for a specified aggregate
POST	v2/{tenant_id}/os-aggregates/{aggregate_id}/action	Sets metadata for an aggregate
POST	v2/{tenant_id}/os-aggregates/{aggregate_id}/action	Adds a host to an aggregate
POST	v2/{tenant_id}/os-aggregates/{aggregate_id}/action	Removes a host from an aggregate

## Attach interfaces

Method	URI	Description
POST	v2/{tenant_id}/servers/{server_id}/os-attach-interfaces	Creates and uses a port interface to attach the port to a server instance
GET	v2/{tenant_id}/servers/{server_id}/os-attach-interfaces	Lists port interfaces
GET	v2/{tenant_id}/servers/{server_id}/os-attach-interfaces/{attachment_id}	Shows information about a specified port interface

## Root certificates

Method	URI	Description
POST	v2/{tenant_id}/os-certificates	Creates a root certificate
GET	v2/{tenant_id}/os-certificates	Shows details for a root certificate owned by a specified tenant ID

## Cloudpipe

Method	URI	Description
GET	v2/{tenant_id}/os-cloudpipe	Lists cloudpipes
POST	v2/{tenant_id}/os-cloudpipe	Creates a cloudpipe
POST	v2/{tenant_id}/os-cloudpipe/configure-project	Updates the virtual private network (VPN) IP address and port for a specified cloudpipe instance

## Coverage reports

Method	URI	Description
POST	v2/{tenant_id}/os-coverage/action	Generates a coverage report
POST	v2/{tenant_id}/os-coverage/action	Starts Nova coverage reporting
POST	v2/{tenant_id}/os-coverage/action	Starts coverage reporting for all services
POST	v2/{tenant_id}/os-coverage/action	Stops coverage reporting

## Fixed IPs

Method	URI	Description
GET	v2/{tenant_id}/os-fixed-ips/{fixed_ip}	Shows information for a specified fixed IP address
POST	v2/{tenant_id}/os-fixed-ips/{fixed_ip}/action	Reserves or releases a fixed IP

## Floating IP DNS Records

Method	URI	Description
GET	v2/{tenant_id}/os-floating-ip-dns	Lists registered DNS domains published by the DNS drivers



PUT	v2/{tenant_id}/os-floating-ip-dns/{domain}	Creates or updates a DNS domain
DELETE	v2/{tenant_id}/os-floating-ip-dns/{domain}	Deletes a DNS domain and all associated host entries
PUT	v2/{tenant_id}/os-floating-ip-dns/{domain}/entries/{name}	Creates or updates a DNS entry
GET	v2/{tenant_id}/os-floating-ip-dns/{domain}/entries/{name}	Finds a unique DNS entry for a specified domain and name
DELETE	v2/{tenant_id}/os-floating-ip-dns/{domain}/entries/{name}	Deletes a specified DNS entry
GET	v2/{tenant_id}/os-floating-ip-dns/{domain}/entries/{ip}	Lists DNS entries for a specified domain and IP

## Floating IP pools

Method	URI	Description
GET	v2/{tenant_id}/os-floating-ip-pools	Lists floating IP pools

## Floating IPs

Method	URI	Description
GET	v2/{tenant_id}/os-floating-ips	Lists floating IP addresses associated with the tenant or account
POST	v2/{tenant_id}/os-floating-ips	Allocates a new floating IP address to a tenant or account
GET	v2/{tenant_id}/os-floating-ips/{id}	Shows information for a specified floating IP address
DELETE	v2/{tenant_id}/os-floating-ips/{id}	Deallocates the floating IP address associated with floating_IP_address_ID
POST	v2/{tenant_id}/servers/{server_id}/action	Adds a floating IP address to an instance
POST	v2/{tenant_id}/servers/{server_id}/action	Removes a floating IP from an instance

## Floating IPs bulk

Method	URI	Description
GET	v2/{tenant_id}/os-floating-ips-	Lists all floating IPs

	bulk	
POST	v2/{tenant_id}/os-floating-ips-bulk	Bulk-creates floating IPs
POST	v2/{tenant_id}/os-floating-ips-bulk/delete	Bulk-deletes floating IPs
GET	v2/{tenant_id}/os-floating-ips-bulk/{host_name}	Lists all floating IPs for a specified host

## Hosts

Method	URI	Description
GET	v2/{tenant_id}/os-hosts	Lists hosts
GET	v2/{tenant_id}/os-hosts/{host_name}	Shows information for a specified host
PUT	v2/{tenant_id}/os-hosts/{host_name}	Enables a host or puts it in maintenance mode
GET	v2/{tenant_id}/os-hosts/{host_name}/startup	Starts a host
GET	v2/{tenant_id}/os-hosts/{host_name}/shutdown	Shuts down a host
GET	v2/{tenant_id}/os-hosts/{host_name}/reboot	Reboots a host

## Server Actions

Method	URI	Description
GET	v2/{tenant_id}/servers/{server_id}/os-instance-actions	Lists available actions for a specified server. Deployers set permissions for this request in the policy.json file. By default, all users can list actions
GET	v2/{tenant_id}/servers/{server_id}/os-instance-actions/{action_id}	Gets details for a specified action for a specified server instance. Deployers set permissions for this request in the policy.json file. By default, only administrators can get details for an action

## Keypairs

Method	URI	Description
GET	v2/{tenant_id}/os-keypairs	Lists keypairs that are associated with the account
POST	v2/{tenant_id}/os-keypairs	Generates or imports a keypair

DELETE	v2/{tenant_id}/os-keypairs/ {keypair_name}	Deletes a keypair
GET	v2/{tenant_id}/os-keypairs/ {keypair_name}	Shows a keypair associated with the account

## Networks

Method	URI	Description
POST	v2/{tenant_id}	Creates a network
GET	v2/{tenant_id}/os-networks	Lists networks that are available to the tenant
POST	v2/{tenant_id}/os-networks/add	Adds a specified network to a project
GET	v2/{tenant_id}/os-networks/ {id}	Shows information for a specified network
DELETE	v2/{tenant_id}/os-networks/ {id}	Deletes a specified network
POST	v2/{tenant_id}/os-networks/ {id}/action	Associates a specified network with a host
POST	v2/{tenant_id}/os-networks/ {id}/action	Disassociates the host from a specified network
POST	v2/{tenant_id}/os-networks/ {id}/action	Disassociates a specified network from a project so that the network can be reused
POST	v2/{tenant_id}/os-networks/ {id}/action	Disassociates the project from a specified network

## Server rescue and unrescue

Method	URI	Description
POST	v2/{tenant_id}/servers/ {server_id}/action	Puts a server in rescue mode. Changes status to RESCUE
POST	v2/{tenant_id}/servers/ {server_id}/action	Unrescues a server

## Server start and stop

Method	URI	Description
POST	v2/{tenant_id}/servers/ {server_id}/action	Starts a stopped server and changes its status to ACTIVE
POST	v2/{tenant_id}/servers/ {server_id}/action	Stops a running server and changes its status to

	{server_id}/action	STOPPED
--	--------------------	---------

## Security Groups

Method	URI	Description
GET	v2/{tenant_id}/os-security-groups	Lists security groups
POST	v2/{tenant_id}/os-security-groups	Creates a security group
GET	v2/{tenant_id}/os-security-groups/servers/{server_id}/os-security-groups	Lists security groups for a specified server
GET	v2/{tenant_id}/os-security-groups/{security_group_id}	Shows information for a specified security group
DELETE	v2/{tenant_id}/os-security-groups/{security_group_id}	Deletes a specified security group

## Rules for default security group

Method	URI	Description
GET	v2/{tenant_id}/os-security-group-rules	Lists default security group rules
POST	v2/{tenant_id}/os-security-group-rules	Creates a default security group rule
GET	v2/{tenant_id}/os-security-group-rules/{security_group_rule_id}	Shows information for a specified security group rule

## Server Password

Method	URI	Description
GET	v2/servers/{server_id}/os-server-password	Gets the administrative password for a specified server
DELETE	v2/servers/{server_id}/os-server-password	Clears the encrypted copy of the password in the metadata server. This is done after the client has retrieved the password and knows it doesn't need it in the metadata server anymore. The password for the server remains the same

## Manage services

Method	URI	Description
GET	v2/{tenant_id}/os-services	Lists running services
PUT	v2/{tenant_id}/os-services/enable	Enables scheduling for a service
PUT	v2/{tenant_id}/os-services/disable	Disables scheduling for a service
PUT	v2/{tenant_id}/os-services/disable-log-reason	Logs information to the service table about why a service was disabled
GET	v2/{tenant_id}/os-services/detail	Lists disabled services. If information exists, includes reasons why services were disabled

## Usage Reports

Method	URI	Description
GET	v2/{tenant_id}/os-simple-tenant-usage	Lists usage information for all tenants
GET	v2/{tenant_id}/os-simple-tenant-usage/{tenant_id}	Gets usage information for a tenant

## Virtual interfaces

Method	URI	Description
GET	v2/{tenant_id}/servers/{server_id}/os-virtual-interfaces	Lists the virtual interfaces for a specified instance
GET	v2/{tenant_id}/servers/{server_id}/os-virtual-interfaces	Shows the virtual interface for for a specified instance. Includes the OS-EXT-VIF-NET:net_id attribute that shows to which network the interface is attached

## Volume extensions

Method	URI	Description
GET	v1.1/{tenant_id}/os-volumes	Lists the volumes associated with the account
GET	v1.1/{tenant_id}/os-volumes/detail	Lists details for a specified volume
POST	v1.1/{tenant_id}/os-volumes/{volume_id}	Creates a volume

GET	v1.1/{tenant_id}/os-volumes/ {volume_id}	Shows information for a specified volume
DELETE	v1.1/{tenant_id}/os-volumes/ {volume_id}	Deletes a specified volume
GET	v1.1/{tenant_id}/os-volume- types	Lists volume types
GET	v1.1/{tenant_id}/os-volume- types/{volume_type_id}	Shows information for a specified volume type
POST	v1.1/{tenant_id}/os-snapshots	Creates a snapshot
GET	v1.1/{tenant_id}/os-snapshots	Lists snapshots
GET	v1.1/{tenant_id}/os- snapshots/detail	Lists details for a specified snapshot
GET	v1.1/{tenant_id}/os-snapshots/ {snapshot_id}	Shows information for a specified snapshot
DELETE	v1.1/{tenant_id}/os-snapshots/ {snapshot_id}	Deletes a specified snapshot from the account

## Volume attachments

Method	URI	Description
GET	v1.1/{tenant_id}/os-volumes	Lists the volumes associated with the account
GET	v1.1/{tenant_id}/os- volumes/detail	Lists details for a specified volume
POST	v1.1/{tenant_id}/os-volumes/ {volume_id}	Creates a volume
GET	v1.1/{tenant_id}/os-volumes/ {volume_id}	Shows information for a specified volume
DELETE	v1.1/{tenant_id}/os-volumes/ {volume_id}	Deletes a specified volume
GET	v1.1/{tenant_id}/os-volume- types	Lists volume types
GET	v1.1/{tenant_id}/os-volume- types/{volume_type_id}	Shows information for a specified volume type
POST	v1.1/{tenant_id}/os-snapshots	Creates a snapshot
GET	v1.1/{tenant_id}/os-snapshots	Lists snapshots
GET	v1.1/{tenant_id}/os- snapshots/detail	Lists details for a specified snapshot
GET	v1.1/{tenant_id}/os-snapshots/ {snapshot_id}	Shows information for a specified snapshot

DELETE	v1.1/{tenant_id}/os-snapshots/{snapshot_id}	Deletes a specified snapshot from the account
--------	---	---

## Volume attachments

Method	URI	Description
POST	v2/{tenant_id}/servers/{server_id}/os-volume_attachments	Attaches a volume to the specified server
GET	v2/{tenant_id}/servers/{server_id}/os-volume_attachments	Lists the volume attachments for a specified server
GET	v2/{tenant_id}/servers/{server_id}/os-volume_attachments/{attachment_id}	Shows details for the specified volume attachment
DELETE	v2/{tenant_id}/servers/{server_id}/os-volume_attachments/{attachment_id}	Deletes the specified volume attachment from a specified server

## Servers with Block device mapping format

Method	URI	Description
GET	v2/{tenant_id}/servers	Lists IDs, names, and links for all servers
POST	v2/{tenant_id}/servers	Creates a server with a block device mapping

## Server OS-EXT-IPS-MAC:mac\_addr extended attribute

Method	URI	Description
POST	v2/{tenant_id}/servers	Creates a server with the OS-EXT-IPS-MAC:mac_addr extended attribute
GET	v2/{tenant_id}/servers/{server_id}	Shows information for a specified server. Includes the OS-EXT-IPS-MAC:mac_addr extended attribute
GET	v2/{tenant_id}/servers/detail	Lists details for all servers. Includes the OS-EXT-IPS-MAC:mac_addr extended attribute

## Configuration drive

Method	URI	Description
POST	v2/{tenant_id}/servers	Creates a server with the configuration drive extended attribute
GET	v2/{tenant_id}/servers/{server_id}	Shows information for a specified server including the configuration drive extended attribute
GET	v2/{tenant_id}/servers/{server_id}/detail	Lists details for all servers including the configuration drive extended attribute

## Servers and images with disk config

Method	URI	Description
POST	v2/{tenant_id}/servers	Creates a server
GET	v2/{tenant_id}/servers/{server_id}	Shows information for a specified server
PUT	v2/{tenant_id}/servers/{server_id}	Updates a specified server
POST	v2/{tenant_id}/servers/{server_id}/action	Resizes a server
POST	v2/{tenant_id}/servers/{server_id}/action	Rebuilds a specified server
GET	v2/{tenant_id}/servers/detail	Lists servers
GET	v2/{tenant_id}/images/{image_id}	Gets information for a specified image
GET	v2/{tenant_id}/images/detail	Lists images

## Server IP type

Method	URI	Description
GET	v2/{tenant_id}/servers/{server_id}/action	Shows the type of IP assigned to a specified server, either fixed or floating
GET	v2/{tenant_id}/servers/detail	Lists all servers showing their IPs by type, either fixed or floating

## Server extended attributes

Method	URI	Description
GET	v2/servers	Lists detailed extended server attribute information for all servers



GET	v2/servers/{server_id}	Shows extended server attributes for a specified server
-----	------------------------	---

## Server extended status

Method	URI	Description
GET	v2/{tenant_id}/servers/{server_id}	Shows the extended status attributes in the response for a specified server
GET	v2/{tenant_id}/servers/detail	Lists the extended status attributes in the detailed response for all servers

## Servers multiple create

Method	URI	Description
POST	v2/{tenant_id}/servers	Creates one or more servers with an optional reservation ID

## Image Service API

These are used to create, update and delete image metadata records, enable users to share images with each other, and upload and download raw image data.

An image entity is represented by a JSON-encoded data structure and its raw binary data. An image entity has an identifier (ID) that is guaranteed to be unique within the endpoint to which it belongs. The ID is used as a token in request URIs to interact with that specific image.

An image is always guaranteed to have the following attributes: id, status, visibility, protected, tags, created\_at, file and self. The other attributes defined in the image schema below are guaranteed to be defined, but will only be returned with an image entity if they have been explicitly set.

## Images

Method	URI	Description
GET	v2/images	Lists public virtual machine (VM) images
POST	v2/images	Creates a virtual machine (VM) image
GET	v2/images/{image_id}	Gets details for a specified image
PATCH	v2/images/{image_id}	Updates a specified image
DELETE	v2/images/{image_id}	Deletes a specified image
PUT	v2/images/{image_id}/file	Uploads binary image data
GET	v2/images/{image_id}/file	Downloads binary image data
PUT	v2/images/{image_id}/tags/{tag}	Adds a specified tag to a specified image
DELETE	v2/images/{image_id}/tags/{tag}	Deletes a specified tag from a specified image

GET	v2/images/{image_id}/members	Lists the tenants with whom this image has been shared
POST	v2/images/{image_id}/members	Adds a specified tenant ID as an image member
GET	v2/images/{image_id}/members/{member_id}	Gets details for a specified image member
DELETE	v2/images/{image_id}/members/{member_id}	Deletes a specified tenant ID from the member list of the specified image
PUT	v2/images/{image_id}/members/{member_id}	Sets the specified status for the specified member of the specified image
GET	v1/images	Lists public VM images
POST	v1/images	Registers a virtual machine (VM) image
GET	v1/images/detail	Lists details for available images
GET	v1/images/{image_id}	Shows details for the specified image
PUT	v1/images/{image_id}	Updates an image, uploads an image file, or updates metadata for an image
DELETE	v1/images/{image_id}	Deletes the specified image
PUT	v1/images/{image_id}/members	Replaces a membership list for an image
PUT	v1/images/{image_id}/members/{owner_id}	Adds a member to an image
DELETE	v1/images/{image_id}/members/{owner_id}	Removes a member from an image
GET	v1/shared-images/{owner}	Lists the VM images shared with a specified owner. The owner ID is the tenant ID

## Image schemas

Method	URI	Description
GET	v2/schemas/images	Gets a json-schema document that represents an images entity
GET	v2/schemas/image	Gets a json-schema document that represents an image entity
GET	v2/schemas/members	Gets a json-schema document that represents an image members entity
GET	v2/schemas/member	Gets a json-schema document that represents an image member entity

## Object Storage API

These are used to manage the accounts, containers and objects in the object storage system.

Jio Cloud Object Storage is an affordable, redundant, scalable, and dynamic storage service offering. The core storage system is designed to provide a safe, secure, automatically re-sizing and network-accessible way to store data. You can store an unlimited quantity of files and each file can be as large as 5 GBs, plus with large object creation, you can upload and store objects of virtually any size.

Jio Cloud Object Storage allows users to store and retrieve files and content through a simple Web Service interface (ReST: Representational State Transfer). There are also language-specific APIs that utilize the ReSTful API but make it much easier for developers to integrate into their applications.

This section is intended to assist software developers who want to develop applications using the Jio Cloud Object Storage API. It fully documents the ReST application programming interface (API) that allows developers to interact with the storage components of the Jio Cloud Object Storage system. To use the information provided here, you should be familiar with:

- ReSTful web services
- HTTP/1.1

Method	URI	Description
GET	v1/{account}	Lists storage containers, sorted by name
HEAD	v1/{account}	Gets the number of containers and the total bytes that are stored in Object Storage for the tenant
POST	v1/{account}	Creates or updates account metadata
POST	v1/{account}	Deletes account metadata
GET	v1/{account}/{container}	Lists the objects stored in the container
PUT	v1/{account}/{container}	Creates a container
DELETE	v1/{account}/{container}	Deletes an empty container
HEAD	v1/{account}/{container}	Gets container metadata, including the number of objects and the total bytes of all objects stored in the container
POST	v1/{account}/{container}	Creates or updates the arbitrary container metadata by associating custom metadata headers with the container level URI. These headers must have the format, <b>X-Container-Meta- name</b>
GET	v1/{account}/{container}/{object}	Gets data for a specified object
PUT	v1/{account}/{container}/{object}	Creates or updates the content and metadata for a specified object

DELETE	v1/{account}/{container}/ {object}	Permanently deletes an object from the OpenStack Object Storage system. In combination with the COPY command, you can use COPY and then DELETE to effectively move an object
COPY	v1/{account}/{container}/ {object}	Copies an object to another object with a new name in the Object Storage system
HEAD	v1/{account}/{container}/ {object}	Gets object metadata and other standard HTTP headers
POST	v1/{account}/{container}/ {object}	Updates arbitrary key/value metadata. Metadata must be in the format X-Object-Meta-*. You can also assign X-Delete-At or X-Delete-After to expiring objects. You cannot use this operation to change other headers, such as Content-Type