

# APIs - REST APIs

[ PDF generated February 25 2026. For all recent updates please see the Nutanix Bible releases notes located at [https://nutanixbible.com/release\\_notes.html](https://nutanixbible.com/release_notes.html). Disclaimer: Downloaded PDFs may not always contain the latest information. ]

## REST API and SDK Introduction

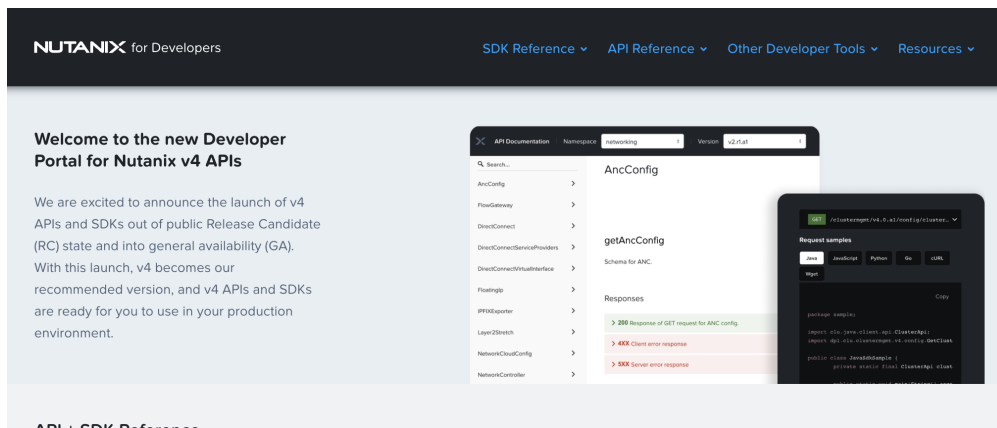
Note: The Nutanix v4 APIs are the recommended version for all use cases. This introduction will focus on the v4 Prism Central APIs; legacy APIs (v1, v2 and v3) should only be used when the a specific function or feature is not available in the v4 APIs.

The Prism Central REST API exposes nearly every capability and data point of the Prism Central UI and allows for orchestration or automation tools to easily drive Nutanix actions. This enables tools like Terraform, Ansible, Saltstack, Puppet, Ansible and more to easily create custom workflows for Nutanix. Also, this means that any third-party developer could create their own custom UI and pull in Nutanix data via REST.

The following figure shows a small snippet of the Nutanix REST API explorer which allows developers to interact with the API and see expected data formats.

## Nutanix v4 API and SDK Documentation

All Nutanix v4 API documentation is accessed through the new documentation portal, available at <https://developers.nutanix.com>. The screenshot below shows the home page for the Nutanix v4 API documentation.



Note: Legacy v3 APIs are still supported in the Nutanix REST API Explorer, available when logged into Prism Central.

## Nutanix v4 API: SDK Language Support

The following language-specific SDKs are currently available.

- Python (3.6, 3.7, 3.8 and 3.9 are officially supported)
- Java
- Go
- JavaScript

## REST API Explorer for legacy v2 APIs (Prism Element)

Legacy APIs are still supported in the Nutanix REST API Explorer, as shown in the screenshots below.

REST API Explorer   Version 2				For additional information, code samples, please refer to the Nutanix Developer Portal		
API Version 2						
alerts	Show/Hide	List Operations	Expand Operations			
authconfig	Show/Hide	List Operations	Expand Operations			
cloud	Show/Hide	List Operations	Expand Operations			
cluster	Show/Hide	List Operations	Expand Operations			
clusters	Show/Hide	List Operations	Expand Operations			
data_at_rest_encryption	Show/Hide	List Operations	Expand Operations			
data at rest encryption certificates	Show/Hide	List Operations	Expand Operations			

Operations can be expanded to display details and examples of the REST call:

REST API Explorer | Version 2 | VMs | Show/Hide | List Operations | Expand Operations

**VMs** | [Get a list of Virtual Machines.](#)

Implementation Nodes  
Get a list of Virtual Machines.

Virtual Machine disk information and network information are not included by default as fetching these are expensive operations. These can be included by setting the `include_vm_disk_config` and `include_vm_nic_config` flags respectively.

Response Class (Status 200)

Model | Example Value

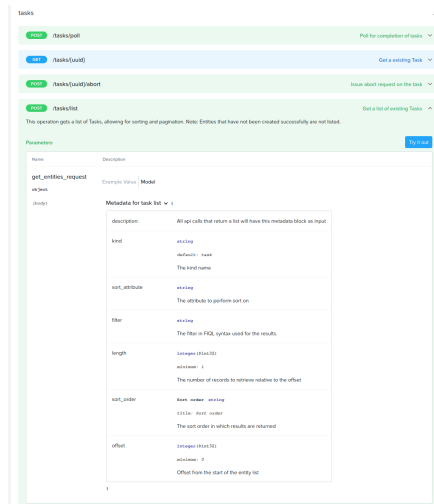
```
{
  "entities": [
    {
      "allow_live_migrate": true,
      "boot": {
        "boot_device_order": [
          "string"
        ],
        "boot_device_type": "CDROM",
        "disk_address": {
          "NetworkAdapter": "NetworkAdapter"
        }
      }
    }
  ]
}
```

Response Content Type: application/json

Parameter	Value	Description	Parameter Type	Data Type
filter	<input type="text"/>	Filter criteria - semicolon for AND, comma for OR	query	string
offset	<input type="text"/>	offset - Default 0	query	long
length	<input type="text"/>	Number of VMs to retrieve	query	long
sort_order	<input type="text"/>	Sort order	query	string
sort_attribute	<input type="text"/>	Sort Attribute	query	string
include_vm_disk_config	<input type="checkbox"/>	Whether to include Virtual Machine disk information.	query	boolean
include_vm_nic_config	<input type="checkbox"/>	Whether to include network information.	query	boolean

## REST API Explorer for legacy v3 APIs (Prism Central)

REST API Explorer   Version 3		For additional information, code samples, please refer to the Nutanix Developer Portal	
Nutanix Intentful API <span>3.1.0</span>			
[ Base URL: /api/nutanix/v3 ]			
<a href="#">/static/v3/swaggerjson</a>			
Move programming from the user to the machine.			
categories			▼
gcp/firewalls			▼
vm_recovery_points			▼
vmware/template			▼
portal_services			▼
tasks			▼
network_security_rules			▼
layer2 stretch			▼



## REST API Authentication

Nutanix v4 REST APIs support two authentication types:

- HTTP Basic Authentication by username and password
- Nutanix IAM API key authentication using built-in or environment-specific Authorization Policies

Authentication may be completed in two "modes":

1. Read-only: To collect and inspect information exposed by API only.
2. Administrative: All actions available to Read-Only user accounts plus entity management, including but not exclusive to CRUD operations on virtual machines, NCM Self-Service Blueprints and Nutanix Flow Network Security policies.

Nutanix IAM Authorization Policies can apply granular control over specific actions a user can complete.

## REST API Versions

Nutanix Prism Element and Prism Central currently offer three generally available APIs:

- v4, available in Prism Central only. Recommended API for all administrative activity such as virtual machine management and CRUD operations as well as Prism Central integrated product management such as NCM Self-Service and Flow Network Security.
- Legacy v3 API, available in Prism Central only. Fully supported until deprecation at a later date and used for multicloud activities such as virtual machine management, Nutanix product management including NCM Self-Service (formerly Calm), Flow Network Security.
- Legacy v2.0 API, available in Prism Element only. Fully supported until deprecation at a later date. Used for cluster-local activities such as storage container management operations, specifically in environments without a Prism Central deployment.

For detailed information on the available API versions and the operations they expose, see [API Versions](#) on [Nutanix.dev](#).

## Nutanix Products Providing APIs

A non-exhaustive list of Nutanix products that can be fully or partially managed by API is as follows. Documentation is linked, where appropriate.

- [Nutanix Prism Central: v4](#)
- [Nutanix Prism Central: v3](#)
- [Nutanix Prism Element](#)
- [NCM Self-Service](#)
- [Nutanix Database Service](#)

For a complete list of all API references, see [Nutanix.dev API Reference](#).

## Prism Central v4 APIs and SDKs

Current GA and primary REST API for all Prism Central programmatic operations.

## Prism Element v2.0 vs Prism Central v3 APIs

Note: Provided for legacy and backwards-compatibility support only.

The chosen API will depend on the required operations.

Prism Central v3 is multi-cluster and available through Prism Central only. It exposes operations that can impact multiple clusters such as distributed disk images, NCM Self-Service (formerly Calm) blueprints and apps, marketplace items and network security rules.

Prism Element v2.0 is cluster-local and available through Prism Element only. It exposes operations specific to the local cluster entities such as storage containers, data-at-rest encryption, storage pools and hosts.

## Usage Examples: Prism Central API v4 REST APIs

### Request

Accessible through Nutanix Prism Central only; all API requests are as constructed as follows.

```
METHOD https://PRISM_CENTRAL_IP_OR_FQDN:9440/api/NAMESPACE/VERSION/PATH/ACTIONS
```

Example of a request to list all Prism Central images, assuming the Prism Central IP is 192.168.1.110.

```
GET https://192.168.1.110:9440/api/vmm/v4.0/content/images
```

Example of image creation:

```
POST https://192.168.1.110:9440/api/vmm/v4.0/content/images
```

Accompanied by image configuration in JSON format. For example:

```
{
  "name": "{{image_name}}",
  "type": "DISK_IMAGE",
  "description": "Image created with Nutanix v4 APIs",
  "source": {
    "url": "{{image_qcow2_url}}",
    "$objectType": "vmm.v4.content.UrlSource"
  },
  "clusterLocationExtIds": [
    "{{cluster_extid}}"
  ]
}
```

To ensure request idempotency, this request must be accompanied by the

```
Ntnx-Request-Id
```

header, the value of which is a UUID that uniquely identifies this specific request. UUID generation will vary by environment but can be generated by various online tools.

### Response

The response from this request contains the task status, indicated by the

```
data.extId
```

field. This task, when viewed in Prism Central or via API, would show the operation's status, including percentage complete and any errors that occurred during the process.

```

{
  "data": {
    "$reserved": {
      "$fv": "v4.r0"
    },
    "$ObjectType": "prism.v4.config.TaskReference",
    "extId": "ZXJnb24=:447c5e9e-b7ad-408e-ae4f-82de667ef72e"
  },
  "$reserved": {
    "$fv": "v4.r0"
  },
  "$ObjectType": "vmm.v4.content.CreateImageApiResponse",
  "metadata": {
    "flags": [
      {
        "$reserved": {
          "$fv": "v1.r0"
        },
        "$ObjectType": "common.v1.config.Flag",
        "name": "hasError",
        "value": false
      },
      {
        "$reserved": {
          "$fv": "v1.r0"
        },
        "$ObjectType": "common.v1.config.Flag",
        "name": "isPaginated",
        "value": false
      },
      {
        "$reserved": {
          "$fv": "v1.r0"
        },
        "$ObjectType": "common.v1.config.Flag",
        "name": "isTruncated",
        "value": false
      }
    ],
    "$reserved": {
      "$fv": "v1.r0"
    },
    "$ObjectType": "common.v1.response.ApiResponseMetadata",
    "links": [
      {
        "$reserved": {
          "$fv": "v1.r0"
        },
        "$ObjectType": "common.v1.response.ApiLink",
        "href": "https://192.168.1.110:9440/api/prism/v4.0/config/tasks/ZXJnb24=:447c5e9e-b7ad-408e-ae4f-82de667ef72e",
        "rel": "self"
      },
      {
        "href": "https://192.168.1.110:9440/api/vmm/v4.0/content/images",
        "rel": "image-list"
      }
    ]
  }
}

```

To use this example in your own environment, change the

```
{{placeholder}}
```

variables to those matching your requirements.

## Related Resources

- [v4 REST API: Create Prism Central Image](#) complete REST API code sample on Nutanix.dev
- [v4 Python SDK: Create Prism Central Image](#) complete Python SDK code sample on Nutanix.dev
- [Nutanix v4 API Code Sample Library](#).

## Usage Examples: Prism API v3.0

### Request

Nutanix Prism API v3 is available through Prism Central only. All API requests are as constructed as follows.

```
METHOD https://PRISM_CENTRAL_IP:9440/api/nutanix/v3/API_NAME/VARIABLES
```

Example of a request to list all virtual machines, assuming the Prism Central IP is 192.168.1.110.

```
POST https://192.168.1.110:9440/api/nutanix/v3/vms/list
```

Accompanied by an appropriate payload in JSON format. In this example Prism Central is instructed to return all items of type **vm**.

```
{
  "kind": "vm"
}
```

Example of VM creation:

```
POST https://192.168.1.110:9440/api/nutanix/v3/vms
```

Accompanied by VM configuration in JSON format. For example:

```
{
  "spec": {
    "name": "vm_api_v3",
    "resources": {},
    "cluster_reference": {
      "uuid": "0005f2f7-eee7-1995-6145-ac1f6b35fe5e",
      "kind": "cluster"
    }
  },
  "metadata": {
    "kind": "vm"
  }
}
```

### Response

The response from this request contains the task status, currently **PENDING**, and task details including the cluster that owns the new VM and a **task\_uuid** that can be queried using the **tasks** API.

```
{
  "status": {
    "state": "PENDING",
    "execution_context": {
      "task_uuid": "1c64cc8b-241d-4132-955f-f6e26239ac02"
    }
  },
  "spec": {
    "name": "vm_api_v3",
    "resources": {},
    "cluster_reference": {
      "kind": "cluster",
      "uuid": "0005f2f7-eee7-1995-6145-ac1f6b35fe5e"
    }
  },
  "api_version": "3.1",
  "metadata": {
    "use_categories_mapping": false,
    "kind": "vm",
    "spec_version": 0,
    "uuid": "d965afdc-4606-4a0b-bc99-f868ae615039"
  }
}
```

```
}  
}
```

To use this example in your own environment, change the cluster **uuid** to a value appropriate for your cluster.

For more information see, [Create a new VM](#) in the Nutanix.dev API reference.

## Usage Examples: Prism API v2.0

### Request

Nutanix Prism API v2.0 is available through Prism Element only. All API requests are as constructed as follows.

```
METHOD https://CVM_OR_CLUSTER_IP:9440/api/nutanix/v2.0/API_NAME/VARIABLES
```

Example of a request to list all storage containers, assuming the cluster IP is 192.168.1.100.

```
GET https://192.168.1.100:9440/api/nutanix/v2.0/storage-containers
```

Example of VM creation:

```
POST https://192.168.1.100:9440/api/nutanix/v2.0/vms
```

Accompanied by VM configuration in JSON format. For example:

```
{  
  "description": "VM created by v2.0 API",  
  "memory_mb": 1024,  
  "name": "vm_api_v2.0",  
  "num_vcpus": 1,  
  "num_cores_per_vcpu": 1,  
  "vm_disks": [  
    {  
      "is_cdrom": false,  
      "vm_disk_create": {  
        "size": 128849018880,  
        "storage_container_uuid": "b2fefc62-6274-4c84-8e6c-a61f5313ea0e"  
      }  
    }  
  ]  
}
```

### Response

The response from this request contains a Prism task UUID that can be queried using the

```
prism
```

namespace's **tasks** API.

```
{  
  "task_uuid": "eec7d743-b6c7-4c6d-b821-89b91b142f1d"  
}
```

To use this example in your own environment, change **storagecontaineruuid** to a value appropriate for your cluster.

For more information see [Create a Virtual Machine](#) in the Nutanix.dev API reference.

Note: The **/api/nutanix/v2.0** path is an alias for **/PrismGateway/services/rest/v2.0**. The **PrismGateway** path will often be referenced in the Prism API v2.0 documentation; both paths can be used interchangeably and will produce identical results.