

# Skyport

Skyport is an open source Java application that enables users to use [Openstack Nova 2.0 API](#) to manage multicloud environment.

## Table of contents

- [Overview](#)
- [Installation](#)
- [Configuration](#)
  - [Profile](#)
  - [Security](#)
    - [Disabling Authentication](#)
    - [Integrate with](#) `Openstack Keystone` [or](#) `Keystone4j`
  - [Configure](#) `policy.json`
- [Using Rest API](#)
- [Release notes](#)
- [Acknowledgements](#)
- [License](#)

## Overview

---

For some reason, many of us use multiple virtualization platforms concurrently. But the more platforms you use, the more work it takes to keep up with them. Skyport is an application that provides a single API compatibility with [Openstack Nova 2.0 API](#) to work with multiple cloud computing providers.

## Installation

---

The latest binary downloads of the Skyport4 are available [here](#). Unpack the binary distribution so that it resides in its own directory (conventionally named "skyport[version]"). Skyport can be run as a service using the `skyport` script located under bin/ location. The script accepts a single parameter with the following values:

- console: Run Skyport in the foreground.
- start: Run Skyport in the background.

- stop: Stops Skyport if its running.
- install: Install Skyport to run on system startup (windows only).
- remove: Removes Skyport from system startup (windows only).

Skyport uses Java Service Wrapper which is a small native wrapper around the Java virtual machine which also monitors it. Note, passing JVM level configuration (such as -X parameters) should be set within the config/wrapper.conf file. The wrapper.java.maxmemory environment variable controls the maximum memory allocation for the JVM (set in megabytes). It defaults to 512. For more information about configuring Java Service Wrapper, refer to [Java Service Wrapper](#)

## Configuration

---

### Profile

To add or configure a profile on Skyport, use the Skyport Administrator GUI.

1. To create a profile, click the **Add** button.
2. You now need to configure the specific fields for the Cloud Services Profile you are creating through the Configuration dialog. 
  - **ID** :Enter the identifier of the profile to access (**Optional**). If the ID field is left blank, Skyport will automatically generate an ID to be assigned to the Profile.
  - **Name** :Enter some text to help identify the profile (**Required**). It can be any valid name that you choose.
  - **Provider Class** :Select the Cloud Services driver for this profile (**Required**).
  - **Provider** :Select the Cloud Service Provider you want to connect to (**Required**).
  - **Endpoint** :Enter the URL at which the Cloud Services API is sitting (**Required**).
  - **Account** :Enter the account/tenant ID to which your API keys are tied (**Required**).
  - **RegionId** :Enter the ID for the region you are working against (**Required**).
  - **AccessPublic** :Enter the public part of your API keys or username (**Required**).
  - **AccessPrivate** :Enter the private part of your API keys or password (**Required**).
3. To enable cache to reduce the response time of the API, select the **Cache** check box to enable caching.
4. To enable the timeout mechanism of the API , select the **Timeout** check box to enable the timeout mechanism.
5. To verify the profile using the parameter you have entered, click the **Test** button. If the connection could be made successfully, you will be notified with a *Connection Successful!* dialog.
6. You can configure a number of options for a specific profile by using the **Advance** button.   
Each profile holds three thread pools for segregating application resources between different types of

task - short-term, medium-term and long-term. You can configure these thread pools on the top of the Advance dialog:

- `Core size`: The number of threads to keep in the pool, even if they are idle.
- `Max size`: The maximum number of threads to allow in the pool.
- `Queue capacity`: The maximum work queue size of the thread pool.

By clicking on the function name within the tree view in the left-side panel, the right side of the Advance dialog will load the function settings and allow you to modify these settings.

- `ThreadPool`: The thread pool on which the function is assigned to.
- `Delay`: A fixed period between the end of the last invocation and the start of the next (caching only).
- `Timeout`: The maximum time to wait for function execution (timeout only).

7. Click **OK** to save the profile.

## Security

### Disabling Authentication

Skyport normally enforce the *noauth* authentication strategy. The *noauth* strategy still performs authentication, but does not validate any credentials. It provides administrative credentials only if 'admin' is specified as the username. This is configured in the *config/nova.conf* file:

```
[DEFAULT]
# The strategy to use for auth: noauth or keystone. (string
# value)
auth_strategy=noauth
```

With the noauth strategy registered, any access to the server with a token format like

`X-Auth-Token:{profile_id}:admin` will be granted admin privileges.

### Integrate with `Openstack Keystone` or `Keystone4j`

By using *Openstack Keystone* or [Keystone4j](#) as a common authentication and authorization mechanism, you have to set the auth strategy to *Keystone* and update the `[keystoneauthtoken]` section to point to the *Keystone internal API IP address* and change the default *admintenantname*, *adminuser*, and *admin\_password* to the correct values in the *config/nova.conf* file:

```
[DEFAULT]
# The strategy to use for auth: noauth or keystone. (string
# value)
auth_strategy=keystone

[keystone_authtoken]
auth_host = 127.0.0.1
auth_port = 35357
auth_protocol = http
admin_user = admin
admin_password = SuperSekretPassword
admin_tenant_name = service
```

Once the Keystone strategy is registered, each profile's ID will need to be configured to correspond with a separate Keystone tenant's ID to work with it.

## Configure policy.json

Skyport follows OpenStack's access control mechanism using the config/policy.json file to define access control rules that applies to each API. For more information about configuring policy.json, refer to [the policy.json file](#)

## Using Rest API

Skyport listens to port 9999 by default. The examples in this section use [cURL](#) commands. For information about the OpenStack Compute(Nova) 2.0 APIs, see [Openstack Nova 2.0 API Reference](#). The following cURL command is used to list servers:

```
curl -H "X-Auth-Token:{token}" http://localhost:9999/v2/{profile_id}/servers
```

example response:

```
{
  "servers" : [ {
    "id" : "f2f5ca29-6c6d-4241-ab05-cc631da2486e",
    "name" : "3TLTCrryMQiM",
    "links" : [ {
      "rel" : "self",
      "href" : "http://localhost:9999/v2/98f4cd52-fa8b-4e5f-a031-bc73686cf3fc/servers/f2f5ca29-6c6d-4241-ab05-cc631da2486e"
    }, {
      "rel" : "bookmark",
      "href" : "http://localhost:9999/98f4cd52-fa8b-4e5f-a031-bc73686cf3fc/servers/f2f5ca29-6c6d-4241-ab05-cc631da2486e"
    } ]
  } ]
```

```
  }, {
    "id" : "8d8679af-f2fd-42b3-9803-1a294b75f91a",
    "name" : "cirros",
    "links" : [ {
      "rel" : "self",
      "href" : "http://localhost:9999/v2/98f4cd52-fa8b-4e5f-a031-bc73686cf3fc/servers/8d8679af-f2fd-42b3-9803-1a294b75f91a"
    }, {
      "rel" : "bookmark",
      "href" : "http://localhost:9999/98f4cd52-fa8b-4e5f-a031-bc73686cf3fc/servers/8d8679af-f2fd-42b3-9803-1a294b75f91a"
    } ]
  }, {
    "id" : "127dbebd-d189-4124-8bd3-4a35dc638909",
    "name" : "cirrow-jcnet",
    "links" : [ {
      "rel" : "self",
      "href" : "http://localhost:9999/v2/98f4cd52-fa8b-4e5f-a031-bc73686cf3fc/servers/127dbebd-d189-4124-8bd3-4a35dc638909"
    }, {
      "rel" : "bookmark",
      "href" : "http://localhost:9999/98f4cd52-fa8b-4e5f-a031-bc73686cf3fc/servers/127dbebd-d189-4124-8bd3-4a35dc638909"
    } ]
  }, {
    "id" : "30d563ce-f7a7-476a-b7e9-9baa1b2a4453",
    "name" : "cirros",
    "links" : [ {
      "rel" : "self",
      "href" : "http://localhost:9999/v2/98f4cd52-fa8b-4e5f-a031-bc73686cf3fc/servers/30d563ce-f7a7-476a-b7e9-9baa1b2a4453"
    }, {
      "rel" : "bookmark",
      "href" : "http://localhost:9999/98f4cd52-fa8b-4e5f-a031-bc73686cf3fc/servers/30d563ce-f7a7-476a-b7e9-9baa1b2a4453"
    } ]
  } ]
}
```

## Dependencies

---

Java Runtime Environment 1.7 or above is required to run Skyport.

## Release notes

---

Check them here: [Release notes](#)

# Acknowledgements

---

The Skyport infrastructure relies upon these free and openly available projects: - [OpenStack compute\(Nova\)](#) - [Dasein clouds](#) - [Java Websockify](#)

# License

---

*Skyport* is available under the Apache License, Version 2.0.

(c) All rights reserved InfinitiesSoft Solutions Inc.