
Magnum UI Documentation

Release 11.0.1.dev3

OpenStack Foundation

Nov 09, 2022

Contents

1	Installation	1
1.1	Enabling in DevStack	1
1.2	Manual Installation	2
2	Configuration	3
2.1	Magnum UI specific settings	3
	CLUSTER_TEMPLATE_GROUP_FILTERS	3
	MAGNUM_INGRESS_CONTROLLERS	3
	MAGNUM_AVAILABLE_ADDONS	4
2.2	Horizon Settings	5
3	Contributor Guide	5
3.1	Source Code Reference	5
	magnum_ui package	5
	magnum_ui	12
	Python Module Index	13

1 Installation

1.1 Enabling in DevStack

Add this repo as an external repository into your `local.conf` file:

```
[[local|localrc]]
enable_plugin magnum-ui https://github.com/openstack/magnum-ui
```

1.2 Manual Installation

Begin by cloning the Horizon and Magnum UI repositories:

```
git clone https://github.com/openstack/horizon
git clone https://github.com/openstack/magnum-ui
```

Create a virtual environment and install Horizon dependencies:

```
cd horizon
python tools/install_venv.py
```

Set up your `local_settings.py` file:

```
cp openstack_dashboard/local/local_settings.py.example openstack_dashboard/
↪local/local_settings.py
```

Open up the copied `local_settings.py` file in your preferred text editor. You will want to customize several settings:

- `OPENSTACK_HOST` should be configured with the hostname of your OpenStack server. Verify that the `OPENSTACK_KEYSTONE_URL` and `OPENSTACK_KEYSTONE_DEFAULT_ROLE` settings are correct for your environment. (They should be correct unless you modified your OpenStack server to change them.)

Install Magnum UI with all dependencies in your virtual environment:

```
tools/with_venv.sh pip install -e ../magnum-ui/
```

And enable it in Horizon:

```
cp ../magnum-ui/magnum_ui/enabled/_1370_project_container_infra_panelgroup.py ↪
↪openstack_dashboard/local/enabled
cp ../magnum-ui/magnum_ui/enabled/_1371_project_container_infra_clusters_
↪panel.py openstack_dashboard/local/enabled
cp ../magnum-ui/magnum_ui/enabled/_1372_project_container_infra_cluster_
↪templates_panel.py openstack_dashboard/local/enabled
```

To run horizon with the newly enabled Magnum UI plugin run:

```
python manage.py runserver 0.0.0.0:8080
```

to have the application start on port 8080 and the horizon dashboard will be available in your browser at <http://localhost:8080/>

2 Configuration

2.1 Magnum UI specific settings

CLUSTER_TEMPLATE_GROUP_FILTERS

New in version 5.3.0: (Ussuri)

Default: None

Examples:

```
CLUSTER_TEMPLATE_GROUP_FILTERS = {  
    "dev": ".*-dev-.*",  
    "prod": ".*-prod-.*"  
}
```

The settings expects a dictionary of group name, to group regex.

When set allows a cloud provider to specify template groups for their cluster templates based on their naming convention. This helps limit users from upgrading their cluster to an invalid template that will not work based on their current template type.

This filtering is only relevant when choosing a new template for upgrading a cluster.

MAGNUM_INGRESS_CONTROLLERS

New in version 5.3.0: (Ussuri)

Default: None

Examples:

```
MAGNUM_INGRESS_CONTROLLERS = [  
    {  
        "name": "NGINX",  
        "labels": {  
            "ingress_controller": "nginx"  
        }  
    },  
    {  
        "name": "Traefik",  
        "labels": {  
            "ingress_controller": "traefik"  
        }  
    },  
    {  
        "name": "Octavia",  
        "labels": {  
            "ingress_controller": "octavia"  
        }  
    }  
]
```

This setting specifies which [Kubernetes Ingress Controllers](#) are supported by the deployed version of magnum and map directly to the response returned by the magnum-ui `api/container-infra/ingress_controllers` endpoint.

MAGNUM_AVAILABLE_ADDONS

New in version 5.3.0: (Ussuri)

Default: None

Examples:

```
MAGNUM_AVAILABLE_ADDONS = [  
  {  
    "name": "Kubernetes Dashboard",  
    "selected": True,  
    "labels": {  
      "kube_dashboard_enabled": True  
    },  
    "labels_unselected": {  
      "kube_dashboard_enabled": False  
    }  
  },  
  {  
    "name": "Influx Grafana Dashboard",  
    "selected": False,  
    "labels": {  
      "influx_grafana_dashboard_enabled": True  
    },  
    "labels_unselected": {  
      "influx_grafana_dashboard_enabled": False  
    }  
  }  
]
```

Specifies which Addon Software is available or supported in the deployed version of magnum and specifies which labels need to be included in order to enable or disable the Software Addon.

Examples of *Addon Software* include but are not limited to:

- [Kubernetes Dashboard](#)
- [Influx Grafana Dashboard](#)

Values specified in the `MAGNUM_AVAILABLE_ADDONS` setting map directly to the values returned in the response of the `api/container-infra/available_addons` endpoint.

2.2 Horizon Settings

For more configurations, see [Configuration Guide](#) in the Horizon documentation.

3 Contributor Guide

There is no topic specific to Magnum UI now.

See [Horizon Developer Documents](#)

3.1 Source Code Reference

magnum_ui package

Subpackages

magnum_ui.api package

Subpackages

magnum_ui.api.rest package

Submodules

magnum_ui.api.rest.magnum module

```
class magnum_ui.api.rest.magnum.Addons(**kwargs)
```

```
    Bases: django.views.generic.base.View
```

```
    get(request)
```

```
    url_regex = 'container_infra/available_addons/'
```

```
class magnum_ui.api.rest.magnum.Certificate(**kwargs)
```

```
    Bases: django.views.generic.base.View
```

```
    API for retrieving a single certificate
```

```
    delete(request, cluster_id)
```

```
        Rotate a certificate from a cluster.
```

```
        Returns HTTP 204 (no content) on successful deletion.
```

```
    get(request, cluster_id)
```

```
        Get a certificate from a cluster.
```

```
        Returns the CA.pem string on success
```

```
    url_regex = 'container_infra/certificates/(?P<cluster_id>[^/]+)$'
```

```

class magnum_ui.api.rest.magnum.Certificates(**kwargs)
    Bases: django.views.generic.base.View
    API for Magnum Certificates

    post(request)
        Create a new Certificate.

        Returns the new Cert.pem string from csr for a cluster on success.

    url_regex = 'container_infra/certificates/$'

class magnum_ui.api.rest.magnum.Cluster(**kwargs)
    Bases: django.views.generic.base.View
    API for retrieving a single cluster

    get(request, cluster_id)
        Get a specific cluster

    patch(request, cluster_id)
        Update a Cluster.

        Returns the Cluster object on success.

    url_regex = 'container_infra/clusters/(?P<cluster_id>[^/]+)$'

class magnum_ui.api.rest.magnum.ClusterResize(**kwargs)
    Bases: django.views.generic.base.View

    get(request, cluster_id)
        Get cluster details for resize

    post(request, cluster_id)
        Resize a cluster

    url_regex = 'container_infra/clusters/(?P<cluster_id>[^/]+)/resize$'

class magnum_ui.api.rest.magnum.ClusterTemplate(**kwargs)
    Bases: django.views.generic.base.View
    API for retrieving a single cluster template

    get(request, template_id)
        Get a specific cluster template

    patch(request, template_id)
        Update a Cluster Template.

        Returns the Cluster Template object on success.

    url_regex = 'container_infra/cluster_templates/(?P<template_id>[^/]+)$'

class magnum_ui.api.rest.magnum.ClusterTemplates(**kwargs)
    Bases: django.views.generic.base.View
    API for Magnum Cluster Templates

```

delete(*request*)

Delete one or more Cluster Templates by id.

Returns HTTP 204 (no content) on successful deletion.

get(*request*)

Get a list of the Cluster Templates for a project.

The returned result is an object with property items and each item under this is a Cluster Template.

If a GET query param for `related_to` is specified, and the setting for template filtering is set, then Horizon will only return template groups which the given template falls into, or all if none match.

post(*request*)

Create a new Cluster Template.

Returns the new ClusterTemplate object on success.

url_regex = 'container_infra/cluster_templates/\$'

class magnum_ui.api.rest.magnum.ClusterUpgrade(**kwargs)

Bases: django.views.generic.base.View

post(*request*, *cluster_id*)

Upgrade a cluster

url_regex = 'container_infra/clusters/(?P<cluster_id>[^/]+)/upgrade\$'

class magnum_ui.api.rest.magnum.Clusters(**kwargs)

Bases: django.views.generic.base.View

API for Magnum Clusters

delete(*request*)

Delete one or more Clusters by id.

Returns HTTP 204 (no content) on successful deletion.

get(*request*)

Get a list of the Clusters for a project.

The returned result is an object with property items and each item under this is a Cluster.

post(*request*)

Create a new Cluster.

Returns the new Cluster object on success.

url_regex = 'container_infra/clusters/\$'

class magnum_ui.api.rest.magnum.IngressControllers(**kwargs)

Bases: django.views.generic.base.View

get(*request*)

url_regex = 'container_infra/ingress_controllers/'

```

class magnum_ui.api.rest.magnum.Networks(**kwargs)
    Bases: django.views.generic.base.View
    API for Neutron networks for Cluster Templates creation

    get(request)
        Get a list of the Networks for a project.

        Networks includes external and private. Also, each network has subnets. The returned result
        is an object with property items and each item under this is a Network.

    url_regex = 'container_infra/networks/$'

class magnum_ui.api.rest.magnum.Quota(**kwargs)
    Bases: django.views.generic.base.View
    API for retrieving a single Quota

    delete(request, project_id, resource)
        Delete one Quota by id and resource.

        Returns HTTP 204 (no content) on successful deletion.

    get(request, project_id, resource)
        Get a specific quota

    patch(request, project_id, resource)
        Update a Quota.

        Returns the Quota object on success.

    url_regex =
    'container_infra/quotas/(?P<project_id>[^/]+)/(?P<resource>[^/]+)$'

class magnum_ui.api.rest.magnum.Quotas(**kwargs)
    Bases: django.views.generic.base.View
    API for Magnum Quotas

    get(request)
        Get a list of the Quotas for a project.

        The returned result is an object with property items and each item under this is a Quota.

    post(request)
        Create a new Quota.

        Returns the new Quota object on success.

    url_regex = 'container_infra/quotas/$'

class magnum_ui.api.rest.magnum.Stats(**kwargs)
    Bases: django.views.generic.base.View
    API for Magnum Stats

    get(request)
        Get a list of the Stats.

        The returned result is an object with property items and each item under this is a Stat.

```



```
url_regex = 'container_infra/stats/$'
```

```
magnum_ui.api.rest.magnum.change_to_id(obj)
```

Change key named uuid to id

Magnum returns objects with a field called uuid many of Horizons directives however expect objects to have a field called id.

Module contents

Submodules

magnum_ui.api.heat module

```
magnum_ui.api.heat.format_parameters(params)
```

```
magnum_ui.api.heat.heatclient(request, password=None)
```

```
magnum_ui.api.heat.stack_get(request, stack_id)
```

magnum_ui.api.magnum module

```
magnum_ui.api.magnum.certificate_create(request, **kwargs)
```

```
magnum_ui.api.magnum.certificate_rotate(request, id)
```

```
magnum_ui.api.magnum.certificate_show(request, id)
```

```
magnum_ui.api.magnum.cluster_create(request, **kwargs)
```

```
magnum_ui.api.magnum.cluster_delete(request, id)
```

```
magnum_ui.api.magnum.cluster_list(request, limit=None, marker=None, sort_key=None,  
                                   sort_dir=None, detail=True)
```

```
magnum_ui.api.magnum.cluster_resize(request, cluster_id, node_count,  
                                       nodes_to_remove=None, nodegroup=None)
```

```
magnum_ui.api.magnum.cluster_show(request, id)
```

```
magnum_ui.api.magnum.cluster_template_create(request, **kwargs)
```

```
magnum_ui.api.magnum.cluster_template_delete(request, id)
```

```
magnum_ui.api.magnum.cluster_template_list(request, limit=None, marker=None,  
                                             sort_key=None, sort_dir=None, detail=True)
```

```
magnum_ui.api.magnum.cluster_template_show(request, id)
```

```
magnum_ui.api.magnum.cluster_template_update(request, id, **kwargs)
```

```
magnum_ui.api.magnum.cluster_update(request, id, **kwargs)
```

```
magnum_ui.api.magnum.cluster_upgrade(request, cluster_uuid, cluster_template,
                                       max_batch_size=1, nodegroup=None)

magnum_ui.api.magnum.magnumclient(request)

magnum_ui.api.magnum.quotas_create(request, **kwargs)

magnum_ui.api.magnum.quotas_delete(request, project_id, resource)

magnum_ui.api.magnum.quotas_list(request, limit=None, marker=None, sort_key=None,
                                   sort_dir=None, all_tenants=True)

magnum_ui.api.magnum.quotas_show(request, project_id, resource)

magnum_ui.api.magnum.quotas_update(request, project_id, resource, **kwargs)

magnum_ui.api.magnum.stats_list(request, project_id=None)
```

Module contents

magnum_ui.content package

Subpackages

magnum_ui.content.cluster_templates package

Submodules

magnum_ui.content.cluster_templates.panel module

```
class magnum_ui.content.cluster_templates.panel.ClusterTemplates
    Bases: horizon.base.Panel
    name = 'Cluster Templates'
    slug = 'cluster_templates'
```

magnum_ui.content.cluster_templates.tests module

```
class magnum_ui.content.cluster_templates.tests.ClusterTemplateTests(methodName='runTest')
    Bases: openstack_dashboard.test.helpers.TestCase
    test_me()
```

magnum_ui.content.cluster_templates.urls module

Module contents

magnum_ui.content.clusters package

Submodules

magnum_ui.content.clusters.panel module

class magnum_ui.content.clusters.panel.Clusters

Bases: horizon.base.Panel

name = 'Clusters'

slug = 'clusters'

magnum_ui.content.clusters.tests module

class magnum_ui.content.clusters.tests.ClusterTests (*methodName='runTest'*)

Bases: openstack_dashboard.test.helpers.TestCase

test_me()

magnum_ui.content.clusters.urls module

Module contents

magnum_ui.content.container_infra package

Subpackages

magnum_ui.content.container_infra.quotas package

Submodules

magnum_ui.content.container_infra.quotas.panel module

class magnum_ui.content.container_infra.quotas.panel.Quotas

Bases: horizon.base.Panel

name = 'Quotas'

slug = 'container_infra.quotas'

magnum_ui.content.container_infra.quotas.urls module

Module contents

Module contents

Module contents

magnum_ui.enabled package

Module contents

Submodules

magnum_ui.version module

Module contents

magnum_ui

Python Module Index

m

- magnum_ui, 12
- magnum_ui.api, 10
- magnum_ui.api.heat, 9
- magnum_ui.api.magnum, 9
- magnum_ui.api.rest, 9
- magnum_ui.api.rest.magnum, 5
- magnum_ui.content, 12
- magnum_ui.content.cluster_templates, 11
- magnum_ui.content.cluster_templates.panel, 10
- magnum_ui.content.cluster_templates.tests, 10
- magnum_ui.content.cluster_templates.urls, 11
- magnum_ui.content.clusters, 11
- magnum_ui.content.clusters.panel, 11
- magnum_ui.content.clusters.tests, 11
- magnum_ui.content.clusters.urls, 11
- magnum_ui.content.container_infra, 12
- magnum_ui.content.container_infra.quotas, 12
- magnum_ui.content.container_infra.quotas.panel, 11
- magnum_ui.content.container_infra.quotas.urls, 12
- magnum_ui.enabled, 12
- magnum_ui.version, 12