

---

# **NodeConductor CostPlanning Documentation**

***Release 0.5.4***

**OpenNode**

**Nov 29, 2017**



---

## Contents

---

<b>1</b>	<b>Guide</b>	<b>3</b>
<b>2</b>	<b>API</b>	<b>5</b>
<b>3</b>	<b>Endpoints</b>	<b>7</b>
<b>4</b>	<b>License</b>	<b>11</b>
<b>5</b>	<b>Indices and tables</b>	<b>13</b>



This plugin allows to get a price estimate without actually creating the infrastructure. Consider example workflow:

- admin creates categories: webservers and databases;
- admin creates presets, Apache and MySQL, each preset is linked to set of default price list items;
- user creates new deployment plan for his customer;
- user selects several presets, for example 20 MySQL databases and 2 Apache servers;
- user selects service, for example Azure;
- price list items are found by matching default price list items of presets against selected service;
- total price for deployment plan is calculated;
- user generates and downloads PDF report with deployment plan details;
- user sends email with deployment plan details to another user.



### 1.1 Installation

- Install Waldur
- Clone Waldur CostPlanning repository

```
git clone https://github.com/opennode/nodeconductor-cost-planning.git
```

- Install Waldur cost planning plugin into Waldur virtual environment

```
cd /path/to/nodeconductor-cost-planning/  
python setup.py install
```





## CHAPTER 2

---

### API

---

CostPlanning API documentation goes here!



### 3.1 Cost planning

Cost planning

#### 3.1.1 `/api/deployment-presets/`

Supported actions and methods:

```
/api/deployment-presets/
```

Methods: GET

```
/api/deployment-presets/<uuid>/
```

Methods: GET

#### 3.1.2 `/api/deployment-plans/`

Supported actions and methods:

```
/api/deployment-plans/
```

Methods: GET, POST

Supported fields for creation:

- **name** – string

- **project** – link to `/api/projects/<uuid>/`
- **items** – list of `[{preset: {cores: integer, ram: integer, storage: integer, name: string}, quantity: integer}]`
- **certifications** – list of `[{link: URL, name: string, description: string}]`

Example request for creating deployment plan.

```
POST /api/deployment-plans/
Accept: application/json
Content-Type: application/json
Authorization: Token c84d653b9ec92c6cbac41c706593e66f567a7fa4
Host: example.com

{
  "name": "WebApps",
  "customer": "http://example.com/api/customers/2f8b4e0f101545508d52c7655d6386c8/
↪",
  "items": [
    {
      "preset": "http://example.com/api/deployment-presets/
↪2debb6d109954afaa03910babc6791a6/",
      "quantity": 1
    }
  ]
}
```

Run **PUT** request against `/api/deployment-plans/<uuid>/` to update deployment plan. Only name and list of items can be updated. List of items should have the same format as POST request. Only customer owner and staff can update deployment plan.

`/api/deployment-plans/<uuid>/`

Methods: GET, PUT, PATCH, DELETE

Supported fields for update:

- **name** – string
- **items** – list of `[{preset: {cores: integer, ram: integer, storage: integer, name: string}, quantity: integer}]`
- **certifications** – list of `[{link: URL, name: string, description: string}]`

Example rendering of deployment plan and configuration.

```
GET /api/deployment-plans/c218cbb2f56c4d52a82638ca9fffd85a/
Accept: application/json
Content-Type: application/json
Authorization: Token c84d653b9ec92c6cbac41c706593e66f567a7fa4
Host: example.com

{
  "url": "http://example.com/api/deployment-plans/
↪b12bb98a661749ffb02c8a8439299288/",
  "uuid": "b12bb98a661749ffb02c8a8439299288",
  "name": "Webapp for Monster Inc",
  "customer": "http://example.com/api/customers/790b3c131e894581b3dcf66796d9fa30/
↪",
  "items": [
    {
      "preset": {
        "url": "http://example.com/api/deployment-presets/
↪628cd853ba2a4ce7af5d4fff510b5bd2/",
        "uuid": "628cd853ba2a4ce7af5d4fff510b5bd2",
        "name": "MySQL",
        "category": "Databases",
        "variant": "Large",
        "ram": 2048,
        "cores": 2,
        "storage": 1024000
      },
      "quantity": 1
    }
  ]
}
```

Example request for creating deployment plan.

```
POST /api/deployment-plans/
Accept: application/json
Content-Type: application/json
Authorization: Token c84d653b9ec92c6cbac41c706593e66f567a7fa4
Host: example.com

{
  "name": "WebApps",
  "customer": "http://example.com/api/customers/2f8b4e0f101545508d52c7655d6386c8/
↪",
  "items": [
    {
      "preset": "http://example.com/api/deployment-presets/
↪2debb6d109954afaa03910ba1c6791a6/",
      "quantity": 1
    }
  ]
}
```

Run **PUT** request against `/api/deployment-plans/<uuid>/` to update deployment plan. Only name and list of items can be updated. List of items should have the same format as POST request. Only customer owner and staff can update deployment plan.

`/api/deployment-plans/<uuid>/evaluate/`

Methods: GET

## CHAPTER 4

---

### License

---

NodeConductor cost planning plugin is open-source under MIT license.





## CHAPTER 5

---

### Indices and tables

---

- `genindex`
- `search`