Oracle PaaS Documentation

Release 0.6.2

OpenNode

December 16, 2016

Contents

Plugin for custom Oracle PaaS

CHAPTER 1

Guide

1.1 Installation

- Install NodeConductor
- Clone NodeConductor nodeconductor_paas_oracle repository

```
git clone https://github.com/opennode/nodeconductor-paas-oracle.git
```

- Install NodeConductor nodeconductor_paas_oracle into NodeConductor virtual environment

```
cd /path/to/nodeconductor-paas-oracle/
python setup.py install
```

CHAPTER 2

Endpoints

2.1 Oracle

Oracle

2.1.1 /api/oracle/

A filter backend that uses django-filter. Supported actions and methods:

/api/oracle/

Methods: `GET`, `POST`
Supported fields for creation:

- **name** – `string`
- project – `link to /api/projects/<uuid>/`
- **customer** – `link to /api/customers/<uuid>/`
- **settings** – `link to /api/service-settings/<uuid>/`
- token – `string` (JIRA project key (e.g. 'GM'))
- available_for_all – `boolean` (Service will be automatically added to all customers projects if it is available for all)
- scope – `link to any: /api/oracle-deployments/<uuid>/, /api/openstack-instances/<uuid>/, /api/openstack-tenants/<uuid>/, /api/openstack-volumes/<uuid>/, /api/openstack-snapshots/<uuid>/, /api/openstack-dr-backups/<uuid>/, /api/openstacktenant-volumes/<uuid>/, /api/openstacktenant-snapshots/<uuid>/, /api/openstacktenant-instances/<uuid>/, /api/openstacktenant-backups/<uuid>/, /api/jira-projects/<uuid>/` (VM that contains service)

Filter fields:

- ?customer = `UUIDFilter`
- ?name = `string`
- ?settings = `link`
- ?project_uuid = `UUIDFilter`
- ?project = `link`
- ?tag = `ModelMultipleChoiceField`
- ?rtag = `ModelMultipleChoiceField`
- ?shared = `boolean`
- ?type = `ServiceTypeFilter`

To list all services without regard to its type, run **GET** against */api/services/* as an authenticated user.
To list services of specific type issue **GET** to specific endpoint from a list above as a customer owner. Individual endpoint used for every service type.
To create a service, issue a **POST** to specific endpoint from a list above as a customer owner. Individual endpoint used for every service type.
You can create service based on shared service settings. Example:

```
POST /api/digitalocean/ HTTP/1.1
Content-Type: application/json
Accept: application/json
Authorization: Token c84d653b9ec92c6cbac41c706593e66f567a7fa4
Host: example.com

{
    "name": "Common DigitalOcean",
    "customer": "http://example.com/api/customers/1040561ca9e046d2b74268600c7e1105/",
    "settings": "http://example.com/api/service-settings/93ba615d6111466ebe3f792669059cb1/"
}
```

Or provide your own credentials. Example:

```
POST /api/oracle/ HTTP/1.1
Content-Type: application/json
Accept: application/json
Authorization: Token c84d653b9ec92c6cbac41c706593e66f567a7fa4
Host: example.com

{
    "name": "My Oracle",
    "customer": "http://example.com/api/customers/1040561ca9e046d2b74268600c7e1105/",
    "backend_url": "https://oracle.example.com:7802/em",
    "username": "admin",
    "password": "secret"
}
```

/api/oracle/<uuid>/

Methods: GET, PUT, PATCH, DELETE
Supported fields for update:

- **name** – string
- available_for_all – boolean (Service will be automatically added to all customers projects if it is available for all)

/api/oracle/<uuid>/link/

Methods: GET, POST
To get a list of resources available for import, run **GET** against */<service_endpoint>/link/* as an authenticated user. Optionally project_uuid parameter can be supplied for services requiring it like OpenStack.
To import (link with NodeConductor) resource issue **POST** against the same endpoint with resource id.

```
POST /api/openstack/08039f01c9794efc912f1689f4530cf0/link/ HTTP/1.1
Content-Type: application/json
Accept: application/json
Authorization: Token c84d653b9ec92c6cbac41c706593e66f567a7fa4
Host: example.com

{
    "backend_id": "bd5ec24d-9164-440b-a9f2-1b3c807c5df3",
    "project": "http://example.com/api/projects/e5f973af2eb14d2d8c38d62bcbaccb33/"
}
```

/api/oracle/<uuid>/managed_resources/

Methods: GET

/api/oracle/<uuid>/unlink/

Methods: POST
Unlink all related resources, service project link and service itself.

2.1.2 /api/oracle-service-project-link/

A filter backend that uses django-filter. Supported actions and methods:

/api/oracle-service-project-link/

Methods: `GET`, `POST`
Supported fields for creation:
- **project** – `link to /api/projects/<uuid>/`
- **service** – `link to /api/oracle/<uuid>/`

Filter fields:
- ?project = `link`
- ?service_uuid = `UUIDFilter`
- ?customer_uuid = `UUIDFilter`
- ?project_uuid = `UUIDFilter`

To get a list of connections between a project and an service, run **GET** against service_project_link_url as authenticated user. Note that a user can only see connections of a project where a user has a role.
If service has *available_for_all* flag, project-service connections are created automatically. Otherwise, in order to be able to provision resources, service must first be linked to a project. To do that, **POST** a connection between project and a service to service_project_link_url as stuff user or customer owner.

/api/oracle-service-project-link/<pk>/

Methods: `GET`, `DELETE`
To remove a link, issue **DELETE** to URL of the corresponding connection as stuff user or customer owner.

2.1.3 /api/oracle-flavors/

A filter backend that uses django-filter. Supported actions and methods:

/api/oracle-flavors/

Methods: `GET`
Filter fields:
- ?name = `string`

/api/oracle-flavors/<uuid>/

Methods: `GET`

2.1.4 /api/oracle-deployments/

SLA filter

Allows to filter or sort resources by actual_sla Default period is current year and month.

Example query parameters for filtering list of OpenStack instances:

```
/api/openstack-instances/?actual_sla=90&period=2016-02
```

Example query parameters for sorting list of OpenStack instances:

```
/api/openstack-instances/?o=actual_sla&period=2016-02
```

Monitoring filter

Filter and order resources by monitoring item. For example, given query dictionary

```
{
    'monitoring__installation_state': True
}
```

it produces following query

```
{
    'monitoring_item__name': 'installation_state',
    'monitoring_item__value': True
}
```

Example query parameters for sorting list of OpenStack instances:

```
/api/openstack-instances/?o=monitoring__installation_state
```

Tags ordering. Filtering for complex tags.

Example: ?tag__license-os=centos7 - will filter objects with tag "license-os:centos7".

Allow to define next parameters in view:

- tags_filter_db_field - name of tags field in database. Default: tags.
- tags_filter_request_field - name of tags in request. Default: tag.

In PostgreSQL NULL values come *last* with ascending sort order. In MySQL NULL values come *first* with ascending sort order. This filter provides unified sorting for both databases. Supported actions and methods:

/api/oracle-deployments/

Methods: GET, POST
Supported fields for creation:

- **name** – string
- description – string
- **service_project_link** – link to /api/oracle-service-project-link/<pk>/
- **tenant** – link to /api/openstack-tenants/<uuid>/
- **flavor** – link to /api/oracle-flavors/<uuid>/
- ssh_public_key – link to /api/keys/<uuid>/
- db_name – string
- **db_size** – integer (Data storage size in GB)
- db_arch_size – integer (Archive storage size in GB)
- **db_type** – choice('No database', 'RAC', 'Single Instance', 'Single Instance/ASM')
- db_version – choice('11.2.0.4', '12.1.0.2')
- db_template – choice('Data Warehouse', 'General Purpose')
- db_charset – choice('AL32UTF8 - Unicode UTF-8 Universal Character Set', 'AR8ISO8859P6 - ISO 8859-6 Latin/Arabic', 'AR8MSWIN1256 - MS Windows Code Page 1256 8-Bit Latin/Arabic', 'Other - please specify in Addtional Data field.')
- user_data – string

Filter fields:

- ?customer_native_name = string
- ?service_settings_name = string
- ?service_name = string
- ?customer_uuid = UUIDFilter
- ?customer_abbreviation = string
- ?tag = ModelMultipleChoiceField
- ?customer_name = string
- ?uuid = UUIDFilter
- ?project_group = UUIDFilter
- ?state = choice('Creating', 'Creation Scheduled', 'Deleting', 'Deletion Scheduled', 'Erred', 'OK', 'Update Scheduled', 'Updating')
- ?rtag = ModelMultipleChoiceField
- ?project_name = string
- ?description = string
- ?service_uuid = UUIDFilter
- ?service_settings_uuid = UUIDFilter
- ?db_name = string
- ?customer = UUIDFilter
- ?name = string
- ?project_uuid = UUIDFilter
- ?project = UUIDFilter
- ?project_group_uuid = UUIDFilter
- ?project_group_name = string

Order fields: state

Request for DB Instance deletion or confirm deletion success. A proper action will be taken depending on the current deployment state.

/api/oracle-deployments/<uuid>/

Methods: GET, PUT, PATCH, DELETE
Supported fields for update:

- **name** – string
- description – string

Optional *field* query parameter (can be list) allows to limit what fields are returned. For example, given request /api/openstack-instances/<uuid>/?field=uuid&field=name you get response like this:

```
{
    "uuid": "90bcfe38b0124c9bbdadd617b5d739f5",
    "name": "Azure Virtual Machine"
}
```

Request for DB Instance deletion or confirm deletion success. A proper action will be taken depending on the current deployment state.

/api/oracle-deployments/<uuid>/provision/

Methods: POST
Complete provisioning. Example:

```
    POST /api/oracle-deployments/a04a26e46def4724a0841abcb81926ac/provision/ HTTP/1.1
    Content-Type: application/json
    Accept: application/json
    Authorization: Token c84d653b9ec92c6cbac41c706593e66f567a7fa4
    Host: example.com

    {
        "report": "ORACONF=TST12DB
```

DBTYPE=single DBNAME='TST12DB'"
}

/api/oracle-deployments/<uuid>/resize/

Methods: POST
Request for DB Instance resize. Example:

```
POST /api/oracle-deployments/a04a26e46def4724a0841abcb81926ac/resize/ HTTP/1.1
Content-Type: application/json
Accept: application/json
Authorization: Token c84d653b9ec92c6cbac41c706593e66f567a7fa4
Host: example.com

{
    "flavor": "http://example.com/api/oracle-flavors/ef86802458684056b18576a91daf7690/"
}
```

To confirm resize completion, issue an empty POST request to the same endpoint.

/api/oracle-deployments/<uuid>/restart/

Methods: `POST`
Request for DB Instance restarting or confirm restarting success. A proper action will be taken depending on the current deployment state.

/api/oracle-deployments/<uuid>/start/

Methods: `POST`
Request for DB Instance starting or confirm starting success. A proper action will be taken depending on the current deployment state.

/api/oracle-deployments/<uuid>/stop/

Methods: `POST`
Request for DB Instance stopping or confirm stopping success. A proper action will be taken depending on the current deployment state.

/api/oracle-deployments/<uuid>/support/

Methods: `POST`
File custom support request.

```
POST /api/oracle-deployments/a04a26e46def4724a0841abcb81926ac/support/ HTTP/1.1
Content-Type: application/json
Accept: application/json
Authorization: Token c84d653b9ec92c6cbac41c706593e66f567a7fa4
Host: example.com

{
    "message": "Could you make that DB running faster?
```

Thanks." }

/api/oracle-deployments/<uuid>/unlink/

Methods: `POST`

/api/oracle-deployments/<uuid>/update_report/

Methods: `POST`
Update provisioning report. Example:

```
POST /api/oracle-deployments/a04a26e46def4724a0841abcb81926ac/update_report/ HTTP/1.1
Content-Type: application/json
Accept: application/json
Authorization: Token c84d653b9ec92c6cbac41c706593e66f567a7fa4
Host: example.com

{
    "report": "ORACONF=TST12DB
```

DBTYPE=single DBNAME='PRD12DB"'
}

CHAPTER 3

License

NodeConductor Oracle PaaS plugin is open-source under MIT license.

CHAPTER 4

Indices and tables

- genindex
- search