
nocloutdotnet

Release 1.0.5

unknown

Jun 07, 2022

GETTING STARTED

1	Installation	3
2	Usage	7
2.1	Instance Data	7
2.2	Phone home	8
3	Plugins	9
3.1	nocloutdotnet.instanceid	9
4	Endpoints	11
5	Metrics	13
6	Changelog	15
	Python Module Index	17
	Index	19

Cloud-init is the *industry standard* multi-distribution method for cross-platform cloud instance initialization. It is supported across all major public cloud providers, provisioning systems for private cloud infrastructure, and bare-metal installations.

Cloud instances are initialized from a disk image and instance data:

- Cloud metadata
- User data (optional)
- Vendor data (optional)

NoCloudDotNet is a back-end for the [NoCloud Data Source](#).

INSTALLATION

A production setup should involve an Apache/mod_wsgi installation and a RDBMS backend. The nocloudnet server itself is deployed as a wheel and should be installed onto the host along with apache and mod_wsgi.

An example Apache configuration:

```
#
# This file is part of NoCloud.Net.
#
# Copyright (C) 2022 Last Bastion Network Pty Ltd
#
# NoCloud.Net is free software: you can redistribute it and/or modify it under the
# terms of the GNU General Public License as published by the Free Software
# Foundation, either version 3 of the License, or (at your option) any later version.
#
# NoCloud.Net is distributed in the hope that it will be useful, but WITHOUT ANY
# WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A
# PARTICULAR PURPOSE. See the GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License along with
# NoCloud.Net. If not, see <https://www.gnu.org/licenses/>.
#

Listen 5000
<VirtualHost *:5000>
    ServerName nocloud.last-bastion.net
    DocumentRoot /var/www/html

    # TODO - you need these as env vars within systemd/init of Apache
    SetEnv FLASK_APP nocloudnet
    SetEnv FLASK_ENV production
    SetEnv NOCLOUDNET_SETTINGS /path/to/settings.yaml

    WSGIDaemonProcess nocloudnet display-name=%{GROUP} user=apache group=apache threads=5
    request-timeout=30
    WSGIScriptAlias / /usr/lib/python3.8/site-packages/nocloudnet/app.wsgi

    <Directory /usr/lib/python3.8/site-packages/nocloudnet>
        WSGIProcessGroup nocloudnet
        WSGIApplicationGroup %{GLOBAL}
        #Order deny,allow
```

(continues on next page)

(continued from previous page)

```
#Allow from all
  Require all granted
</Directory>
</VirtualHost>
```

The nocluddotnet application uses [Dynaconf](#) for configuration: there are many ways to set/override variables. An example config is:

```
---
#
# This file is part of NoCloud.Net.
#
# Copyright (C) 2022 Last Bastion Network Pty Ltd
#
# NoCloud.Net is free software: you can redistribute it and/or modify it under the
# terms of the GNU General Public License as published by the Free Software
# Foundation, either version 3 of the License, or (at your option) any later version.
#
# NoCloud.Net is distributed in the hope that it will be useful, but WITHOUT ANY
# WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A
# PARTICULAR PURPOSE. See the GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License along with
# NoCloud.Net. If not, see <https://www.gnu.org/licenses/>.
#
default:
  DEBUG: false
  VENDOR_DATA: ''
  USER_DATA: ''
  SQLALCHEMY_TRACK_MODIFICATIONS: false
  DOMAIN: last-bastion.net
  INSTANCE_TYPE: mymachinetype
  # stevedore nocluddotnet.instanceid namespace value...
  INSTANCEID: reversedns

development:
  DEBUG: true
  SECRET_KEY: dev
  SQLALCHEMY_DATABASE_URI: 'sqlite:///nocloud.net.db'

test:
  DEBUG: true
  SQLALCHEMY_DATABASE_URI: 'sqlite:///memory:'

production:
  SQLALCHEMY_DATABASE_URI: 'postgresql://user:password@localhost/nocloud'
  SECRET_KEY: prod
```

In order to create a/the RDBMS as per your configuration; you may need to do something along the lines of the following:

Listing 1: database setup

```
export FLASK_APP=nocluddotnet.app
export FLASK_ENV=production
export NOCLOUD_DOT_NET_SETTINGS=<path to settings.yaml>
flask createdb
```


USAGE

Once you have a running `nocloudnet`, you need to configure your consumer Linux physical/virtual machines to come under your cloud-init regime.

To do this permanently, you should edit `/etc/default/grub` and the `GRUB_CMDLINE_LINUX` should include `ds=nocloud-net;s=<nocloudnet ip>:<nocloudnet port>`

You will then need to run the following:

Listing 1: regenerate grub

```
grub2-mkconfig
```

You may also do one-off cloud-inits by editing the boot loader line via the grub menu on machine startup.

To close the loop; making your system secure; you would bake the grub configuration into your machine image and password protect your grub/boot configuration:

Listing 2: secure grub

```
grub2-setpassword  
vi /boot/grub2/user.cfg  
grub2-mkconfig
```

2.1 Instance Data

It is perhaps beyond the scope of this document to discuss how you deploy a custom `/etc/cloud/cloud.cfg` and any scripts into `/var/lib/cloud` that is a task for the tool(s) you use to create machine images; and/or orchestrate/configure your machines.

Your metaserver does, however, support serving [instance metadata](#). Instance (and vendor) data may be prepared/bundled into files as per [cloud-init formats](#) using cloud-init tools (or otherwise) and may be deployed by placing them in the configured paths for `VENDOR_DATA`, `USER_DATA`

2.2 Phone home

Cloud-init has a [phone home](#) module which you can configure to point to your nocloud.net server to capture the data publishable by this mechanism.

PLUGINS

We use `stevedore` to allow you to extend the application. To do this, you need to register endpoint(s) in a module. There is a configuration setting of the second component of the namespace, and that should be the function name you've chosen.

3.1 `nocluddotnet.instanceid`

3.1.1 `reversedns`

Make an instance-id based upon reverse lookup of remote address.

param request
flask request object

returns
instance id, hostname tuple

3.1.2 `simple`

Return a uuid-based hostname.

param request
flask request object

returns
instance id, hostname tuple

ENDPOINTS

Endpoint	Methods	Rule
-----	-----	-----
prometheus_metrics	GET	/metrics
static	GET	/static/<path:filename>

`nocluddotnet.instance.routes.debug()`

Show debug info; from request.

Returns

yaml of request and application configuration

`nocluddotnet.instance.routes.fetch()`

Return all registered instance records.

Returns

yaml instance data responding to query

`nocluddotnet.instance.routes.meta_data()`

Respond to meta-data request; either returning previously associated record or generating a new one.

Returns

yaml instance/host information

`nocluddotnet.instance.routes.phone_home()`

A cloud-init phone-home data/save. The phone-home url should be /phone-home?instance_id=\$INSTANCE_ID
Note that a phone-home call only happens once per cloud-instance.

Returns

http return code

`nocluddotnet.instance.routes.user_data()`

User data (scripts).

Returns

gzip/blob of cloud-int formatted user data

`nocluddotnet.instance.routes.vendor_data()`

Vendor data (scripts).

Returns

gzip/blob of cloud-int formatted user data

METRICS

Prometheus metrics may be gathered using the */metrics* endpoint.

CHANGELOG

- **pin sphinx version** by *Alan Milligan* at 2022-06-07 13:38:59
- **fallback for instance_path** by *Alan Milligan* at 2022-06-07 13:14:44
- **noclouddotnet into rtd/docs requirements** by *Alan Milligan* at 2022-06-07 12:15:01
- **explicit pip install for rtd** by *Alan Milligan* at 2022-06-06 17:22:24
- **explicit pip install for rtd** by *Alan Milligan* at 2022-06-06 16:13:54
- **added path to get noclouddotnet into docs** by *Alan Milligan* at 2022-06-06 15:52:32
- **install_requires to requirements.txt - for rtd** by *Alan Milligan* at 2022-06-06 15:45:24
- **version file implementation; for rtd** by *Alan Milligan* at 2022-06-06 14:58:42
- **prometheus metrics + jaeger tracing** by *Alan Milligan* at 2022-06-06 13:23:32
- **documentation requirements into requirements-docs.txt - for rtd builds** by *Alan Milligan* at 2022-06-03 06:29:20

PYTHON MODULE INDEX

n

`nocloutdotnet.instance.routes`, [11](#)

INDEX

D

`debug()` (*in module `nocluddotnet.instance.routes`*), 11

F

`fetch()` (*in module `nocluddotnet.instance.routes`*), 11

M

`meta_data()` (*in module `nocluddotnet.instance.routes`*),
11

`module`
`nocluddotnet.instance.routes`, 11

N

`nocluddotnet.instance.routes`
`module`, 11

P

`phone_home()` (*in module `nocluddotnet.instance.routes`*), 11

U

`user_data()` (*in module `nocluddotnet.instance.routes`*),
11

V

`vendor_data()` (*in module `nocluddotnet.instance.routes`*), 11