How to Deploy Virtual Test Agents in OpenStack

Executive Summary

This guide explains how to deploy Netrounds’ Virtual Test Agents (vTAs) in OpenStack and how to control these from Netrounds Control Center.

In ETSI NFV terminology, these vTAs are likewise referred to as Virtual Test Agents, and the centralized controller is referred to as the combination of Test Controller (TC) and Test Result Analysis Module (TRAM).

This guide assumes that you have a basic knowledge of OpenStack and Network Function Virtualization (NFV), and that you have your own OpenStack environment in which to launch the Netrounds vTA images.

Table of Contents

1 Netrounds Solution Overview ..............................................................................................................................................2
2 Prerequisites .......................................................................................................................................................................2
  2.1 Netrounds Control Center Account ...................................................................................................................................2
  2.2 Netrounds vTA Image .........................................................................................................................................................2
3 Launching the vTA Image in OpenStack .............................................................................................................................3
  3.1 Uploading the vTA Image to OpenStack ............................................................................................................................3
  3.2 Creating a Flavor for the vTA Image ....................................................................................................................................3
  3.3 Launching an Instance of the vTA Image ............................................................................................................................4
    3.3.1 Launching a vTA Image Manually in Horizon ..................................................................................................................4
    3.3.1.1 Format of “cloud-config” Metadata ..............................................................................................................................5
    3.3.2 Launching a vTA Image Using a Heat Orchestration Template in Horizon ............................................................................5
    3.3.2.1 Example of HOT File ....................................................................................................................................................6
    3.3.3 Launching a vTA Image Using the Python OpenStack API .............................................................................................7
4 Appendix: Description of the vTA VNF and Its Requirements ...........................................................................................9
1  **Netrounds Solution Overview**

Netrounds consists of two parts:

1. **Test Agents** – software-based active traffic generators. Virtual Test Agents (vTAs) are ones that you upload and boot from your own OpenStack environment. These vTAs will automatically connect to Netrounds Control Center as part of the deployment process described in this guide. (Netrounds also offers non-virtual Test Agents in the form of software that is installed on stand-alone x86 hardware.)

2. **Netrounds Control Center** – for centralized control and coordination of Test Agents, including distributed VNF vTAs. This includes initiating test sequences and monitoring sessions, as well as evaluating collected measurement data, SLAs and KPIs.

Netrounds vTAs are controlled through Netrounds Control Center. The interface towards Netrounds Control Center is either a web GUI or an orchestration API, as illustrated below:

2  **Prerequisites**

2.1  **Netrounds Control Center Account**

You need an account in a Netrounds Control Center in order to access it: either the one belonging to the Netrounds SaaS solution or one installed on-premise in your organization. If you do not already have a Netrounds account, please contact sales@netrounds.com or (in the SaaS case) reach out to our regional sales responsible for the SaaS solution.

2.2  **Netrounds vTA Image**

The VNF vTA image is provided either directly from Netrounds or from one of Netrounds’ partners. The vTA image for OpenStack is provided in either raw or QCOW2 format.
3 Launching the vTA Image in OpenStack

3.1 Uploading the vTA Image to OpenStack
First you need to upload the vTA image to your NFV/OpenStack platform. In OpenStack’s Horizon GUI, do as follows:

- On the navigation bar on the left, expand Project, and under Compute, click Images.
- Click the Create Image button. Fill out the dialog that appears, and create the image.

3.2 Creating a Flavor for the vTA Image
Flavors in OpenStack are virtual hardware templates, defining RAM allocation, block storage, and number of CPU cores. In the Horizon GUI, flavors are shown and created under Admin → System → Flavors.

The minimum recommended flavor for a vTA is:

- 1 vCPU
- 512 MB RAM
- 2 GB block storage

See below for some examples of flavors, including one called “netrounds.small.2GB” which corresponds to the above specifications.
3.3 Launching an Instance of the vTA Image

The final step is to launch an instance of the vTA image in OpenStack. Things will look a bit different depending on whether you are using a HEAT Orchestration Template (HOT), which is essentially a virtual machine descriptor, or whether you are specifying the details when launching the vTA. See below for instructions on how to launch the vTA:

- manually – section 3.3.1;
- using HOT automation – section 3.3.2;
- using the Python OpenStack API – section 3.3.3.

Once you have launched the vTA image, the vTA will register with Netrounds Control Center and appear in its web GUI under Test Agents. Check for the vTA name in that view to verify that the vTA has registered. You can then initiate tests and monitoring sessions on the vTA from Netrounds Control Center.

3.3.1 Launching a vTA Image Manually in Horizon

Here is how to deploy a vTA manually, without using a HEAT Orchestration Template.

- On the navigation bar, select Project → Compute → Instances. Then click Launch instance.
- Under Details, enter a name for the vTA instance.
- Under Source, select the image to boot the vTA instance from. In general, the default volume settings can be kept.
- Under Flavor, select a suitable flavor for the vTA. Please refer back to section 3.2.
- Under Networks, select whatever is most appropriate in your case.
- The Network Ports section can be skipped.
- Under Security Groups, the group chosen must satisfy the following:
  - The vTA must be able to establish an outgoing session towards its Netrounds Control Center. In the cloud server case, to login.netrounds.com using TCP port 443; in the on-premise server case, to the host IP using TCP port 6000 (default).
  - UDP port 123 must be open to permit NTP time sync.
  - Traffic must be allowed on all ports needed for the testing you intend to do with the vTA.
- The Key Pair section can be skipped, as SSH is not in use for vTAs.
- Under Configuration, in the Customization Script box, "cloud-config" metadata must be entered specifying among other things how to connect to Netrounds Control Center. The format of this metadata is given in section 3.3.1.1 below.
- The Server Groups, Scheduler Hints, and Metadata sections can be skipped.
- Finish by clicking the Launch Instance button at the bottom of the dialog.

A vTA image can also be deployed from the command line, but that method is not explained further in this introductory guide.
### 3.3.1.1 Format of "cloud-config" Metadata

The vTA supports only `#cloud-config` metadata. Its format is as shown below. **Note** that the `#cloud-config` and `netrounds_test_agent` lines must be present, and that all of the remaining lines must be indented.

**Basic configuration:**

```
#cloud-config
netrounds_test_agent:
    name: <vTA name>
    email: <Netrounds user email address>
    password: <Netrounds password>
    account: <Netrounds account name>
    server: <Netrounds server> (default: login.netrounds.com:443)
    management_interface: eth1 (default: eth0)
    management_address_type: dhcp | static (default: dhcp)
```

The following parameters are required only if `management_address_type` is "static":

```
management_ip: <management IP address>/<prefix>
management_dns: <DNS server IP address>[,<DNS server IP address>]
management_default_gateway: <gateway IP address>
management_ntp: <NTP server IP address or host name> (default: ntp.netrounds.com)
```

The following parameters are required only if `vTA` is connecting to the server through an HTTP proxy:

```
http_proxy: <proxy server>
http_proxy_port: <proxy port>
http_proxy_auth_type: none | basic (default: none)
```

The following parameters are required only if `http_proxy_auth_type` is "basic":

```
http_proxy_username: <proxy authorization user name>
http_proxy_password: <proxy authorization password>
```

### 3.3.2 Launching a vTA Image Using a Heat Orchestration Template in Horizon

To deploy a vTA image in OpenStack using a Heat Orchestration Template (HOT):

- On the navigation bar, select **Project** → **Orchestration** → **Stacks**.
- Click the **Launch Stack** button on the far right. Select the HOT file to be used. For an example of such a file, see section 3.3.2.1.

![Select Template](image)

The template will ask for input to be specified, as shown in the following screenshot:
Some of these fields have default values, as defined in the HOT file, while others are variables in the template.

After a vTA image has been booted up, it must receive its instance-specific userdata/metadata so that it can properly configure the network interfaces and call back home to Netrounds Control Center. Netrounds vTA supports two ways to provide this instance-specific data:

1. Requesting the data from a service running at http://169.254.269.254 on the compute host. This is similar to the approach taken by an EC2 instance running in Amazon WS.

2. Reading the data from a special configuration drive that is attached to the vTA instance when it boots. This is referred to as the “config-drive method”.

The format of the instance-specific userdata/metadata is given in section 3.3.1.1.

Using either of the above methods, it is possible to automate connecting the vTA to Netrounds Control Center.

**Note:** You need to ensure that the vTA can establish an outgoing session towards its Netrounds Control Center: in the cloud server case, to login.netrounds.com using TCP port 443; in the on-premise server case, to the host IP using TCP port 6000 (default). Alternatively, the vTA may connect via an HTTP proxy. In addition, UDP port 123 needs to be open to permit NTP time sync.

### 3.3.2.1 Example of HOT File

An example of a HOT file is provided below. Replace the red text in square brackets with your input.

```yaml
heat_template_version: 2015-07-15

description: Heat template to deploy a single vTA in OpenStack

parameters:
  account_name:
    type: string
    description: Netrounds account name to be used for autoregistration
  email:
    type: string
    description: User name (email address) for Netrounds account
  account_password:
    type: string
    hidden: true
    description: Netrounds account PW to be used for autoreg
```
3.3.3 Launching a vTA Image Using the Python OpenStack API

It is possible to use the OpenStack Nova and Keystone Python APIs to automatically launch the vTA image from a Python script:

```python
def create_instance(net, vta_name, os_password):
    # connect using credentials
    creds = get_nova_creds(os_password)
    nova = nvclient.Client(**creds)

    # get image, flavor, network/nics
    image = nova.images.find(name="vTA_cloudinit_image")
    flavor = nova.flavors.find(name="netrounds.small.2GB")
    network = nova.networks.find(label=net)
    nics = [{"netid": network.id}]

    # create instances, requires cloudinit on vTA images
    instance = nova.servers.create(name=vta_name, image=image,
                                   flavor=flavor, key_name="default", nics=nics,
                                   user_data=open("./userdata.txt"))
```
The cloud-init file (user-data) in "userdata.txt" above has this content:

```plaintext
#cloud-config
netrounds_test_agent:
  email: [email you use when logging in to Netrounds]
  password: [Netrounds login password]
  account: [Netrounds account]
  server: [Netrounds Control Center address; for SaaS: login.netrounds.com:443]
  name: [Name of vTA to appear in Netrounds Control Center inventory]
```

Note again that the #cloud-config and netrounds_test_agent lines must be present.

Please contact the Netrounds support team at support@netrounds.com to get example Python code.
4 Appendix: Description of the vTA VNF and Its Requirements

1. The vTA VNF is capable of running in a plain, "vanilla" environment using a standard cloud configuration and orchestration based on OpenStack. There might be some limitations in terms of performance and also some minor limitations in terms of jitter and delay accuracy, depending on your OpenStack infrastructure and how heavily loaded it is. However, for early proof-of-concepts and evaluations, this should not be a major issue. To obtain line rate packet generation and optimal usage of your specific hypervisor environment, an integration project would be required.

2. The vTA VNF consists of a single stand-alone VNF. However, the VNF must be able to connect and communicate securely with Netrounds Control Center, which is not a VNF. Netrounds Control Center is readily available in the public cloud (in addition to private cloud installations), something which simplifies test and evaluation projects.

3. Interfaces trust the natural OS bootstrap order in terms of how they are identified.

4. The performance is dependent on the underlying hardware. The more powerful the hardware, the higher the performance. For a 3 GHz quad-core processor, achievable performance is up to 10 Gbit/s using five concurrent unidirectional TCP streams.

5. The minimum recommended specification is: 1 vCPU, 512 MB RAM, and 2 GB of block storage.

6. It is assumed that a generic VNF manager which is not part of the Netrounds solution does the instantiation, scaling, and termination of the vTA VNF.

7. The vTA VNF needs to register with Netrounds Control Center to receive commands. For public cloud Netrounds Control Center scenarios, the VNF needs connectivity to the Internet from the eth0 interface. For plug-and-play configuration of the VNF, DHCP should be used for IP addressing of the vTA’s interfaces, as well as for assignment of an available DNS server.

8. The VNF will resolve the Netrounds Control Center address and initiate an outbound connection using TCP. To successfully connect and authenticate itself to the correct Netrounds account, the VNF needs to have credentials provided to it during initialization. The VNF supports cloud-init and configdrive for this purpose for its day-zero configuration. Once the VNF has connected to Netrounds Control Center, it can be controlled either via a web browser or through the Netrounds cloud API to start monitoring user experience KPIs, conduct a service turn-up test, or perform on-demand troubleshooting tests. The connection is an encrypted OpenVPN connection.

9. The vTA VNF also requires synchronization to an NTP server in order to achieve accurate delay and jitter measurements. By default, Test Agents will synchronize their internal clock to an NTP server provided by Netrounds (ntp.netrounds.com); however, any NTP server (internal or external) can be used.

10. Rescaling of the VNF again needs to be handled by a generic VNF manager (compare paragraph 6). For example, if the available connection bandwidth is increased, the VNF might need to be scaled up to be able to push enough bandwidth through the link for testing purposes.