How to Deploy an OVA Virtual Test Agent Image in VMware

Executive Summary
This guide explains how to start a Netrounds Virtual Test Agent as a vApp on a VMware virtual machine.

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 Uploading and Deploying a vTA Image .............................................................................................................................3
   3.1 Uploading and Deploying a vTA Image via vSphere Client........................................................................................3
      3.1.1 Powering On the vTA.............................................................................................................................................7
   3.2 Uploading and Deploying a vTA Image with OVF Tool..............................................................................................7
4 Troubleshooting.................................................................................................................................................................8
5 Appendix: Description of the vTA VNF and Its Requirements........................................................................................11
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 virtualized environment. These vTAs will automatically connect to the 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 VMware is provided in OVA (OVF/VMDK) format and is packaged using the OVF Tool which uses SHA1 checksum. The OVF file specifies version VMX-09, since that is the lowest version which has the required functionality.

The OVF file also specifies 512 MB RAM and 2 GB block storage for the vTA.
3  **Uploading and Deploying a vTA Image**

Once you have your vTA image, you need to upload it to your VMware environment and deploy it. This can be done either via the VMware vSphere Client or with the OVF Tool.

The supplying of Netrounds of user data is done in the process of this deployment.

3.1  **Uploading and Deploying a vTA Image via vSphere Client**

This is possible only in Windows and iOS. If you are using a different operating system, you need to use the method in section 3.2 instead.

- Log in to vSphere Client.
- In vSphere Client, navigate to [vCenter Inventory Lists](#).

  ![vCenter Inventory Lists](image)

- Select [vApps](#).

  ![vApps](image)

- Click the [Deploy OVF template](#) button (circled in the screenshot below).

© 2019 Netrounds. All rights reserved.
• In the wizard that appears, select **Local file** and browse to select your OVA/OVF file. Then click **Next**.

• On the **Review details** screen, click **Next**.

• On the **Select name and folder** screen, **Name** is predefined as “netrounds”. Select a **folder or datacenter** as exemplified in the screenshot below. Then click **Next**.
• On the **Select a resource** screen, select where to run the deployed template. Click **Next**.

• On the **Select storage** screen, the settings can be left as-is. Select a datastore in which to store the files for the deployed template. Click **Next**.

• On the **Setup networks** screen, edit the configuration if necessary; otherwise, no action is required here. Continue by clicking **Next**.
• On the **Customize template** screen, you need to fill in your Netrounds user data as exemplified below (you will need to modify the details according to your setup). Then click **Next**.

• Finally, on the **Ready to complete** screen, review your settings. Then click **Finish**.

The OVF template will now be deployed in VMware. This will take a non-trivial amount of time; the progress of the deployment is shown in the **Recent Tasks** pane in vSphere Web Client:

---

© 2019 Netrounds. All rights reserved.
3.1.1 Powering On the vTA

For the vTA to come online, you must power it on. In vSphere Client, do as follows:

- From the Home screen in vSphere Client, navigate to vCenter Inventory Lists → vApps, and select your vApp (by default named “netrounds”).

- Click the Power on vApp link to power on the “netrounds” vApp.

This powers on the NetroundsVTA virtual machine as well.

The vTA will now 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.

3.2 Uploading and Deploying a vTA Image with OVF Tool

With OVF Tool the procedure for vTA deployment is as follows:

- You first need to configure vTA user data in the OVF file. To this end, uncompress the OVA file, which among other things contains the OVF.
• Open the OVF file in your text editor of choice and scroll down to the `ovf:ProductSection` tag:

```xml
<ovf:ProductSection ovf:class="" ovf:instance="" ovf:required="true">
  <ovf:Info>Information about the installed software</ovf:Info>
  <ovf:Category>Netrounds Control Center registration</ovf:Category>
...
```

• Inside that tag you find several `ovf:Property` tags, each of which controls one data entry, identified by its `ovf:key`. To change the value of a Property, edit its `ovf:value`:

```xml
...<ovf:Property ovf:key="netrounds.http_proxy" ovf:password="false" ovf:type="string" ovf:userConfigurable="true" ovf:value=""/>
  <ovf:Label>Test Agent HTTP-Proxy server</ovf:Label>
  <ovf:Description>The address to a HTTP-Proxy. This is optional</ovf:Description>
</ovf:Property>
...```

• When done configuring vTA user data, compress the files back into an OVA again (use the tar format and then replace the file extension).

• You are now ready to upload your vTA image with OVF Tool. The vTA image also needs to be powered on in the same command. Use this syntax:

```
$ ovftool --acceptAllEulas "--datastore=DATASTORE" "--network=NETWORK" --powerOn
file.ova vi://USER:PASSWORD@SERVER/DATACENTER/host/HOST
```

Here, each CAPITALIZED word should be replaced with the appropriate value, and `file.ova` is the name of your OVA file.

Example:

```
$ ovftool --acceptAllEulas "--datastore=datastore1" "--network=VM Network" --powerOn
vTA.ova vi://admin@netrounds.vsphere:mypassword@vcenter.lulea.netrounds.local/Datacenter/host/esxi.lulea.netrounds.local
```

4 Troubleshooting

If the vTA does not show up in Netrounds Control Center, it is useful to open its local console to investigate the cause of the problem.

• From the Home screen in vSphere Client, navigate to vCenter Inventory Lists → vApps, and select your vApp (by default named “netrounds”).

• Click the “netrounds” vApp once more in the left-hand navigation pane, then click Top Level Objects.

• Click the “NetroundsVTA” object.
• In the right-hand pane, at the top, open the Actions menu and select Open Console.

• The console opens on a new tab. If you do not see the prompt shown below, click the Send Ctrl+Alt+Delete button in the top right corner.

• Log in using the credentials indicated. You are taken to the top-level Probe Admin Menu:
The functionality here is described in the Netrounds support documentation under **Test Agents → Configuring Test Agents from the local console**. The following functions are particularly helpful:

- **Utilities → Ping** for checking that the vTA has a working internet connection.
- **Utilities → Troubleshoot connection** for verifying that the Netrounds management connection is working.
5 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 VMware. There might be some limitations in terms of performance and also some minor limitations in terms of jitter and delay accuracy, depending on your VMware 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. The latter two settings are in the OVF file, as mentioned in section 2.2.

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. (For details, see the Netrounds Control Center support documentation.) To successfully connect and authenticate itself to the correct Netrounds account, the VNF needs to have credentials provided to it during initialization. In the VMware environment, these credentials are entered in the vSphere Client (see section 3.1) or in the OVF file (see section 3.2). 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.