How to Deploy a VHD Virtual Test Agent Image in Azure

Executive Summary
This guide explains how to deploy a Netrounds Virtual Test Agent as a virtual machine in Microsoft Azure.

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 Through the Azure Web GUI.............................................3
   3.1 Signing In to Azure......................................................................................................................3
   3.2 Creating a Storage Account........................................................................................................3
   3.3 Creating a Storage Container (Blob)..........................................................................................4
   3.4 Uploading the Test Agent VHD File to the Storage Container..................................................5
   3.5 Creating an Image......................................................................................................................5
   3.6 Creating a Virtual Machine.........................................................................................................6
   3.7 Logging In to the Test Agent......................................................................................................7
4 Uploading and Deploying a vTA Image Through the Azure CLI....................................................8
   4.1 Creating a Storage Account........................................................................................................8
   4.2 Creating a Storage Container (Blob)..........................................................................................9
   4.3 Uploading the Test Agent VHD File to the Storage Container..................................................9
   4.4 Creating a Virtual Machine.......................................................................................................9
5 Appendix: Description of the vTA VNF and Its Requirements......................................................10
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:

![Netrounds Control Center Diagram](image)

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 vTA image is provided either directly from Netrounds or from one of Netrounds’ partners.

The vTA image for Azure is provided in VHD format.

Once you have your vTA image, you need to upload it to your Azure environment and deploy it. This can be done either through the Azure web GUI or from the Azure CLI. Both procedures are described in this document: see chapters 3 (web GUI) and 4 (CLI). If you want to orchestrate Test Agent tasks using cloud-init, you need to use the CLI.
3 Uploading and Deploying a vTA Image Through the Azure Web GUI

3.1 Signing In to Azure

- Go to [https://azure.microsoft.com](https://azure.microsoft.com). You will be redirected to a URL associated with your location. The description that follows deals with the English-language version of the GUI.
- Sign in to your Azure account.
- You are taken to a user interface that looks like this:

![Microsoft Azure Dashboard](image)

3.2 Creating a Storage Account

- In the left-hand pane, click Storage accounts. This opens a view listing your existing storage accounts, if any.

![Storage accounts](image)

- Click “+ Add” to create a new storage account.
- Make the appropriate choice under Resource group (we are assuming here that some resource group has already been created). The remaining settings can be left as-is.
Create storage account

Azuré Storage is a Microsoft-managed service providing cloud storage that is highly available, secure, durable, and highly scalable. Azure Storage includes Azure Blobs (objects), Azure Data Lake Storage Gen2, Azure Files, Azure Queues, and your storage account depends on the usage and the options you choose below. Learn more

PROJECT DETAILS
Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize resources.

* Subscription
  Pay-As-You-Go

* Resource group
  Create new  Use existing

- Click the **Review + create** button at the bottom.
- Click **Create**.

Your storage account is now created. Refresh the Storage accounts page if necessary to see your account in the list.

### 3.3 Creating a Storage Container (Blob)

- Click your account in the **Storage accounts** list.
- Click **Blobs**.

![Blobs]

REST-based object storage for unstructured data

Learn more

- Click “+ Container”.
- Enter a name for the container.
- For the other settings, the defaults can be kept here as well.
- Click **OK**. Your container is now created.
3.4 Uploading the Test Agent VHD File to the Storage Container

The next step is to upload your Test Agent VHD file to the storage container you just created.

- Click the storage container in the list of containers.
- Click **Upload**.

  ![Upload Refresh](image)

- Under **Files**, select your Test Agent VHD file.
- Expand **Advanced**.

  ![Advanced](image)

  - **Important:** Under **Blob type**, select **Page blob**.

    ![Blob type](image)

    - Keep the defaults for the remaining settings.
    - Click **Upload**.

The upload will take some time as the Test Agent VHD file is approximately 2 GB in size.

3.5 Creating an Image

- In the left-hand pane, click **Images**.

  ![Help + support Images](image)

- Click **+ Add**.

  ![Create image](image)

  - Enter a name for the image.
  - Make the appropriate choice under **Resource group**.
  - Under **OS type**, select Linux.
• Under Storage blob, browse to select the VHD file you uploaded in section 3.4.
• Leave the remaining settings unchanged.
• Click Create. The image will now appear in the Images view.

3.6 Creating a Virtual Machine
In this section we will create a virtual machine (VM) in which to run the vTA.
• In the Images view, click the image you created.
• Click Create VM.
• Under **Project details**, make the appropriate selections.

• Under **Instance details**, do the following:
  
  • Enter a name for the virtual machine.
  
  • Under **Size**, select “B1ms” (1 vCPU, 2 GB RAM; compare chapter 5, section 5).

• Under **Administrator account**, you need to provide an SSH key pair to be able to log in to the Test Agent admin menu later on:
  
  • Set **Authentication type** to “SSH public key”.
  
  • Enter an arbitrary string under **Username**. This setting cannot be left undefined, but it is not used when logging in to the Test Agent.

  • Under **SSH public key**, paste your SSH public key.

• Under **Inbound port rules**, do as follows:
  
  • Set **Public inbound ports** to “Allow selected ports”.
  
  • Under **Select inbound ports**, select “SSH (22)”.

• Keep the defaults for all other settings.

• Click **Review + create**, then **Create**.

  The virtual machine is now created.

### 3.7  Logging in to the Test Agent

Here is how to log in to the Test Agent via SSH. This is needed in order to register the Test Agent with the Netrounds system, and it is also useful for troubleshooting.

• In the left-hand pane, select **Virtual machines**.

• Select the virtual machine created for the Test Agent.

• Note down the **Public IP address** of the virtual machine.

  ![Public IP address](image)

• At a command prompt, type:

```sh
ssh -i <id_rsa> admin@<vm_public_ip>
```

  where **id_rsa** is the name of the file holding your SSH private key and **vm_public_ip** is the virtual machine’s public IP address.

You are now taken to the Test Agent admin menu:
Here you can register the Test Agent with the Netrounds system as described in the Netrounds support documentation under Test Agents → Configuring Test Agents from the local console → Registering a Test Agent from the local console. Upon registration, the Test Agent will be visible in Netrounds Control Center.

Again, please note that initialization of the Test Agent with user data using cloud-init cannot be done through the web GUI. The Azure CLI must be used for this purpose; see section 4.4.

The other functionality found here is likewise described in the Netrounds support documentation: see the remaining topics 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.

### 4 Uploading and Deploying a vTA Image Through the Azure CLI

A different way to create and manage Azure resources is through the Azure CLI. In this chapter we indicate how to use the CLI to perform the operations done via the web GUI in chapter 3.

Full documentation of the Azure CLI is found here: [https://docs.microsoft.com/en-us/cli/azure](https://docs.microsoft.com/en-us/cli/azure)

#### 4.1 Creating a Storage Account

Here is shown:

- how to create a resource group (this is assumed to exist in the web GUI in chapter 3)
- how to create a storage account within the resource group
- how to create access keys. Access keys are used to authenticate applications when they make requests to the Azure storage account. They are needed for some of the operations that follow.

```
# Create resource group
az group create --location northeurope --name netrounds

# Create storage account
az storage account create --name netroundsstorage --resource-group netrounds --location northeurope
```
export AZURE_STORAGE_ACCOUNT=netroundsstorage

# Get access key
az storage account keys list --resource-group netrounds --account-name netroundsstorage
-o table
export AZURE_STORAGE_KEY=<one of the keys from the above command>

### 4.2 Creating a Storage Container (Blob)

```bash
# Create storage container
az storage container create --name netroundscontainer --account-name netroundsstorage
--account-key AZURE_STORAGE_KEY
```

### 4.3 Uploading the Test Agent VHD File to the Storage Container

The VHD file you have downloaded from Netrounds Control Center is named `netrounds-test-agent_<version number>.vhd`. This is provided as the `--file` argument. The `--name` argument specifies what the VHD file is to be called in Azure.

```bash
# Upload VHD
az storage blob upload --container-name netroundscontainer --file netrounds-test-agent_<version number>.vhd --name test-agent.vhd --type page --account-name netroundsstorage --account-key AZURE_STORAGE_KEY
```

### 4.4 Creating a Virtual Machine

When creating a virtual machine for running the vTA, you need to use the `--admin-username` option to specify an admin user and the `--ssh-key-value` option to supply your public SSH key in a file (assumed to be named `id_rsa.pub` below).

```bash
# Create VM
az vm create --resource-group netroundsstuff --name netroundsvta --os-type Linux
--image https://netroundsstorage.blob.core.windows.net/netroundscontainer/test-agent.vhd
--use-unmanaged-disk --storage-account netroundsstorage --boot-diagnostics-storage netroundsstorage --custom-data user-data.yaml --admin-username <user-name> --ssh-key-value id_rsa.pub
```

The option `--custom-data` is used to initialize the Test Agent with a Netrounds cloud-init config in a YAML file (`userdata.yaml`). Note that this cannot be done through the web GUI. The YAML file has the following format:

```yaml
#cloud-config
netrounds_test_agent:
  name: MyTAA
  email: myuser@email.com
  password: mypassword
  account: myaccount
```

An additional line `server:` can be included in the YAML file to specify a server different from the Netrounds SaaS server (which is the default).

Provided that correct credentials are given here, the vTA will register automatically with the Netrounds system and appear in the list of Test Agents in the Netrounds Control Center GUI.
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 Azure. There might be some limitations in terms of performance and also some minor limitations in terms of jitter and delay accuracy, depending on your Azure 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: 1vCPU, 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. (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 Azure environment, these credentials can be entered in a YAML file supplied via the command line interface (see section 4.4). 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.