

How to Deploy a VHD Virtual Test Agent Image in Azure

Published
2021-11-01

RELEASE
3.0.0

Table of Contents

[Executive Summary](#)

[Paragon Active Assurance: Solution Overview](#)

[Prerequisites](#)

[Uploading and Deploying a vTA Image Through the Azure Web GUI](#)

[Uploading and Deploying a vTA Image Through the Azure CLI](#)

[Appendix: Description of the vTA VNF and Its Requirements](#)

Executive Summary

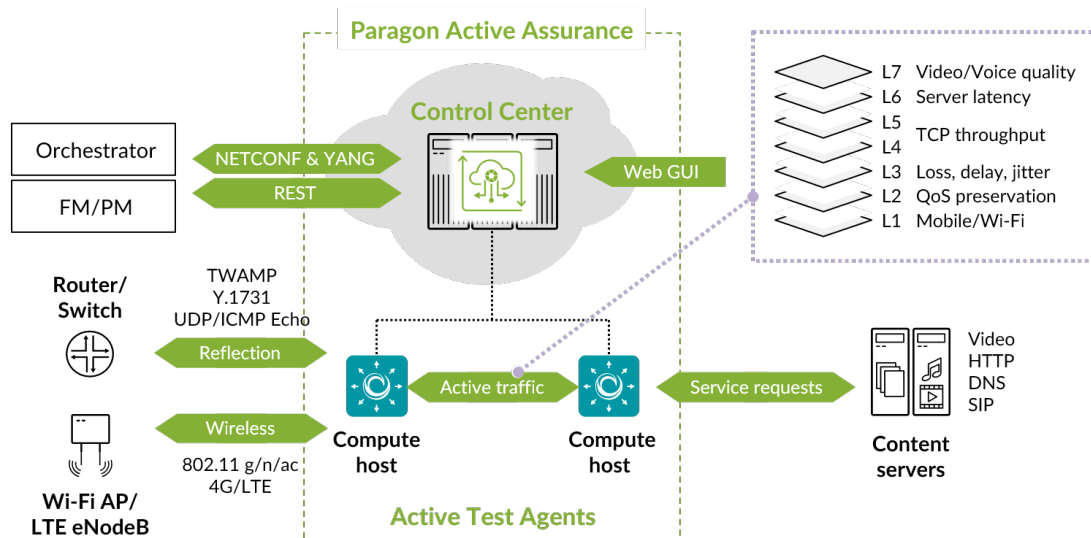
This guide explains how to deploy a Virtual Test Agent from Paragon Active Assurance as a virtual machine in Microsoft Azure.

Paragon Active Assurance: Solution Overview

Paragon Active Assurance 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 Control Center as part of the deployment process described in this guide. (Juniper Networks also offers non-virtual Test Agents in the form of software that is installed on stand-alone x86 hardware.)
2. **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.

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



Prerequisites

IN THIS SECTION

- [Control Center Account | 2](#)
- [vTA Image | 2](#)

Control Center Account

You need an account in a Control Center in order to access it: either the one belonging to the SaaS solution or one installed on-premise in your organization. If you do not already have a Paragon Active Assurance account, please contact your Juniper partner or your local Juniper account manager or sales representative.

vTA Image

The VNF vTA image is provided either by one of Juniper's partners or directly by Juniper.

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 the chapters

- ["Uploading and Deploying a vTA Image Through the Azure Web GUI" on page 3](#)
- ["Uploading and Deploying a vTA Image Through the Azure CLI" on page 10](#)

If you want to orchestrate Test Agent tasks using cloud-init, you need to use the CLI.

The vTA image for VMware is provided in OVA (OVF/VMDK) format and is packaged using the OVF Tool which uses a 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.

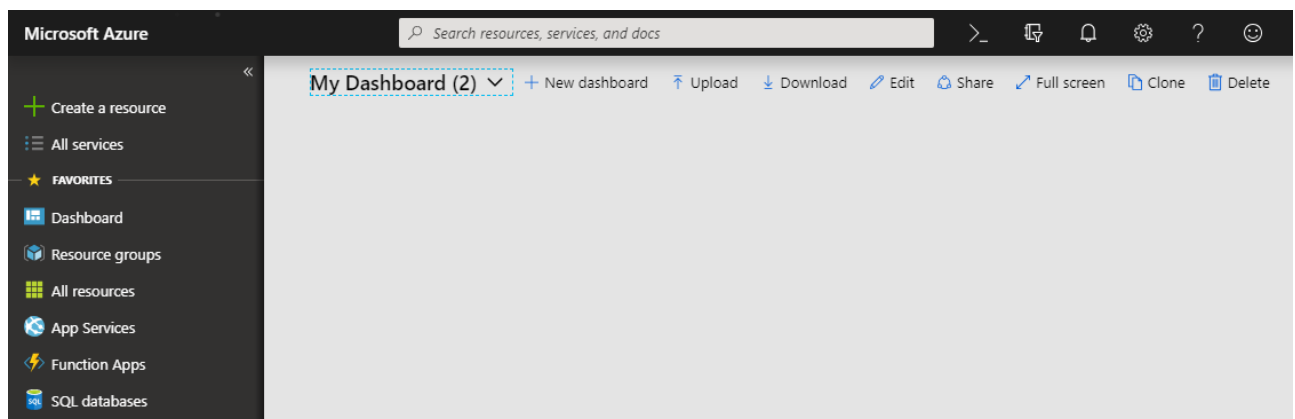
Uploading and Deploying a vTA Image Through the Azure Web GUI

IN THIS SECTION

- [Signing In to Azure | 3](#)
- [Creating a Storage Account | 4](#)
- [Creating a Storage Container \(Blob\) | 5](#)
- [Uploading the Test Agent VHD File to the Storage Container | 5](#)
- [Creating an Image | 6](#)
- [Creating a Virtual Machine | 7](#)
- [Logging In to the Test Agent | 9](#)

Signing In to Azure

- Go to <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:



Creating a Storage Account

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

Home > Storage accounts

Storage accounts
Default Directory

+ Add Edit columns Refresh Assign tags Delete

Subscriptions: Pay-As-You-Go

Filter by name... All resource groups All types All locations All tags No grouping

2 items

	NAME ↑↓	TYPE ↑↓	KIND ↑↓	RESOURCE GROUP ↑↓	LOCATION ↑↓	SUBSCRIPTION ↑↓	
<input type="checkbox"/>	netrounds2	Storage account	StorageV2	p	North Europe	Pay-As-You-Go	...
<input type="checkbox"/>	netroundsci	Storage account	Storage	q	North Europe	Pay-As-You-Go	...

- 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

Basics Advanced Tags Review + create

Azure Storage is a Microsoft-managed service providing cloud storage that is highly available, secure, durable, and easy to use. Azure Storage includes Azure Blobs (objects), Azure Data Lake Storage Gen2, Azure Files, Azure Queues, and Azure Durable Functions. 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

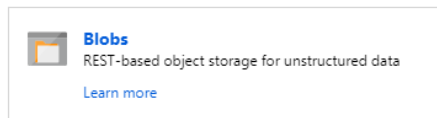
- Click the **Review + create** button at the bottom.
- Click **Create**.

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

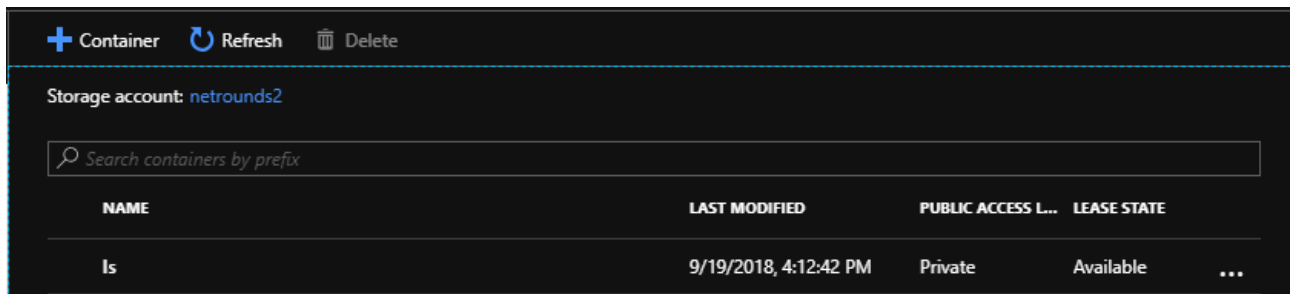
Creating a Storage Container (Blob)

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

Services



- 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.



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.

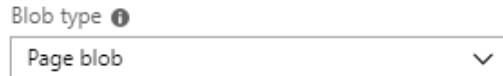
- Click the storage container in the list of containers.
- Click **Upload**.



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



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

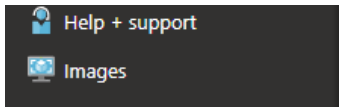


- Keep the defaults for the remaining settings.
- Click **Upload**.

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

Creating an Image

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



- Click "+ Add".

 A dark-themed form titled "Create image" in white. It contains three required fields, each marked with a red asterisk: "Name" with an empty text input box, "Subscription" with a dropdown menu showing "Pay-As-You-Go", and "Resource group" with an empty dropdown menu.

- 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 the section ["Uploading the Test Agent VHD File to the Storage Container" on page 5](#).

- Leave the remaining settings unchanged.
- Click **Create**. The image will now appear in the **Images** view.

Home > Images

Images


Default Directory

+ Add Edit columns Refresh Assign tags

Subscriptions: Pay-As-You-Go

Filter by name... All resource groups All locations All tags

1 items

<input type="checkbox"/>	NAME ↑↓	SOURCE VIRTUA...	OS TYPE	RESOURCE GROUP ↑↓	LOCATION ↑↓	ZONE RE
<input type="checkbox"/>	 netrounds-test-agent_2.26.0-dev....	-	Linux	q	North Europe	No

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**.

Create a virtual machine

[Basics](#) [Disks](#) [Networking](#) [Management](#) [Guest config](#) [Tags](#) [Review + create](#)

Create a virtual machine that runs Linux or Windows. Select an image from Azure marketplace or use your own customized image. Complete the Basics tab then Review + create to provision a virtual machine with default parameters or review each tab for full customization.
Looking for classic VMs? [Create VM from Azure Marketplace](#)

PROJECT DETAILS

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

* Subscription ⓘ

* Resource group ⓘ [Create new](#)

INSTANCE DETAILS

* Virtual machine name ⓘ ✓

* Region ⓘ

Availability options ⓘ

* Image ⓘ [Browse all images and disks](#)

* Size ⓘ **Standard B1ms**
1 vcpu, 2 GB memory
[Change size](#)

- 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 ["Appendix: Description of the vTA VNF and Its Requirements" on page 13](#), paragraph 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:

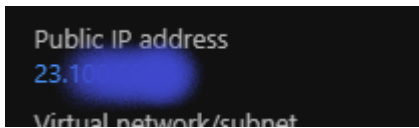
- Set **Public inbound ports** to “Allow selected ports”.
- Under **Select inbound ports**, select “SSH (22)”.
- Keep the defaults for all other settings.
- Click **Review + create**, then **Create**.

The virtual machine is now created.

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 Paragon Active Assurance 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.

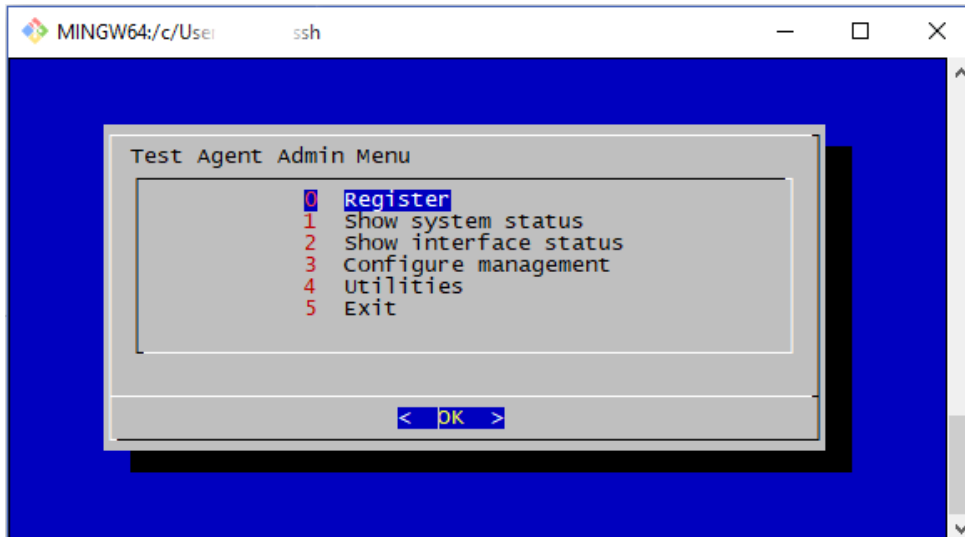


- At a command prompt, type:

```
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 Paragon Active Assurance system as described in the 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 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 the section ["Creating a Virtual Machine" on page 12](#).

The other functionality found here is likewise described in the Paragon Active Assurance 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 Paragon Active Assurance management connection is working.

Uploading and Deploying a vTA Image Through the Azure CLI

IN THIS SECTION

- [Creating a Storage Account | 11](#)

- Creating a Storage Container (Blob) | 12
- Uploading the Test Agent VHD File to the Storage Container | 12
- Creating a Virtual Machine | 12

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 the chapter "[Uploading and Deploying a vTA Image Through the Azure Web GUI](#)" on page 3.

Full documentation of the Azure CLI is found here: <https://docs.microsoft.com/en-us/cli/azure>

Creating a Storage Account

Here is shown:

- how to create a resource group (this is assumed to exist in the web GUI in the chapter "[Uploading and Deploying a vTA Image Through the Azure Web GUI](#)" on page 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>
```

Creating a Storage Container (Blob)

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

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.

```
# 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
```

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).

```
# 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 Paragon Active Assurance 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:

```
#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 Paragon Active Assurance SaaS server (which is the default).

Provided that correct credentials are given here, the vTA will register automatically with the Paragon Active Assurance system and appear in the list of Test Agents in the Control Center GUI.

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 Paragon Active Assurance Control Center, which is not a VNF. 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 Paragon Active Assurance solution does the instantiation, scaling, and termination of the vTA VNF.
7. The vTA VNF needs to register with Control Center to receive commands. For public cloud 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 Control Center address and initiate an outbound connection using TCP. (For details, see the Paragon Active Assurance support documentation.) To successfully connect and authenticate itself to the correct Paragon Active Assurance 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 "[Creating a Virtual Machine](#)" on page 12). Once the VNF has connected to 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](#)" on page 14). 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.