

How to Deploy an AMI Test Agent in AWS

Published
2021-01-14

Table of Contents

[Executive Summary](#)

[Paragon Active Assurance: Solution Overview](#)

[Prerequisites](#)

[Launching an AWS Instance](#)

[Verifying Successful Test Agent Configuration](#)

[Troubleshooting](#)

Executive Summary

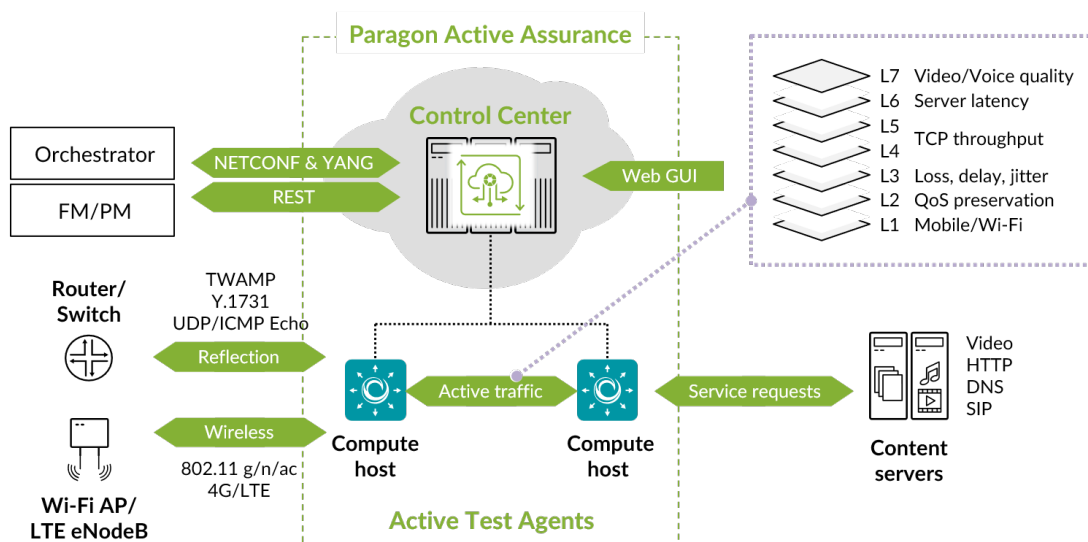
This guide explains how to deploy a Paragon Active Assurance Test Agent in Amazon EC2 (Elastic Compute Cloud) by launching an AWS instance on which to run a Paragon Active Assurance AMI (Amazon Machine Image).

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 virtualized 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](#)
- [Paragon Active Assurance AMI for Test Agent | 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.

Paragon Active Assurance AMI for Test Agent

An Amazon Machine Image (AMI) is a special type of virtual appliance used to create a virtual machine within the Amazon Elastic Compute Cloud ("EC2"), which is part of Amazon Web Services. The AMI serves as the basic unit of deployment for services delivered using EC2.

A public AMI for Paragon Active Assurance is available in AWS Marketplace. The chapter "[Launching an AWS Instance](#)" on page 2 tells how to obtain and configure it.

Launching an AWS Instance

IN THIS SECTION

- [Logging In to Amazon EC2 | 3](#)

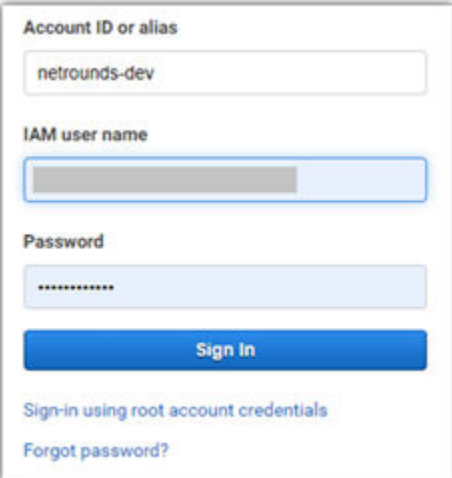
- Choosing an AMI | 4
- Choosing an AWS Instance Type | 4
- Configuring AWS Instance Details | 4
- Selecting Storage | 6
- Adding Tags | 6
- Configuring Security Group | 6
- Reviewing Your Instance Settings and Selecting an SSH Key Pair | 7

This chapter tells how to launch an AWS instance on which to run the Paragon Active Assurance AMI.

Be aware that the AMI is shared to a specific geographical *region* within EC2. Therefore you need to know what region that is and make sure you access the same region.

Logging In to Amazon EC2

- Go to <https://aws.amazon.com/ec2>.
- Click the button **Get started with Amazon EC2**.
- Sign in to your AWS account:

A screenshot of the AWS sign-in interface. It features three input fields: 'Account ID or alias' with the text 'netrounds-dev', 'IAM user name' with a greyed-out field, and 'Password' with masked characters. Below these fields is a blue 'Sign In' button. At the bottom, there are two links: 'Sign-in using root account credentials' and 'Forgot password?'.

Account ID or alias

netrounds-dev

IAM user name

Password

Sign In

Sign-in using root account credentials

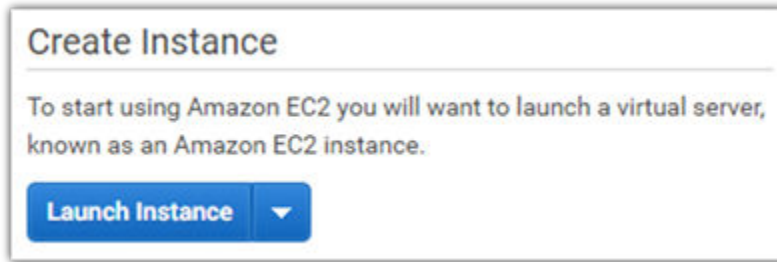
Forgot password?

- Click **Services** on the top bar.

- In the **Compute** section, click **EC2**. You are taken to the EC2 Management Console.

Choosing an AMI

- Under the heading **Create Instance**, click the **Launch Instance** button.



- In the left-hand menu, click **AWS Marketplace**.
- In the search field, enter "paragon". You will find an AMI called "Paragon Active Assurance Test Agent".
- Click the **Select** button next to the Test Agent AMI.
- On the screen that appears, click **Continue**. You are now taken to the next step.

Choosing an AWS Instance Type

A large number of AWS instance types will typically appear in this list. Which one to choose depends on the performance needed when running the AMI. We recommend an Amazon EC2 C5 instance for the Test Agent.

- Select an AWS instance type, then click the button **Next: Configure Instance Details** at the bottom of the page.

Configuring AWS Instance Details

The default settings can be kept here. However, it is highly recommended that you also enter the cloud-init config for the vTA as user data, as explained below. Alternatively, you can configure this after launching the

instance by connecting to the vTA via SSH and navigating the vTA console interface (see chapter ["Troubleshooting" on page 9](#)).

- Expand the **Advanced Details** section at the bottom of the page.
- Under **User data**, provide the cloud-init config for the vTA, either by pasting it into the box (**As text** option) or by browsing to a file (**As file** option).

The basic cloud-init config is as shown below. Text in angle brackets < > needs to be replaced by the proper strings. Note that lines with parameter settings must be indented as shown. Lines where the default value is kept can be omitted.

```
#cloud-config
netrounds_test_agent:
  name: <vTA name>
  email: <Paragon Active Assurance user email address>
  password: <Paragon Active Assurance password>
  account: <Paragon Active Assurance account name>
  server: <Paragon Active Assurance 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 the 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>
```

- The remaining settings can be left as-is.
- Once you have entered your cloud-init config data, click the **Next: Add Storage** button.

Selecting Storage

The recommendation here is at least 2 GB of storage.

- Select a suitable storage device, then click the **Next: Add Tags** button.

Adding Tags

This step is optional. There is no need to add any tags for the AMI Test Agent.

- Click the **Next: Configure Security Group** button.

Configuring Security Group

The security group selected here must allow outgoing traffic on ports that the vTA needs in order to communicate with Control Center. Specifically, for SaaS, TCP port 443; for an on-premise installation, TCP port 6000. In addition, UDP port 123 needs to be open to permit NTP time sync.

The security group must also allow traffic on all ports needed for the testing you intend to do with the vTA.

- After selecting a security group, click the **Review and Launch** button.

Reviewing Your Instance Settings and Selecting an SSH Key Pair

- On this page, check that all settings for the AWS instance are appropriate. Then click **Launch**.
- You are prompted to select a public-private key pair for connecting securely to your AWS instance via SSH. If you have such a private key, select the option **Choose an existing key pair**. Otherwise, select the option **Proceed without a key pair** and check the “I acknowledge...” box.

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. [Learn more about removing existing key pairs from a public AMI.](#)

Choose an existing key pair

Select a key pair

a

☒ I acknowledge that I have access to the selected private key file (a.s.pem), and that without this file, I won't be able to log into my instance.

Cancel

Launch Instances

- Finish by clicking the **Launch Instances** button.

Your instance should now appear under **Instances** in the EC2 Management Console. After it has started up, **Instance State** will be “running”:

<div> <div>Launch Instance</div> <div>Connect</div> <div>Actions</div> </div>						
<div> <div>Filter by tags and attributes or search by keyword</div> </div>						
<input type="checkbox"/>	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks
<input type="checkbox"/>	J-dev	i-05372998d57c424da	c5.large	eu-west-1a	● running	● 2/2 checks ...

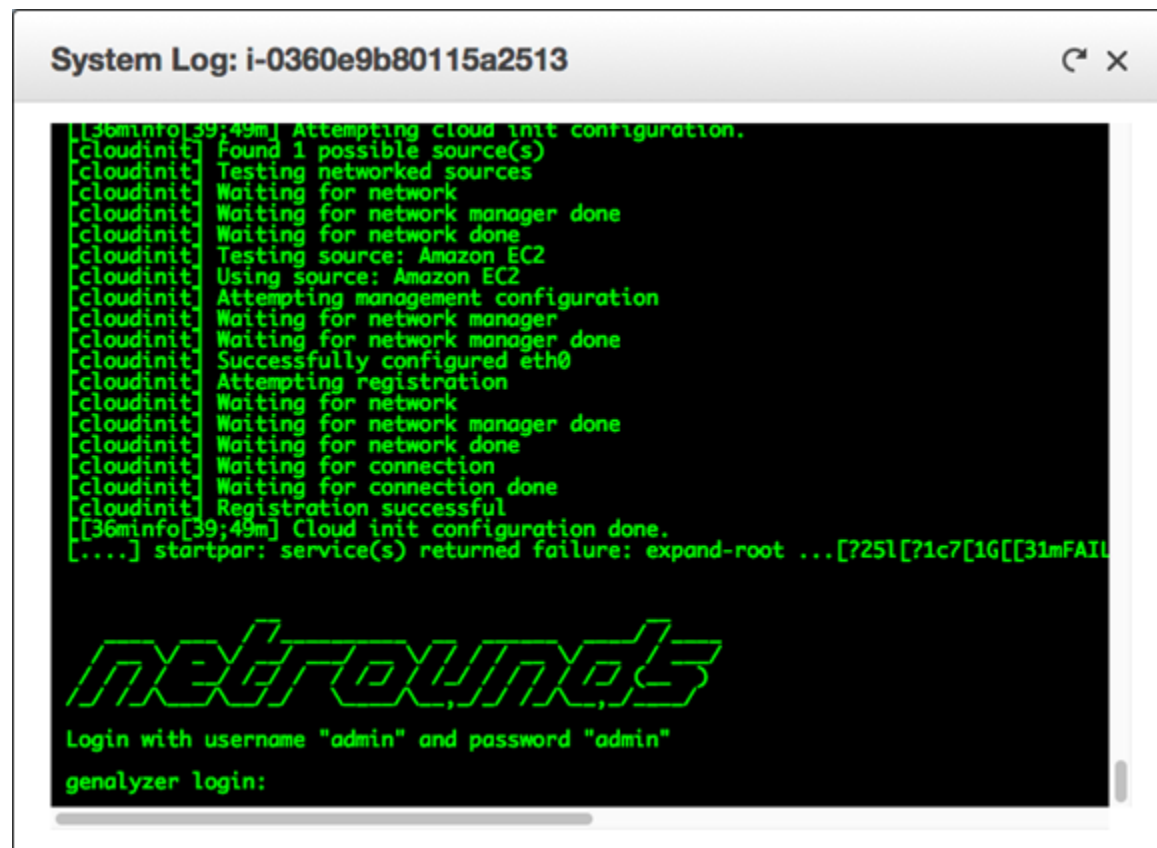
The Test Agent will now automatically register with Control Center and will then appear in the Control Center web GUI under **Test Agents**. Check for the AWS instance name in that view to verify that the Test Agent has registered.

Verifying Successful Test Agent Configuration

To verify that the cloud-init configuration of the vTA instance has been successful and that you get access to the Test Agent user interface, do as follows:

- Select the Paragon Active Assurance AMI in the AMI view.
- Click the **Actions** button and select **Instance Settings** → **Get System Log**.

The log should look something like this:



```

System Log: i-0360e9b80115a2513

[36minfo[39;49m] Attempting cloud init configuration.
[cloudinit] Found 1 possible source(s)
[cloudinit] Testing networked sources
[cloudinit] Waiting for network
[cloudinit] Waiting for network manager done
[cloudinit] Waiting for network done
[cloudinit] Testing source: Amazon EC2
[cloudinit] Using source: Amazon EC2
[cloudinit] Attempting management configuration
[cloudinit] Waiting for network manager
[cloudinit] Waiting for network manager done
[cloudinit] Successfully configured eth0
[cloudinit] Attempting registration
[cloudinit] Waiting for network
[cloudinit] Waiting for network manager done
[cloudinit] Waiting for network done
[cloudinit] Waiting for connection
[cloudinit] Waiting for connection done
[cloudinit] Registration successful
[36minfo[39;49m] Cloud init configuration done.
[....] startpar: service(s) returned failure: expand-root ...[?25l[?1c7[1G[[31mFAIL

netrounds
Login with username "admin" and password "admin"
genalyzer login:
  
```

Troubleshooting

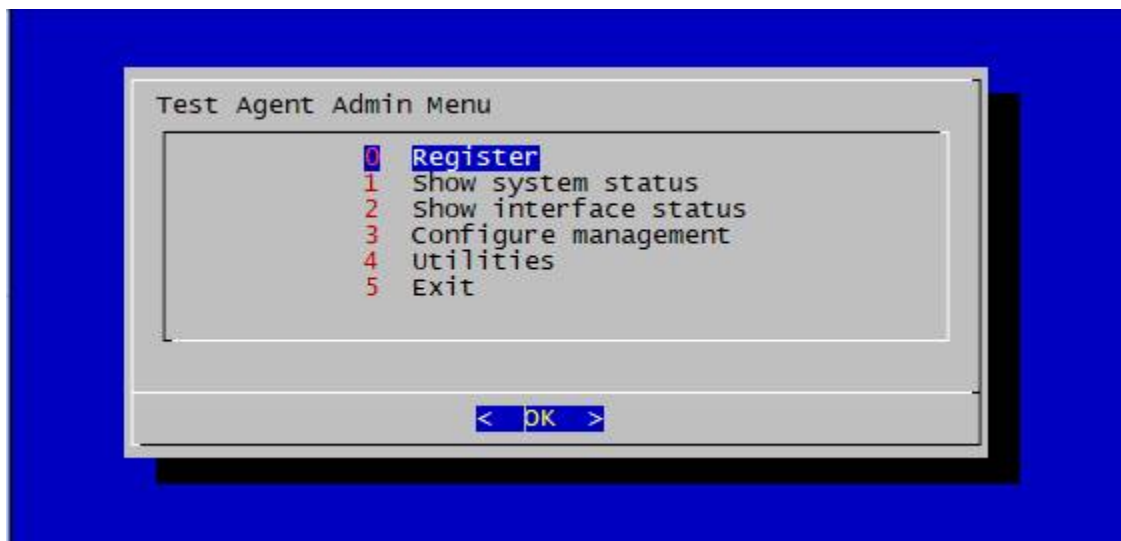
If the vTA does not show up in Control Center, it is useful to open its local console to investigate the cause of the problem. This requires that you supplied an SSH public key when creating the AWS instance (see section ["Reviewing Your Instance Settings and Selecting an SSH Key Pair" on page 7](#)).

- In the **Instances** view, inspect the public IP address of the instance.
- At a command prompt, type:

```
ssh -i <private_key> admin@<instance_public_ip>
```

where **<private_key>** is the name of the file holding your SSH private key and **<instance_public_ip>** is the public IP address of the vTA instance.

You are now taken to the Test Agent admin menu:



The functionality found here is described in the Paragon Active Assurance support documentation: see the 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.
- **Utilities** → **Root shell** for leaving the local console and going to the Linux prompt.