Vectra NDR

Azure vSensor Deployment Guide

Version: November 7, 2025

Table of Contents

© 2025 Vectra AI, Inc.

Introduction

This document describes the pre-requisites, specifications, and steps required to deploy a Vectra NDR (formerly Detect for Network) Virtual Sensor (vSensor) in an Azure subscription to monitor cloud Infrastructure-as-a-Service workloads. These Sensors can be paired with Brains of any type.

Azure Sensors can be deployed in configurations that support up to 1 Gbps or 2 Gbps of network throughput per Sensor. The input to the Sensor can be from any VxLAN-based 3rd party packet broker or the Microsoft virtual network tap (VTAP) when it becomes available in your region.

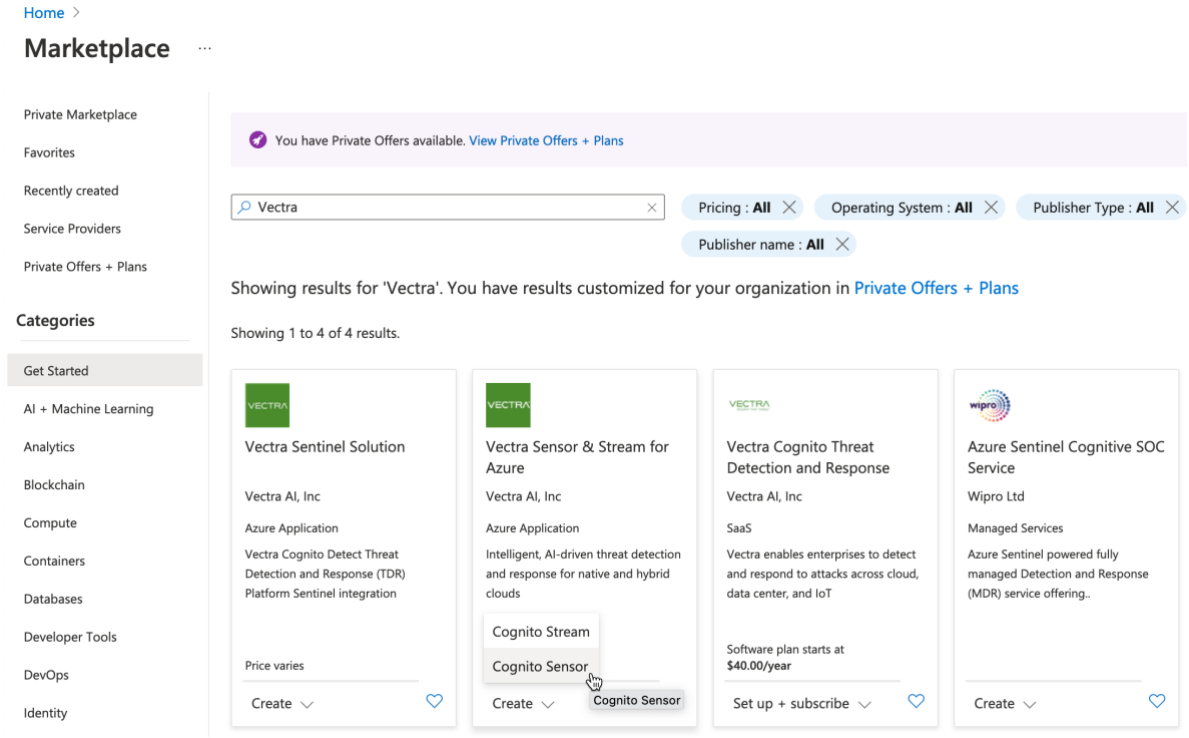
Azure Sensors can be used in both Respond UX and Quadrant UX deployments. For more detail on Respond UX vs Quadrant UX please see Vectra Analyst User Experiences (Respond vs Quadrant). One of the below guides should be the starting point for your overall Vectra deployment:

- Vectra Respond UX Deployment Guide
- Vectra Quadrant UX Deployment Guide

Either of the above guides cover basic firewall rules needed for the overall deployment and initial platform settings. Virtual Sensor (VMware, Hyper-V, KVM, AWS, Amazon, and GCP) configuration and pairing and covered in their respective guides. Physical appliance pairing is covered in the Vectra Physical Appliance Pairing Guide. Please see the Vectra Product Documentation Index on the Vectra support site for additional documentation including deployment guides for CDR for M365 / IDR for Azure AD and CDR for AWS.

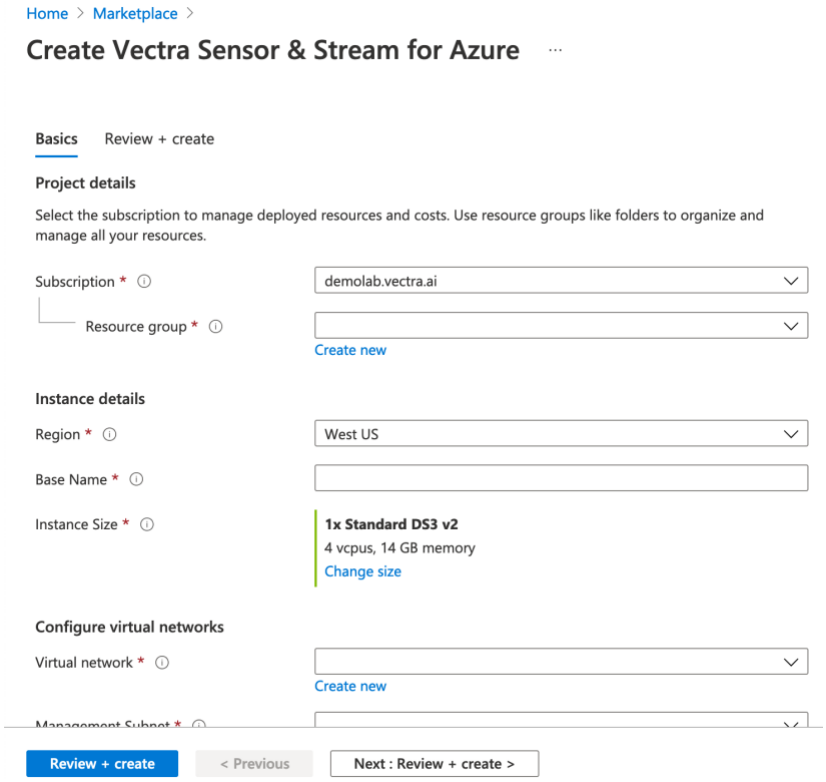
Deployment from Azure Marketplace

- Browse to the Azure Marketplace and search for “Vectra”.
- Under the “Vectra Sensor & Stream for Azure” offering, click on “Create” and select the “Cognito Sensor”.



© 2025 Vectra AI, Inc.

- You will now need to fill in details on the “Create Vectra Sensor & Stream for Azure” screen that follows.



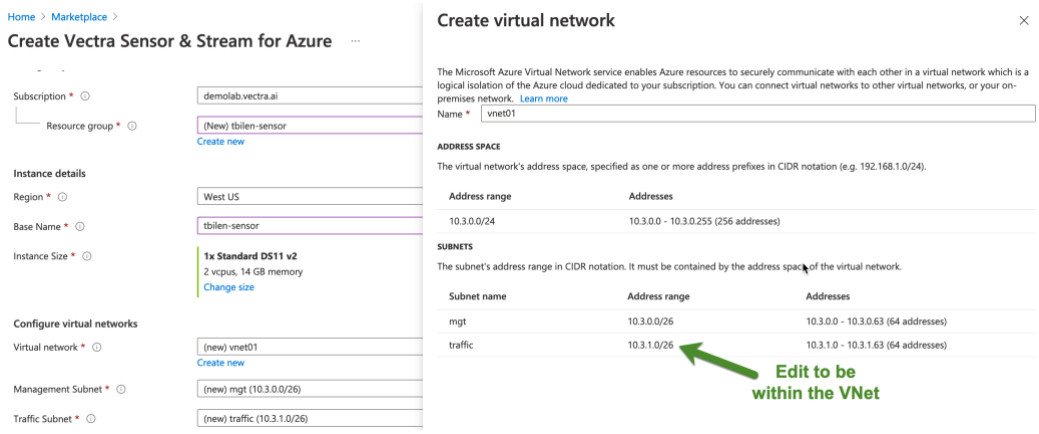
During this process you will be asked to fill in the following fields:

- **Subscription** – This should default to your current subscription. All resources in an Azure subscription are billed together.
- **Resource group** – Each Sensor must be deployed in a different resource group. Either select an existing resource group with no Sensors, or create a new resource group using the “Create new” link.
- **Region** – Select the region to deploy the Sensor into. To optimize costs, always keep the source and target (Sensor and Brain) in the same region.
- **Base Name** – Base name for all the resources that will be created as part of this deployment.
- **Instance Size** – VM instance size for Detect for Network Sensor. DS3_v2 supports approximately 2 Gbps and Ds11_v2 supports approximately 1 Gbps.

VM Size ↑↓	Family ↑↓	vCPUs ↑↓	RAM (GiB) ↑↓	Data disks ↑↓	Max IOPS ↑↓	Temp storage (GiB) ↑↓
Most used by Azure users			**The most used sizes by users in Azure**			
DS3_v2	General purpose	4	14	16	12800	28
D-Series v2			**The 2nd generation D family sizes for your general purpose needs**			
DS3_v2	General purpose	4	14	16	12800	28
DS11_v2	Memory optimized	2	14	8	6400	28

- **Virtual network** – Select an existing virtual network (VNet) or choose the “Create new” option and create a new VNet to deploy the Sensor into.
    - !! Please take special note that when choosing the “Create new” option, Azure previously defaulted the traffic subnet to be out of range with respect to the VNet address space. In the screenshot example below, this could be remedied by simply editing the traffic subnet to 10.3.0.64/26. If this change is not made, the deployment will fail. Please choose appropriate ranges for the VNet, Management Subnet, and Traffic Subnet are selected. Edit if necessary.

© 2025 Vectra AI, Inc.



- **Management Subnet** - The Sensor must be able to reach your Vectra Brain over HTTPS (443) and SSH (22). Select a management subnet that can enable this reachability.
- **Traffic Subnet** - This is the subnet where the Sensor can receive traffic from the packet broker over VxLAN (UDP 4789) or directly from the Azure VTAP (when available).
- **Brain Hostname or IP Address** - The IP address or the Fully Qualified Domain Name (FQDN hostname) of the Vectra Brain.
  - This address must be reachable from the Sensor's management subnet over port 22 and 443.
- **Registration Token** - This token must be copied from the *Data Sources* > *Network* > *Sensors* > *Sensor Configuration* > *Sensor Registration and Pairing* page of the Vectra UI.
  - The token is valid for 24 hours and can be regenerated on-demand
  - A valid registration token must be presented by the Sensor in order to register with the Brain so that the Sensor will become "Available" for pairing.
  - Instructions to generate a Sensor registration token are included in Cloud Sensor Registration Token below.
- **Public SSH Key** – Generate an RSA SSH key pair using any standard tool. Enter the public key in this field. Retain the private key safely for SSH access to the Sensor.
  - This will allow the **"vectra"** user to log into the Sensor's command line interface.
  - Azure has a "SSH Keys" function that can be loaded in another tab to generate a key pair.
  - You may need to make the key readable to you using a command such as:
    - **chmod 400 vectra.pem**
  - Example login command:
    - **ssh -i <private key path> vectra@BrainHostnameOrIP**
- **SSH user (only "vectra" will work)** – Leave this at the default.

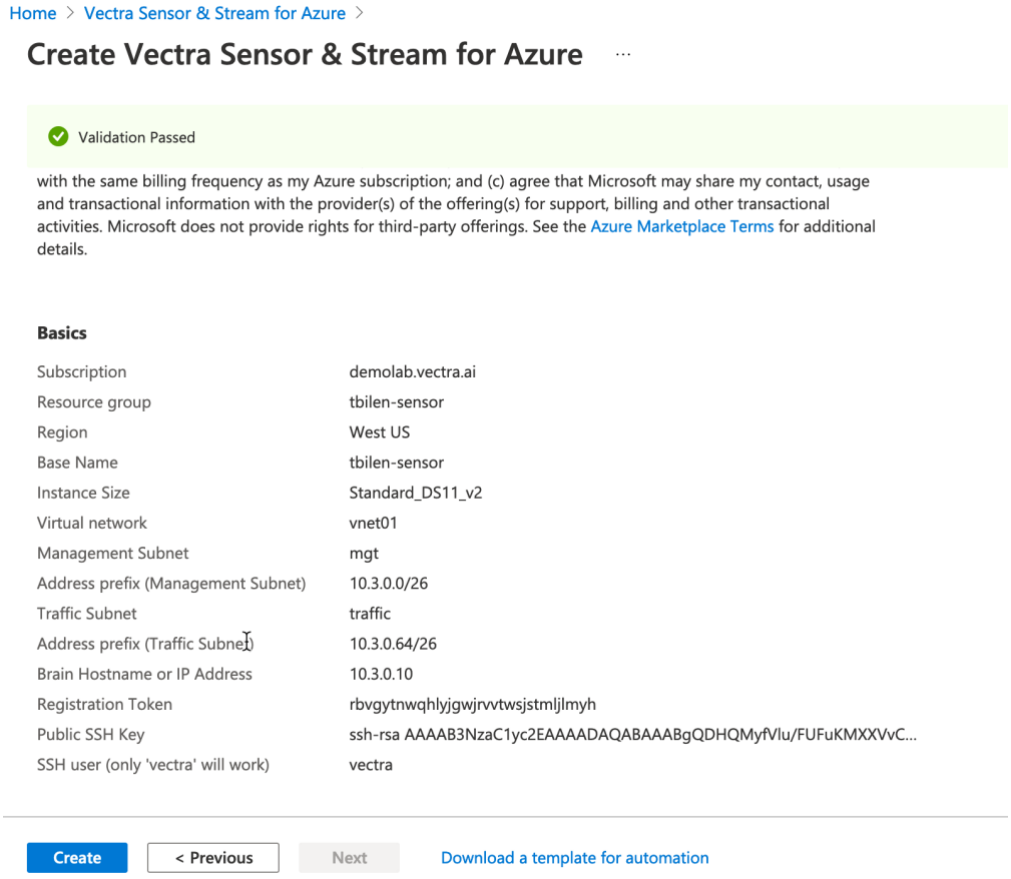
Azure Network Security Group Considerations

The vSensor deployment does not create any Azure Network Security Groups. If you choose to apply a security group, ensure the following connectivity is allowed.

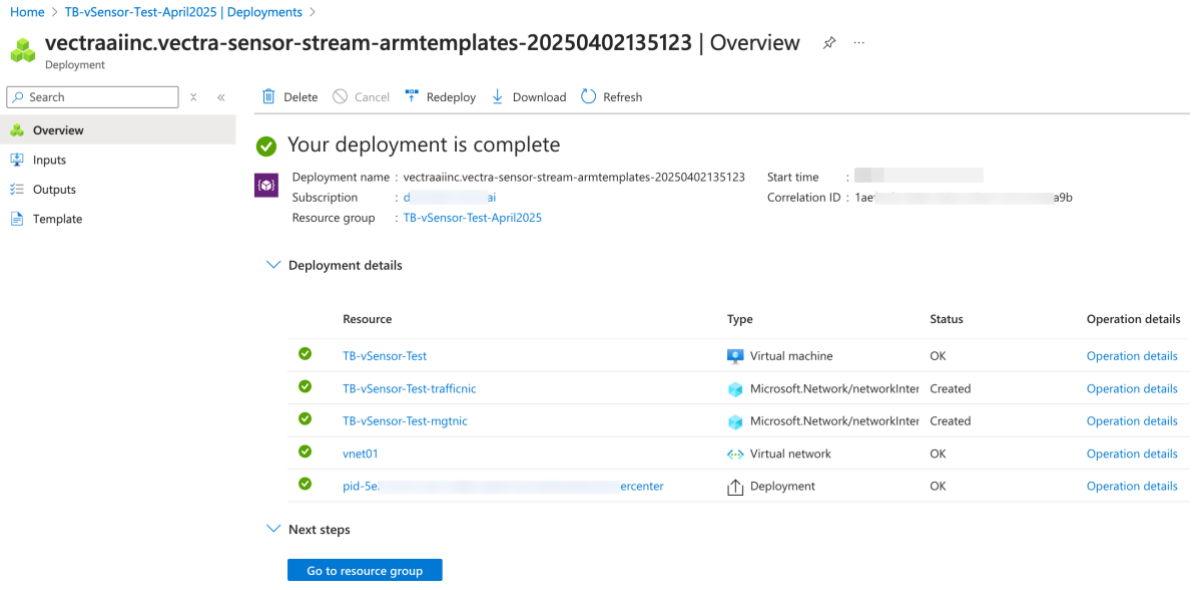
Source	Destination	Protocol/Port	Description
Admin Hosts	vSensors	TCP/22 (SSH)	CLI access to vSensor
Brain	vSensors	TCP/22 (SSH)	Remote management and troubleshooting
vSensors	Brain	TCP/22 (SSH) TCP/443 (HTTPS)	Pairing, metadata transfer, and ongoing communication
Traffic Source	Front end IP for LB	VXLAN 4789	Allows traffic to the Front end IP of the Load Balancer created during vSensor deployment.

© 2025 Vectra AI, Inc.

- Click “Review + create” and you’ll be presented with a screen where you can review your configuration before creating.



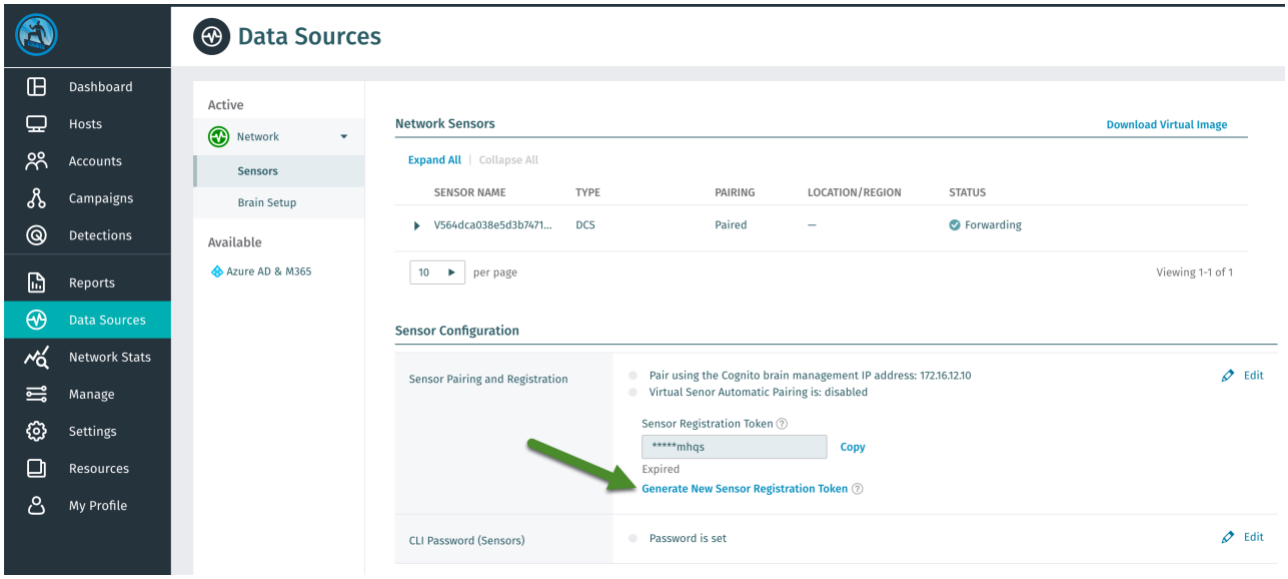
- Click “Create”
  - Some status messages will appear on the top right and then you will see another screen with full details as the deployment progresses and then completes.



© 2025 Vectra AI, Inc.

Cloud Sensor Registration Token

- During Sensor deployment a valid Sensor registration token must be presented to the Brain so that the Sensor can become "Available" to Pair. This token can be created in the Vectra UI. It is valid for 24 hours and can be regenerated at any time.
- Navigate to *Data Sources* > *Network* > *Sensors* > *Sensor Configuration* > *Sensor Registration and Pairing.*



- If you have a valid token, you can "Copy" it here for later use using the link.
  - Otherwise, click the "Generate New Sensor Registration Token" link.
  - Upon saving the page a new token will be generated that can be copied.
  - Use this token for Sensor deployment (tokens expire 24 hours after generation).

Forwarding Traffic to Azure vSensors

Microsoft no longer supports basic load balancer deployment as of March 2025. For more details, please see this Microsoft Azure update. Vectra's previous version of the Azure vSensor deployed with an Azure Basic Load Balancer. The current version does not deploy with any Azure load balancer and is simply a VM with mgt and traffic (capture) NICs. To redeploy, deploy the new vSensor, redirect traffic to its capture port, and then delete the old vSensor.

If you wish to use 3rd party tools to direct traffic at the vSensor capture port, it should be VXLAN encapsulated and sent on port 4789. When using such tools, to ensure accurate metadata generation and session reconstruction, traffic distribution to Vectra sensors must maintain flow-level persistence — that is, all packets from a single network session must be seen by the same sensor. Azure standard load balancers cannot be used because the vSensor cannot respond to required Azure standard load balancer health checks. It only listens for traffic coming inbound.

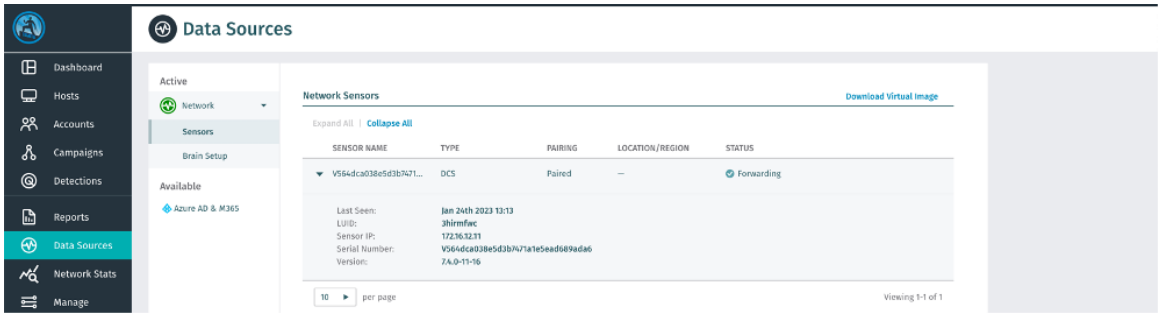
To find the IP assigned during deployment to your traffic NIC, go to your Azure resource group that was used during deployment, click on the traffic NIC (it will be the interface that ends with "trafficnic"), and there you will see the private IP that was assigned to the traffic NIC (capture interface of your newly deployed vSensor).

For guidance using Microsoft VTAP to direct traffic, please see Virtual network TAP at Microsoft. Vectra professional services can also assist in Azure deployments. Please contact your Vectra account team for details.

© 2025 Vectra AI, Inc.

Pairing Azure vSensors

- Once the deployment is complete, the vSensor will automatically boot up and reach out to the Vectra Brain.
  - If the Brain is configured for automatic pairing for virtual Sensors, the Sensor will register with the Brain and then attempt to Pair with the Brain and finally will update its software from the Brain.
  - The Sensor will show up in the *Data Sources* > *Network* > *Sensors* tab of the Vectra UI just like a physical Sensor would.



- Once you see the Sensor here and it is current with the Brain's software version, it is ready for operation.

Additional Guidance for Normal Pairing

- vSensors (vSensors) do not offer a web UI.
  - The Vectra Brain has the GUI for vSensor management at *Data Sources* > *Network* > *Sensors.*
  - Some configuration of the vSensor can be done at the CLI of the vSensor using the **`"vectra"`** user.
- As per the Vectra Respond UX Deployment Guide or Vectra Quadrant UX Deployment Guide, Sensors, including vSensors, support pairing by IP or by hostname.
  - Please refer to that guide for additional guidance on pairing by hostname or IP.
- Once the vSensor is powered on and the interface configuration is set, the vSensor will announce itself to the Brain.
  - This can take a few minutes, check firewall rules if there is an issue.
  - If the vSensor appears in the *Data Sources* > *Network* > *Sensors* page then it has made a successful HTTPS connection to the Brain.
  - If the vSensor does not appear in the *Data Sources* > *Network* > *Sensors* page check that the vSensor has IP connectivity and that TCP port 443 (HTTPS) is permitted through your firewall.
  - If the vSensor is unable to pair with the Brain, complete its initial update or forward metadata to the Brain check that TCP port 22 (SSH) is permitted through your firewall.
- If **"Auto Pairing"** is enabled in *Data Sources* > *Network* > *Sensors* > *Sensor Configuration*, the pairing process will begin automatically.
  - Enabling **"Auto Pairing"** is a best practice during rollout.
- If **"Auto Pairing"** is not enabled, the vSensor must be manually paired by clicking on the "Pair" icon
  - You will then be presented with a dialog box where you can start the pairing process.
- Initially you will see the **"Pairing Status"** as **"Pairing"** once the vSensor has successfully announced itself to the Brain.
  - Once pairing is complete, the **"Pairing Status"** will change to **"Paired"**, and the **"Status"** should change to **"Forwarding"** once traffic is successfully being forwarded from the vSensor to the Brain.

© 2025 Vectra AI, Inc.

- **Please note:**
  - vSensors, like physical Sensors, will update themselves to stay current with their Brain
  - Certain vSensor CLI functions and traffic functions will become available only after the vSensor has fully updated.
  - Depending on the specific version of the vSensor, you may see errors or warnings when running CLI functions during the period of time when the vSensor is still updating.
  - "show version" can be done at the CLI to see the current version and if the vSensor is still upgrading itself from the Brain to become fully updated. Please bear in mind that state changes are not immediate, and you may need to execute this command more than once to see the state change.
- The vSensor can be renamed or have its location labeled as desired by clicking on the pencil icon on the right of the vSensor.

Pairing Guidance for Special Situations

Pairing with new or changed Brains:

- If you have a backup of your Brain and restore it to a new Brain with the same configuration (IP or hostname), previously paired Sensors (including vSensors) will connect to the new Brain automatically as the Sensor state is saved in the backup.
- If the Brain IP has changed but otherwise remains the same, the vSensors may be updated to the new IP address using the **`"set brain"`** command.
- Existing tunnels have to terminate to re-establish connection to a new Brain. This can be accomplished a few different ways.
  - Naturally, because the original Brain is no longer reachable due to firewall change, hardware or software failure, etc.
  - Unpairing the vSensor from the original Brain and having the vSensor attempting communication to the original Brain.
  - Using the **`"set brain"`** command at the CLI will terminate an existing tunnel and attempt to start pairing with a new Brain.
- If you have a Brain that will not be restored from backup that you wish to pair an existing vSensor to, this is possible via the use of the "Sensor Registration Token".
  - Retrieve or generate a current Sensor Registration Token from *Data Sources* > *Network* > *Sensors* > *Sensor Configuration* > *Sensor Pairing and Registration* in the Brain GUI.
  - Perform the **`"set registration-token <token>"`** command at the Sensor CLI.
  - Finally perform the **`"set brain <IP or Hostname>"`** command at the Sensor CLI.

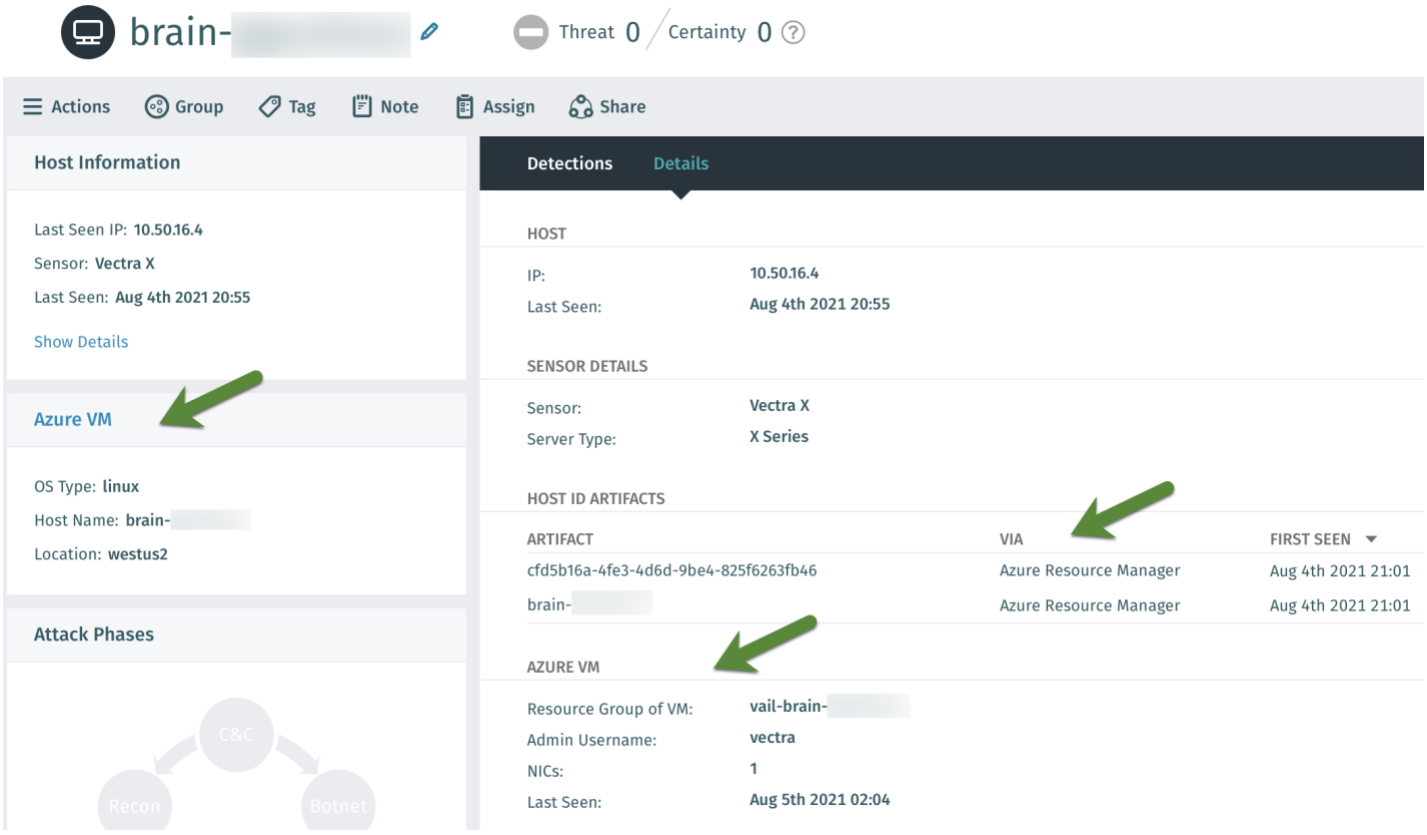
© 2025 Vectra AI, Inc.

Host ID Extraction

To enable efficient investigation of cloud hosts, the Vectra Brain can be configured to periodically query the API offered by the Azure Resource Manager to gather additional information about hosts running in Azure. This information contributes to Vectra's automated Host identification (Host ID) and adds information to the Host details screen.

It is a best practice is to enable this integration. Detect for Network attributes Detections to a Host or an Account. For some algorithms that support learning, Detections must be attributable to a Host (including statically defined hosts), rather than a generic Host container (denoted by IP-x.x.x.x in the Vectra UI). Enabling this integration can help contribute to more complete Detection as a result.

Please see this example screenshot below of a host in Azure where additional context was pulled through API calls:



As you can see, the integration provided the following (viewable by drilling down to the Host and then the Details tab):

- Host ID artifacts including a unique identifier and hostname
- Resource group, Admin, NICs, and Last Seen date/time
- Summary information on the left-hand side including OS, Host Name, and Location

Creating an Azure AD application and service principal that can read Azure management APIs is required. At minimum, read access to the following endpoints of the Azure resource manager will be required:

- https://management.azure.com/subscriptions
- https://management.azure.com/subscriptions/SUBSCRIPTION/providers/Microsoft.Network/networkInterfaces
- https://management.azure.com/subscriptions/SUBSCRIPTION/providers/Microsoft.Compute/virtualMachines
- https://management.azure.com/subscriptions/SUBSCRIPTION/providers/Microsoft.Compute/virtualMachineScaleSets

© 2025 Vectra AI, Inc.

An alternative is to provide the Vectra Brain permissions to query the following APIs (this is a wider scope):

- Microsoft.Network//read
- Microsoft.Compute//read

The simplest way to accomplish this is to apply the built-in Azure "Reader" role to the application. This will allow your Vectra Brain to view all resources but does not allow it to make any changes.

Once created, you will then enter the Application (client) ID, Directory (tenant) ID, and Application Secret value into the Vectra UI to enable the integration. Multiple credential sets can be added if required.

Configuration Example

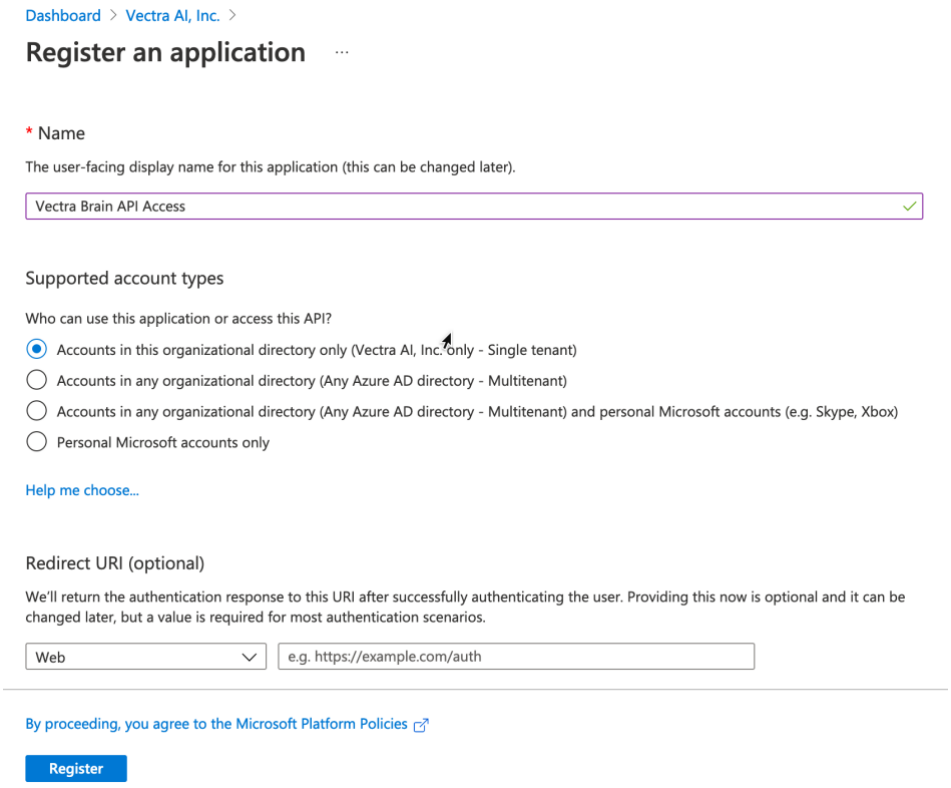
Resources

- Full instructions for creating Azure AD application and service principal are available from Microsoft here:
  - https://docs.microsoft.com/en-us/azure/active-directory/develop/howto-create-service-principal-portal
- Details regarding Azure built-in roles are available from Microsoft here:
  - https://docs.microsoft.com/en-us/azure/role-based-access-control/built-in-roles

Steps

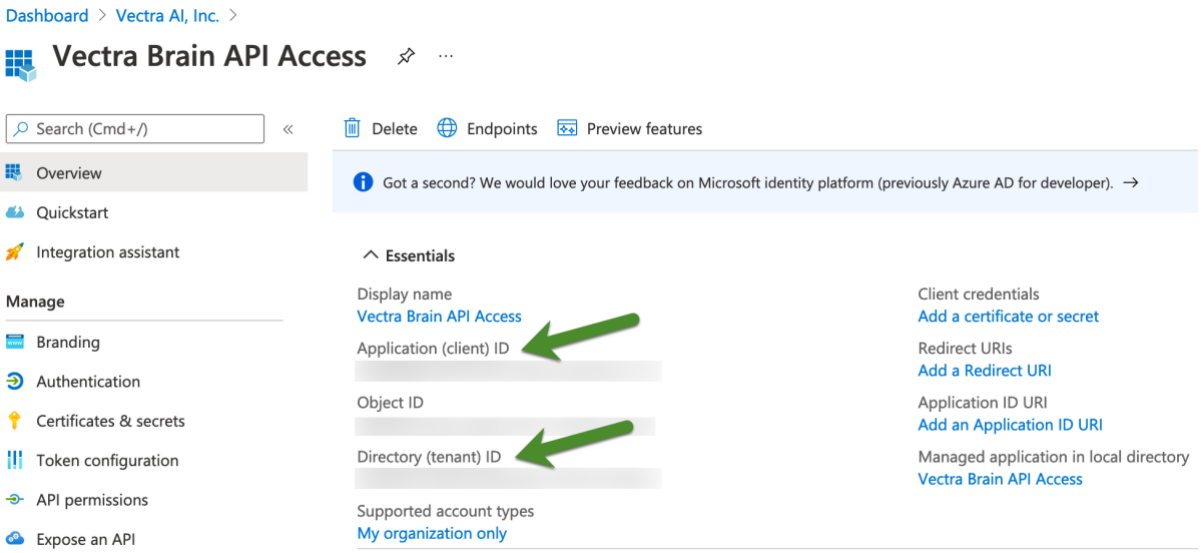
To enable this functionality, the following steps need to be completed for every Azure subscription where there are Sensors deployed. In this example we will apply the "Reader" role at the Azure Subscription level.

- After signing in to the Azure AD portal, navigate to *Azure Active Directory > App registrations* and select "New registration".
  - Give the application a "Name" of your choosing and choose a supported account type.
  - A Redirect URI is not required.

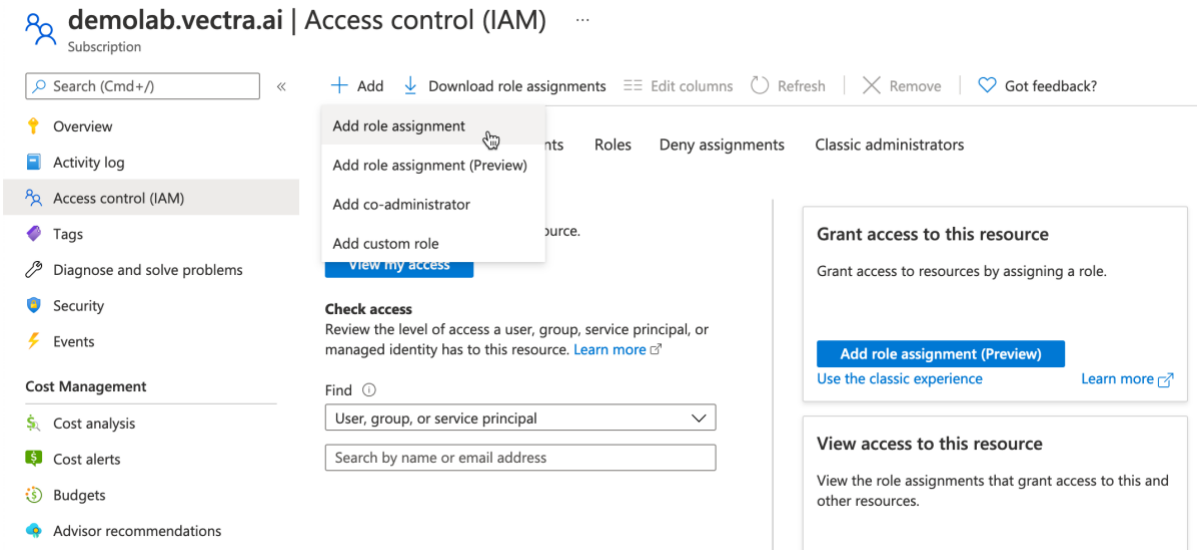


© 2025 Vectra AI, Inc.

- Click “Register” and then copy the Application (client) ID and Directory (tenant) ID for later use:

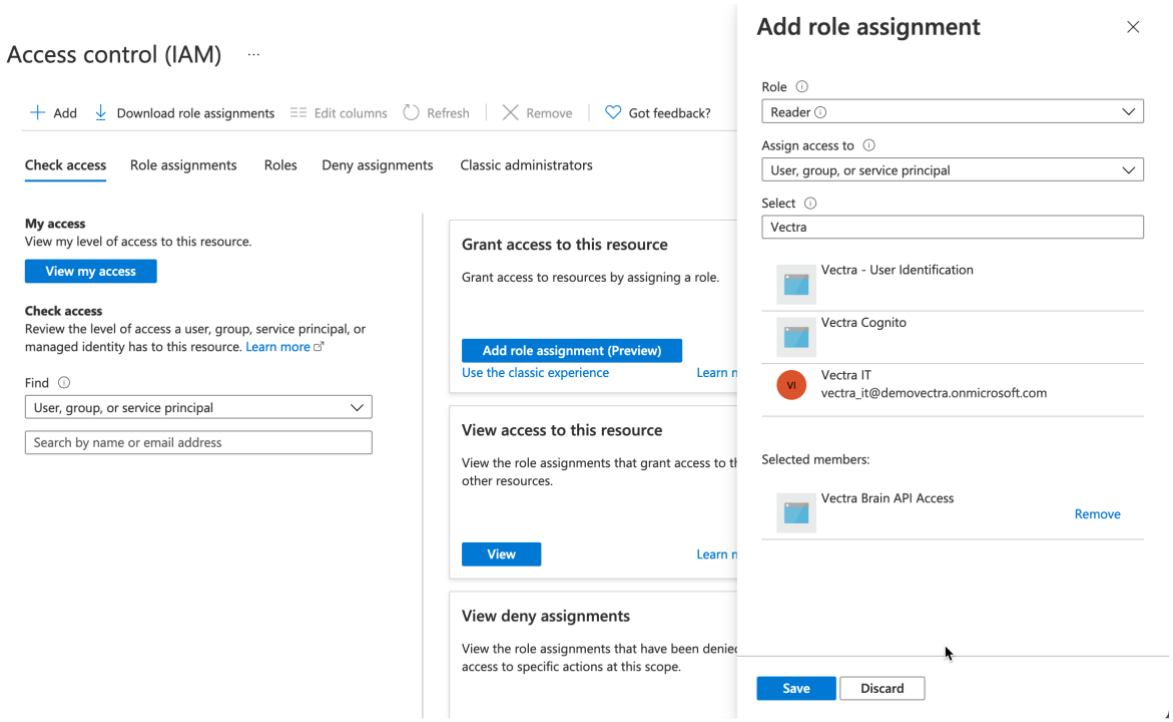


- Now we will assign an Azure role to the application:
  - In the Azure portal select “Subscriptions” and select which subscription to apply the role to by clicking on it.
  - Select “Access control (IAM), then click “+ Add”, and then click “Add role assignment”.

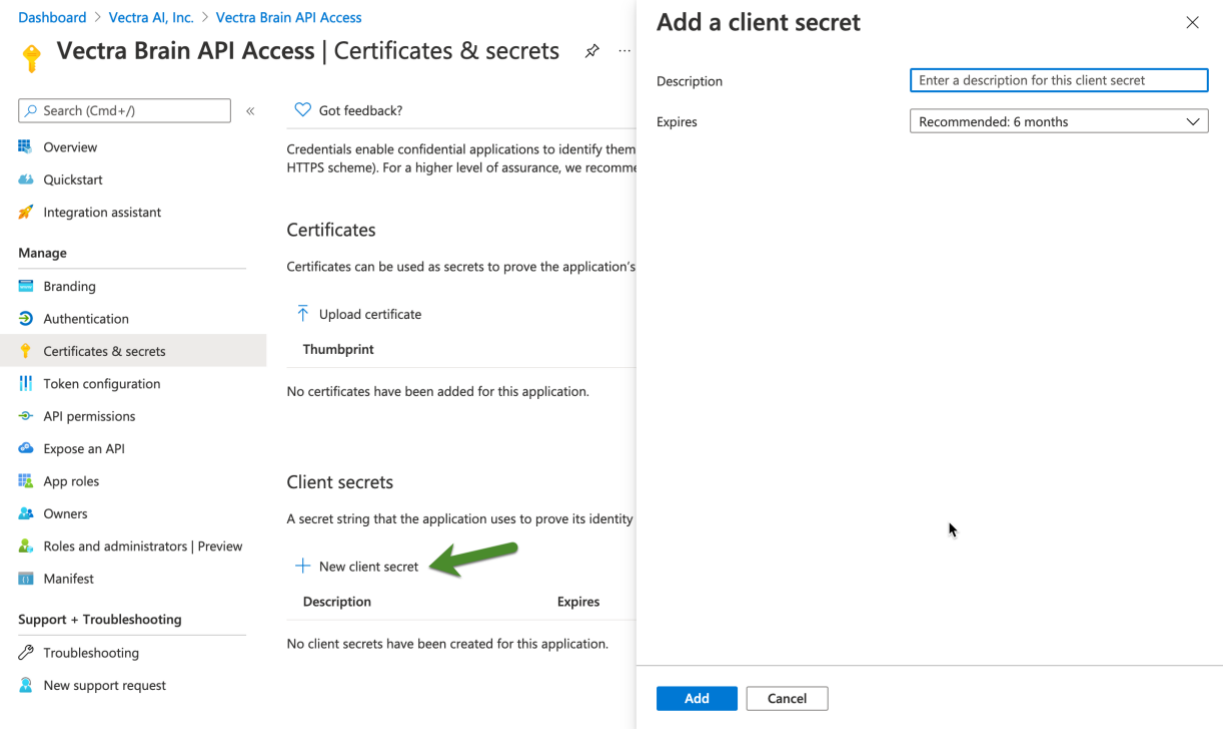


  - Select “Reader” under Role.
  - Leave “Assign access to” unchanged (it should say “User, group, or service principal”)
  - Search for your application and select it.
  - Click “Save”.

© 2025 Vectra AI, Inc.

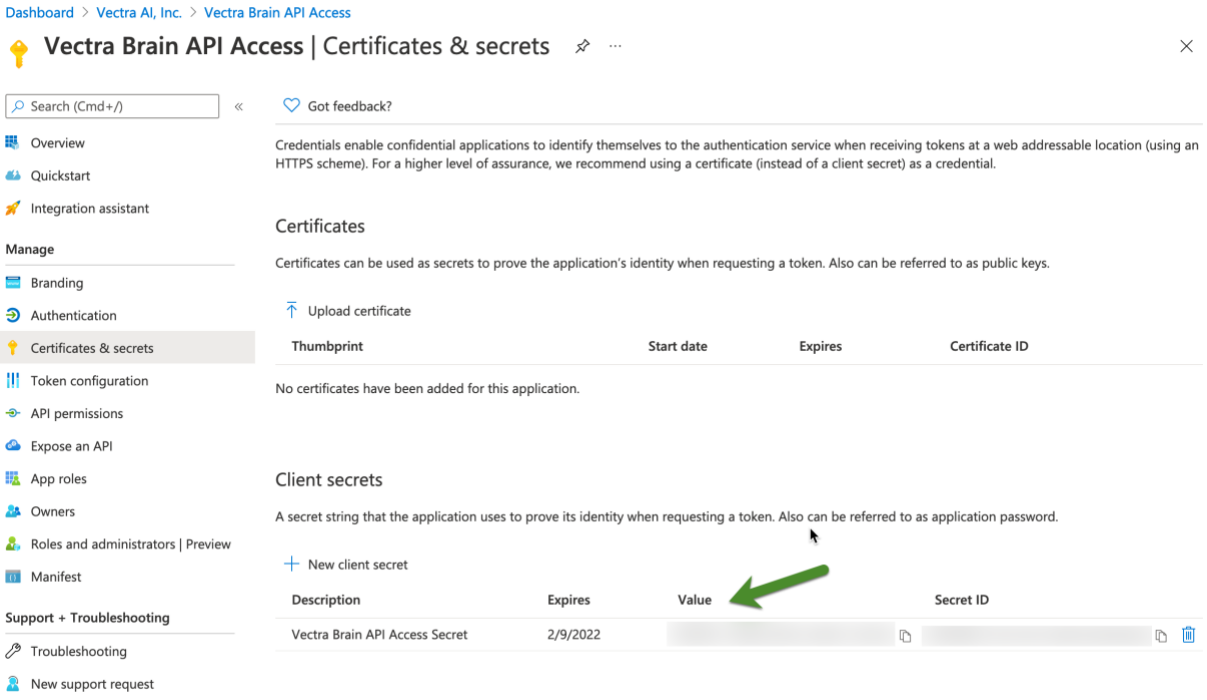


- Next, we will create a new application secret:
  - In the Azure AD portal, navigate to *App registrations* and select your application.
  - Select “Certificates & secrets”, click “+ New client secret”, enter a “Description” and select and when you want this to expire:

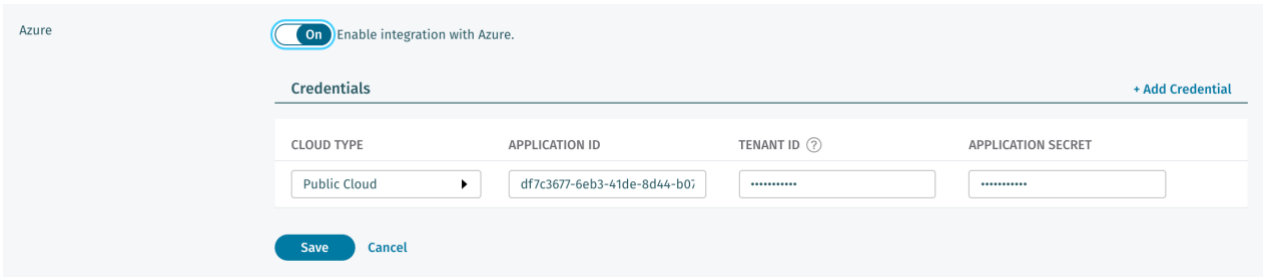


  - Click “Add”
  - Once created, you must copy the “Value” before navigating away from this screen as you cannot retrieve it later.

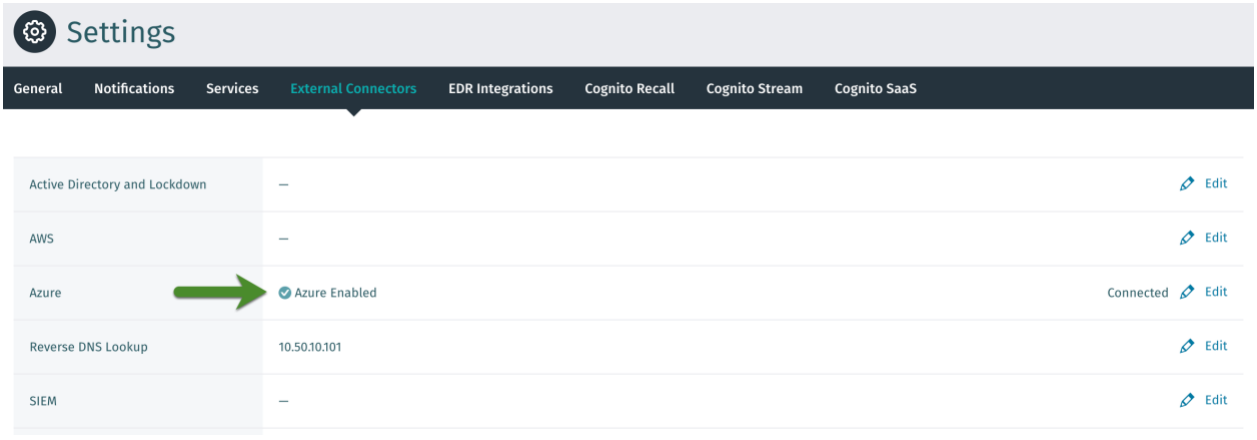
© 2025 Vectra AI, Inc.



- To complete the integration, in the Vectra UI, browse to *Settings* > *External Connectors*.
  - Click on the pencil or "Edit" link for Azure.
  - Toggle the integration to "On" and fill in the Application (client) ID, Directory (tenant) ID, and Application secret values that you created above:
  - Click "Save". Multiple sets of credentials can be configured if required for different subscriptions.



  - The connection will best tested and if successful, you will see a green checkmark showing that Azure Resource Manager integration has been completed.



© 2025 Vectra AI, Inc.

Viewing and Modifying Virtual Networks Considered by the Connector

The connector will only look at virtual machines or virtual machine scale sets that belong to some virtual networks, not necessarily all virtual networks that the credentials have access to. Specifically, by default, the connector will only look for virtual machines with network interfaces in the virtual networks where a Vectra Azure Sensor traffic/capture interface is deployed into, and all the virtual networks with a "double peering" to it (if the Sensor traffic interface lives in virtual network A, and virtual network A is peered with virtual network B and C, but only virtual network B is peered with virtual network A, then the connector will look for virtual machines in virtual networks A and B only). If a virtual machine has multiple network interfaces, the connector will provide host identification only for the IP addresses belonging to interfaces on monitored virtual networks.

Vectra limits the scope of the connector to only virtual networks that can reach an Azure Sensor for the following reasons:

- Vectra wants to avoid an excessive number of connections to the Azure resource manager.
- To improve performance, Vectra wants to avoid storing unnecessary information in the platform for resources that we are not monitoring.
- In the cloud it is possible that you may have some isolated virtual networks that are not monitored by Vectra Sensors but have overlapping IP space with some other networks that are monitored. Duplicate IPs are not supported by Vectra. To avoid Host ID issues, we only consider virtual machines on the networks that appear reachable by our Sensors.

If you want to add or remove some virtual networks from the set of virtual networks that Vectra monitors, you can do so through the CLI (ssh to the Brain as the “vectra” user). This functionality is not available in the Vectra UI. The use case for adding or removing virtual networks could be the following:

- If you have some tunneling or forwarding enabled and Vectra Azure Sensors see traffic from virtual networks that are NOT directly connected to our Sensor:
  - You will want to add the networks to the monitored networks list to extend Host ID coverage to the tunneled/forwarded traffic’s associated Hosts.
- If you do NOT have any Azure Sensors deployed, but mirror Azure traffic into your data center where you have Vectra physical or virtual Sensor coverage:
  - You will want to add the networks to the monitored networks list to extend Host ID coverage to the mirrored traffic.
- If you do have some overlapping IP space on a network that appears reachable by our Sensor, but that it is not monitored by our Sensor:
  - You will want to remove the overlapping networks from the monitored networks to avoid any Host ID issues.
- If you have a lot of peered networks but only some are monitored by the Sensor:
  - You will want to remove the non-monitored networks from the monitored networks list to increase performance and reduce load on the Azure resource manager.

The command to add or remove virtual network from the Vectra CLI is "**`set azure vnets`**". The command has a very helpful help message illustrating its use:

© 2025 Vectra AI, Inc.

```
vscli > set azure vnets -h
Usage: set azure vnets [OPTIONS]

  Set Azure virtual network information.

  Usage example:

      > set azure vnets --whitelist A --whitelist B

      will set the whitelist to [A, B]

      > set azure vnets --blacklist A --blacklist B

      will set the blacklist to [A, B]

      > set azure vnets --clear-whitelist

      will set the whitelist to [] (ie remove the whitelist)

      > set azure vnets --clear-blacklist

      will set the blacklist to [] (ie remove the blacklist)

Options:
  --whitelist TEXT   virtual network to include in whitelist
  --blacklist TEXT   virtual network to include in blacklist
  --clear-whitelist  reset whitelist to empty list
  --clear-blacklist  reset blacklist to empty list
  -h, --help         Show this message and exit.
```

The command "**`show azure vnets`**" will show the virtual networks configuration, including the whitelist (networks to ADD), the blacklist (networks to REMOVE), the Sensors network, and finally the overall monitored networks, which is simply computed as ((Sensors networks + peered) + whitelist) - blacklist. It may take up to 10 minutes for the summarized "**`monitored networks`**" to update in the "**`show azure vnets`**" command output. Example of usage:

```
vscli > set azure vnets --whitelist /subscriptions/b3fe75ab-94a2-4322-84af-016eb01ff43e/resourceGroups/tbilen-test_azure_brain/providers/Microsoft.Network/virtu
alNetworks/tbilen_brain_test
Output: success
vscli > show azure vnets
Output:
    Blacklist Vnets:
    Monitored Vnets:
        /subscriptions/                          /resourcegroups/tbilen-test_azure_brain/providers/microsoft.network/virtualnetworks/tbilen_brain_test
    Sensors Vnets:
    Whitelist Vnets:
        /subscriptions/                          /resourcegroups/tbilen-test_azure_brain/providers/microsoft.network/virtualnetworks/tbilen_brain_test
vscli >
```

Traffic Validation

Please see the following Vectra support article for recommendations on network traffic that should be examined and excluded from analysis:

- Vectra Platform Network Traffic Recommendations

Once you have directred traffic at your Sensor's capture interface though either 3rd party traffic broker or Microsoft's VTAP (once available) and have traffic flowing to the Sensor you can validate the traffic flow.

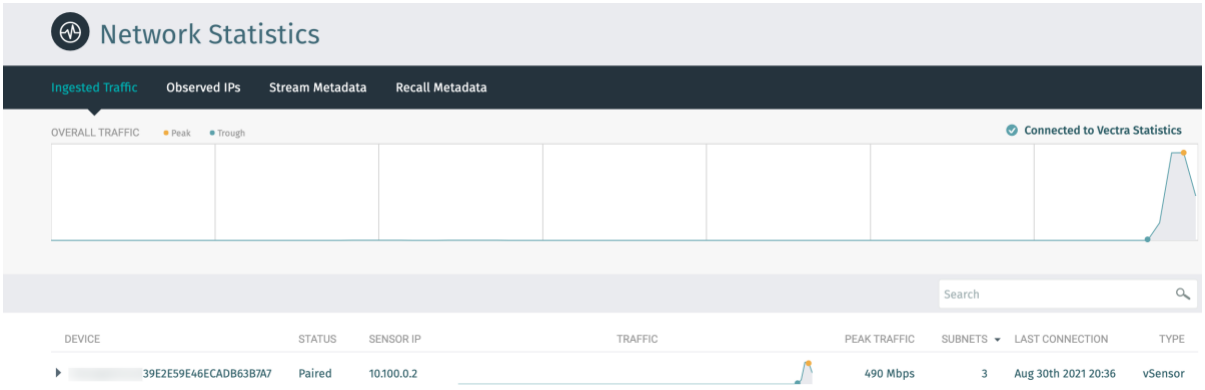
To validate flow from the Vectra side there are several methods

- To see that packets are being receive by the traffic interface, use ssh to login to the CLI of the Sensor as the "vectra" user and use the "**`show traffic stats`**" command.

© 2025 Vectra AI, Inc.

- Run the "`show traffic stats`" several times to see that packet counts are increasing.

- In the Vectra GUI if you navigate to *Network Stats* > *Ingested Traffic*, you can see the traffic graph for your Sensor.
  - For this graph to display, there must be at least 1 Mbps of traffic being captured.
  - Once traffic capture begins, it will take a few minutes for this graph to be populated. Use the CLI of the Sensor as shown above to validate that packets are flowing 1st.



- The Network Stats > Observed IPs page shows the subnets being observed and numbers of hosts seen:

SUBNET	HOSTS
10.100.0.0	1
10.250.0.0	1
10.200.0.0	1

After sending traffic to your Sensors, it is a best practice to validate that the traffic observed meets quality standards required for accurate detection and processing. Vectra's Enhanced Network Traffic Validation feature provides alarms and metrics that can be used to validate the quality of your traffic. Please see the following Vectra support article for details:

- Enhanced Network Traffic Validation

Worldwide Support Contact Information

- Support portal: https://support.vectra.ai
- Email: support@vectra.ai (preferred contact method)
- Additional information: https://www.vectra.ai/support

© 2025 Vectra AI, Inc.