

OV App Streaming (OVAS) API in Azure - Step by Step Guide

Omniverse

Exported on 09/19/2024

Table of Contents

1	Important Note	5
2	Prerequisites	6
3	Infrastructure deployment (Manual).....	7
3.1	Access subscription	7
3.2	Create resource group	8
3.3	VNET/Subnets.....	8
3.4	Create Network Security Group.....	13
3.4.1	Assign to subnet-aks and subnet-apim	16
3.5	DNS/Certificates	17
3.5.1	Create private DNS zone	17
3.5.2	Create self-signed certificates for private DNS zone	18
3.5.3	Create LetsEncrypt certificates manually for public DNS zone	20
3.5.4	Create a .pfx from the full chain certificate and private key	23
3.6	AKS Cluster	23
3.6.1	Basics	23
3.6.2	Nodepools	24
3.6.3	Networking	27
3.6.4	Review and deploy:	29
3.7	Post-deployment.....	29
3.7.1	Accessing Kubernetes (RBAC Required to access cluster resources)	29
3.7.2	Pull kubeconfig locally and check access.....	30
3.8	API Management (APIM) Service	31
3.8.1	Create the base APIM service:	31
3.8.2	Post Deployment.....	33
3.9	Application Gateway (with WAF)	34
3.9.1	Basics	34
3.9.2	Frontends	35

3.9.3	Backends.....	35
3.9.4	Configuration.....	35
3.9.5	Once deployed go to app gateway:.....	38
3.9.5.1	Health Probe.....	38
3.9.5.2	HTTPS Listener	38
3.9.5.3	Routing Rules	40
3.9.6	Post Deployment:.....	41
3.10	RBAC.....	41
4	Kubernetes Services	43
4.1	Deploy base helm charts	43
4.1.1	Set environment-specific values.....	43
4.1.2	Internal ingress controller (helm/nginx-ingress-controller)	44
4.1.3	FluxCD (helm/flux2).....	44
4.1.4	GPU Operator (helm/gpu-operator)	45
4.1.5	Memcached (helm/memcached)	46
4.1.6	ExternalDNS (scripts/external-dns)	46
4.1.7	Create required secrets	47
4.2	Deploy OVAS services	48
4.2.1	Streaming (helm/kit-appstreaming-manager)	48
4.2.2	Applications (helm/kit-appstreaming-applications)	49
4.2.3	RMC (helm/kit-appstreaming-rmc)	49
4.2.4	Deploy the custom streaming resources (manifests/omniverse/azure)	50
4.2.5	Deploy HelmRepository (manifests/helm-repositories).....	53
4.2.6	Create private DNS record for ingress controller.....	54
4.2.7	Create public DNS entry for App Gateway.....	54
4.2.8	Add API route to ingress controller in APIM	55
5	Testing.....	58
5.1	Testing From CLI.....	58
5.2	Testing with swagger docs	59

5.2.1	Disabling Subscription Key.....	59
5.2.2	Creating a stream	60
5.2.3	Verification	61
5.3	Testing with local client app (Locally installed certificates)	62
5.3.1	Client Application On Local Machine.....	63
5.3.1.1	Install NPM.....	64
5.3.2	Enabling WSS:	65
6	Client App in Azure.....	67
6.1	Creating a NGINX Ingress Controller	67
6.2	Adding Entry to Public DNS Zone	68
6.3	Current Bug (Tracked on Jira)	70
6.4	Changing Application Used	70
7	Deleting resources and environment.....	71
8	Troubleshooting	72
8.1	CORS Policy Error.....	72
8.2	NSG Rules.....	72
8.3	Forbidden access to Kubernetes	73
9	Enabling WSS Troubleshooting (Problem resolving DNS)	74
9.1	Obtain Public DNS Certificate	74
9.2	Change ExternalDNS to Use Alternative Zone	75
9.3	Setting up ExternalDNS (Required for TLS).....	76
9.3.1	Create IAM user and attach the policy	77
9.3.2	Create the static credentials	77
9.3.3	Create Kubernetes secret from credentials	78
10	Resources Deployed in This Tutorial	81
10.1	Azure subscription nv-Omniverse-Nucleus	81
10.2	AWS	83
10.2.1	Drew Como's sandbox account: omni-aws-sandbox	83
10.2.2	AWSOS-AD-Admin.....	84

1 Important Note

- ⓘ This is not official documentation. These are notes/instructions on how we got OVAS up and running in our Azure environment. Use this guide as reference.

2 Prerequisites

- Unix/Linux Operating system with Bash Shell
- Ensure you are connected to the NVIDIA VPN (needed to pull Docker Image)
- Install [Kubectl](#)¹ on your deployment system
- Install [Helm](#)² on your deployment system
- Install the [Azure CLI](#)³ on your deployment system

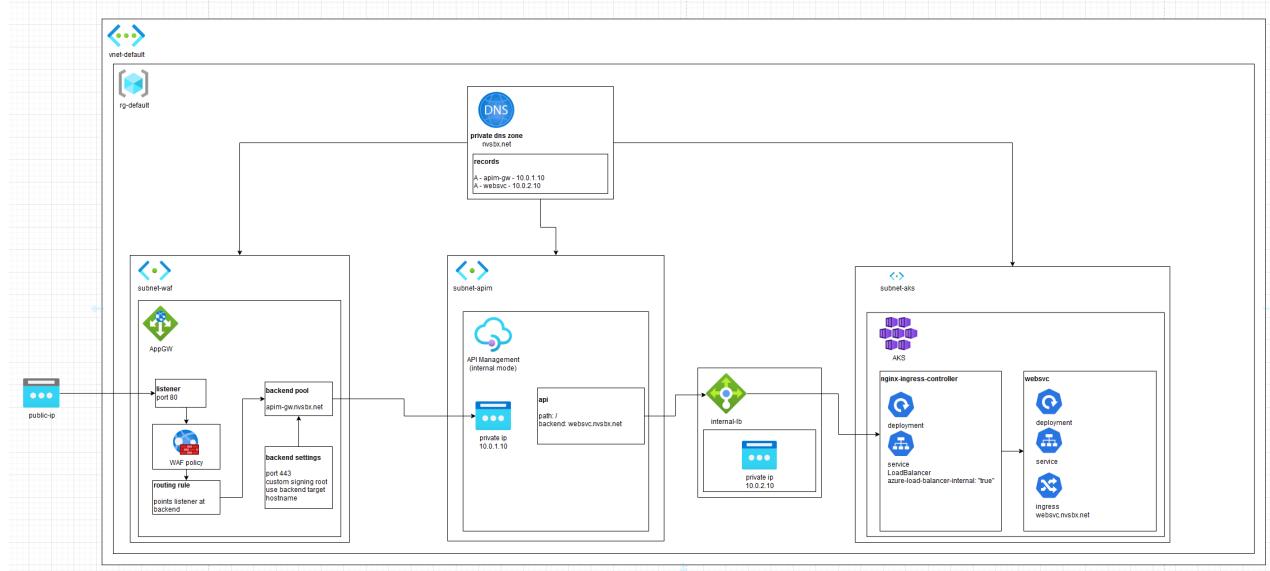
¹ <https://kubernetes.io/docs/tasks/tools/>

² <https://helm.sh/docs/intro/quickstart/>

³ <https://learn.microsoft.com/en-us/cli/azure/install-azure-cli>

3 Infrastructure deployment (Manual)

Diagram not complete, example only



3.1 Access subscription

<https://portal.azure.com/>

(i) I added an "elitang" prefix/postfix to the names of objects, this is just for personal reference. You can name your objects however you like.

In this section we make the following assumptions:

- All resources are created within a single resource group
- All resources are created in the `eastus` region
- All resource names are for example purposes only and are chosen for consistency purposes in documentation; you may choose your own resource names
- The vnet range and subnet ranges do not overlap (or are taken into consideration) with any planned peered vnets
- **For example purposes only**, the private DNS zone of our deployment is `elitang-ovas-streaming.net`

- **For example purposes only,** The public DNS zone of our deployment is nvsbx.omniverse.nvidia.com⁴
 - The public zone was already created prior to this deployment. In subsequent steps we will be adding records to create the subdomain elitang-ovas-streaming.nvsbx.omniverse.nvidia.com⁵

3.2 Create resource group

Search for resource groups and create one

Home > Resource groups >

Create a resource group

Basics Tags Review + create

Resource group - A container that holds related resources for an Azure solution. The resource group can include all the resources for the solution, or only those resources that you want to manage as a group. You decide how you want to allocate resources to resource groups based on what makes the most sense for your organization. [Learn more](#)

Project details

Subscription * [\(nv-Omniverse-Nucleus\)](#)

Resource group * [rg-elitang](#)

Resource details

Region * [\(US\) East US](#)

3.3 VNET/Subnets

1. Create a new virtual network

- **Range:** `10.2.0.0/16`

⁴ <http://elitang-ovas-streaming.nvsbx.omniverse.nvidia.com>

⁵ <http://elitang-ovas-streaming.nvsbx.omniverse.nvidia.com>

⁶ <http://elitang-ovas-streaming.nvsbx.omniverse.nvidia.com>

[Home](#) > [Virtual networks](#) >

Create virtual network

[Basics](#) [Security](#) [IP addresses](#) [Tags](#) [Review + create](#)

Azure Virtual Network (VNet) is the fundamental building block for your private network in Azure. VNet enables many types of Azure resources, such as Azure Virtual Machines (VM), to securely communicate with each other, the internet, and on-premises networks. VNet is similar to a traditional network that you'd operate in your own data center, but brings with it additional benefits of Azure's infrastructure such as scale, availability, and isolation.

[Learn more](#).

Project details

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

Subscription *	<input type="text" value="nv-Omniverse-Nucleus"/>
Resource group *	<input type="text" value="rg-elitang"/> Create new

Instance details

Virtual network name *	<input type="text" value="elitang-vnet"/>
Region * ⓘ	<input type="text" value="(US) East US"/> Deploy to an Azure Extended Zone

[Home](#) > [Virtual networks](#) >

Create virtual network

[Basics](#) [Security](#) [IP addresses](#) [Tags](#) [Review + create](#)

Configure your virtual network address space with the IPv4 and IPv6 addresses and subnets you need. [Learn more](#)

Define the address space of your virtual network with one or more IPv4 or IPv6 address ranges. Create subnets to segment the virtual network address space into smaller ranges for use by your applications. When you deploy resources into a subnet, Azure assigns the resource an IP address from the subnet. [Learn more](#)

Add IPv4 address space Delete address space					
10.2.0.0/16					
This address prefix overlaps with virtual network 'apollo-tunnel-prod-eu2-vnet'. If you intend to peer these virtual networks, change the address space. Learn more					
10.2.0.0	/16				
10.2.0.0 - 10.2.255.255 65,536 addresses					
+ Add a subnet					
Subnets	IP address range	Size	NAT gateway		
subnet-aks	10.2.0.0 - 10.2.0.255	/24 (256 addresses)	-	Edit	Delete
subnet-waf	10.2.1.0 - 10.2.1.255	/24 (256 addresses)	-	Edit	Delete
subnet-apim	10.2.2.0 - 10.2.2.255	/24 (256 addresses)	-	Edit	Delete

1. Create new subnets

- `subnet-aks` - `10.2.0.0/24`
- `subnet-waf` - `10.2.1.0/24`
- `subnet-apim` - `10.2.2.0/24`

See examples for each subnet below:

Select an address space and configure your subnet. You can customize a default subnet or select from subnet templates if you plan to add select services later. [Learn more](#)

Subnet purpose <small> ⓘ</small>	<input type="text" value="Default"/>
Name <small>* ⓘ</small>	<input type="text" value="subnet-aks"/>
IPv4	
Include an IPv4 address space	<input checked="" type="checkbox"/>
IPv4 address range <small>* ⓘ</small>	<input type="text" value="10.2.0.0/16"/> 10.2.0.0 - 10.2.255.255
Starting address <small>* ⓘ</small>	<input type="text" value="10.2.0.0"/>
Size <small> ⓘ</small>	<input type="text" value="/24 (256 addresses)"/>
Subnet address range <small> ⓘ</small>	10.2.0.0 - 10.2.0.255

Select an address space and configure your subnet. You can customize a default subnet or select from subnet templates if you plan to add select services later. [Learn more](#)

Subnet purpose <small> ⓘ</small>	<input type="text" value="Default"/>
Name <small>* ⓘ</small>	<input type="text" value="subnet-waf"/>
IPv4	
Include an IPv4 address space	<input checked="" type="checkbox"/>
IPv4 address range <small>* ⓘ</small>	<input type="text" value="10.2.0.0/16"/> 10.2.0.0 - 10.2.255.255
Starting address <small>* ⓘ</small>	<input type="text" value="10.2.1.0"/>
Size <small> ⓘ</small>	<input type="text" value="/24 (256 addresses)"/>
Subnet address range <small> ⓘ</small>	10.2.1.0 - 10.2.1.255

Select an address space and configure your subnet. You can customize a default subnet or select from subnet templates if you plan to add select services later. [Learn more](#)

Subnet purpose <small> ⓘ</small>	<input type="text" value="Default"/>
Name <small>* ⓘ</small>	<input type="text" value="subnet-apim"/>
IPv4	
Include an IPv4 address space	<input checked="" type="checkbox"/>
IPv4 address range <small>* ⓘ</small>	<input type="text" value="10.2.0.0/16"/> 10.2.0.0 - 10.2.255.255
Starting address <small>* ⓘ</small>	<input type="text" value="10.2.2.0"/>
Size <small> ⓘ</small>	<input type="text" value="/24 (256 addresses)"/>
Subnet address range <small> ⓘ</small>	10.2.2.0 - 10.2.2.255

Home > Virtual networks >

Create virtual network ...

Basics Security IP addresses Tags **Review + create**

[View automation template](#)

Basics

Subscription	nv-Omniverse-Nucleus
Resource Group	rg-elitang
Name	elitang-vnet
Region	East US

Security

Azure Bastion	Disabled
Azure Firewall	Disabled
Azure DDoS Network Protection	Disabled

IP addresses

Address space	10.2.0.0/16 (65,536 addresses)
Subnet	subnet-aks (10.2.0.0/24) (256 addresses)
Subnet	subnet-waf (10.2.1.0/24) (256 addresses)
Subnet	subnet-apim (10.2.2.0/24) (256 addresses)

Tags

Home >

elitang-vnet-1723071805170 | Overview

Deployment

Search X < Delete Cancel Redeploy Download Refresh

Overview

- Your deployment is complete**
- Deployment name : elitang-vnet-1723071805170 Start time : 8/7/2024, 4:03:30 PM
- Subscription : nv-Omniverse-Nucleus Correlation ID : 5a57e9ba-6387-4459-a7ce-05...
- Resource group : rg-elitang

Deployment details

Next steps

[Go to resource](#)

Give feedback
[Tell us about your experience with deployment](#)

Go to resource. We will need this later.

3.4 Create Network Security Group

[Home](#) > [Network security groups](#) >

Create network security group

Basics Tags Review + create

Project details

Subscription * nv-Omniverse-Nucleus

Resource group * rg-elitang

Create new

Instance details

Name * elitang-nsg

Region * East US

Create the following inbound rules:

- Add new rules
 - Port 80 (http)
 - **Source:** IP Addresses
 - **Source:** ideally your specific VPN CIDR blocks, also `10.0.0.0/8`
 - **Source Port Ranges:** *
 - **Destination:** Any
 - **Service:** HTTP
 - **Protocol:** TCP
 - Port 443 (https)
 - **Source:** IP Addresses
 - **Source:** ideally your specific VPN CIDR blocks, also `10.0.0.0/8`
 - **Source Port Ranges:** *
 - **Destination:** Any
 - **Service:** HTTPS
 - **Protocol:** TCP
 - Port 3443 (APIM management)
 - **Source:** Service Tag
 - **Source service tag:** ApiManagement

- **Source port ranges:** *
- **Destination:** Service Tag
- **Destination service tag:** VirtualNetwork
- **Service:** Custom
- **Protocol:** TCP
- **Destination port:** 3443
- Ports 31000-31002 (streaming)
 - **Source:** IP Addresses
 - **Source:** ideally your specific VPN CIDR blocks, also 10.0.0.0/8
 - **Source Port Ranges:** *
 - **Destination:** Any
 - **Service:** Custom
 - **Destination Port Ranges:** 31000-31002
 - **Protocol:** TCP and UDP (need a separate rule for each)

See examples for each rule below:

Add inbound security rule

elitang-nsg

Source IP Addresses Source IP addresses/CIDR ranges * Any Custom Service Destination port ranges * Any Protocol UDP TCP ICMPv4 Action Allow Deny Priority * 140 Name * AllowCidrBlockCustom31000-31002InboundUDP Description

You should see 5 rules when finished.

Home > Microsoft.NetworkSecurityGroup-20240814095303 | Overview > elitang-nsg

elitang-nsg | Inbound security rules Network security group

+ Add Hide default rules Refresh Delete Give feedback

Network security group security rules are evaluated by priority using the combination of source, source port, destination, destination port, and protocol to allow or deny the traffic. A security rule can't have the same priority and direction as an existing rule. You can't delete default security rules, but you can override them with rules that have a higher priority. [Learn more](#)

Priority ↑↓	Name ↑↓	Port ↑↓	Protocol ↑↓	Source ↑↓	Destination ↑↓	Action ↑↓
<input type="checkbox"/> 100	AllowCidrBlockHTTPIn...	80	TCP	10.0.0.0/8,216.228.125...	Any	<input checked="" type="radio"/> Allow
<input type="checkbox"/> 110	AllowCidrBlockHTTPSi...	443	TCP	10.0.0.0/8,216.228.125...	Any	<input checked="" type="radio"/> Allow
<input type="checkbox"/> 120	AllowTagCustom3443I...	3443	TCP	ApiManagement	VirtualNetwork	<input checked="" type="radio"/> Allow
<input type="checkbox"/> 130	AllowCidrBlockCusto...	31000-31002	TCP	10.0.0.0/8,216.228.125...	Any	<input checked="" type="radio"/> Allow
<input type="checkbox"/> 140	AllowCidrBlockCusto...	31000-31002	UDP	10.0.0.0/8,216.228.125...	Any	<input checked="" type="radio"/> Allow
<input type="checkbox"/> 65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	<input checked="" type="radio"/> Allow
<input type="checkbox"/> 65001	AllowAzureLoadBalanc...	Any	Any	AzureLoadBalancer	Any	<input checked="" type="radio"/> Allow
<input type="checkbox"/> 65500	DenyAllInBound	Any	Any	Any	Any	<input checked="" type="radio"/> Deny

3.4.1 Assign to subnet-aks and subnet-apim

- Click Settings > Subnets
- Click Associate and select both.

The screenshot shows the Azure portal interface for managing a Network Security Group (elitang-nsg). The left sidebar includes options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings (Inbound security rules, Outbound security rules, Network interfaces), and Subnets. The main content area displays a table of subnets:

Name	Address range	Virtual network
subnet-aks	10.2.0.0/24	elitang-vnet
subnet-apim	10.2.2.0/24	elitang-vnet

3.5 DNS/Certificates

3.5.1 Create private DNS zone

- Add **Private DNS Zones** from the marketplace if not already added
 - **Name** is the name of the zone: `elitang-ovas-streaming.net`
 - Once created, under DNS Management > Virtual Network Links, link to vnet created in step 1

Example screenshots below:

The screenshot shows the Azure portal interface for creating a Private DNS Zone. The top navigation bar includes Home > Private DNS zones > Create Private DNS Zone.

Project details

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

Subscription * (nv-Omniverse-Nucleus)

Resource group * (rg-elitang)

Instance details

Name * (elitang-ovas-streaming.net)

Resource group location * (US) East US

Create and click go to resource when ready.

The screenshot shows the Azure portal's deployment overview for the resource group 'rg-elitang'. The deployment name is 'elitang-ovas-streaming.net_1723074105960'. The status message says 'Your deployment is complete'. Deployment details include a start time of 8/7/2024, 4:41:49 PM, a subscription of 'nv-Omniverse-Nucleus', and a correlation ID of 2cf47381-7e45-46b1-8af1-4462c7... . There are links for 'Deployment details' and 'Next steps', and a 'Go to resource' button.

Add a virtual network link

The screenshot shows the 'Add Virtual Network Link' configuration page. It includes fields for 'Link name' (set to 'elitang-vnet-link'), 'Virtual network details' (with a note about supported Resource Manager deployment models), and 'Configuration' (with an 'Enable auto registration' checkbox). The 'Subscription' dropdown is set to 'nv-Omniverse-Nucleus' and the 'Virtual Network' dropdown is set to 'elitang-vnet (rg-elitang)'.

3.5.2 Create self-signed certificates for private DNS zone

Use following Powershell script to generate both a root and SSL certificate:

1. Execute the following. Make changes to the CN

```
a. # Create the root signing cert
$root = New-SelfSignedCertificate -Type Custom -KeySpec Signature ` 
-Subject "CN=ovas-streaming-net-signing-root" -KeyExportPolicy Exportable ` 
-HashAlgorithm sha256 -KeyLength 4096 ` 
-CertStoreLocation "Cert:\CurrentUser\My" -KeyUsageProperty Sign ` 
-KeyUsage CertSign -NotAfter (get-date).AddYears(5)
```

2. Execute the following command. Make changes to the -DnsName. Make changes to the CN.

```
a. # Create the wildcard SSL cert.
$ssl = New-SelfSignedCertificate -Type Custom -DnsName "*.elitang-ovas-` 
streaming.net","elitang-ovas-streaming.net" ` 
-KeySpec Signature ` 
-Subject "CN=*.ovas-streaming.net" -KeyExportPolicy Exportable ` 
-HashAlgorithm sha256 -KeyLength 2048 ` 
-CertStoreLocation "Cert:\CurrentUser\My" ` 
-Signer $root
```

3. Execute the following commands to create the two certificates. These will be stored in the directory you are currently in.

```
a. # Export CER of the root and SSL certs
Export-Certificate -Type CERT -Cert $root -FilePath .\ovas-streaming-` 
signing-root.cer
Export-Certificate -Type CERT -Cert $ssl -FilePath .\ovas-streaming-` 
ssl.cer
```

4. Execute the following commands to create the two certificates. These will be stored in the directory you are currently in. NOTE: Password is required.

```
a. # Export PFX of the root and SSL certs
Export-PfxCertificate -Cert $root -FilePath .\ovas-streaming-signing-` 
root.pfx ` 
-Password (read-host -AsSecureString -Prompt "password")
```

```
b. Export-PfxCertificate -Cert $ssl -FilePath .\ovas-streaming-ssl.pfx ` 
-ChainOption BuildChain -Password (read-host -AsSecureString ` 
-Prompt "password")
```

3.5.3 Create LetsEncrypt certificates manually for public DNS zone

Create LetsEncrypt certificates manually for public DNS zone <https://certbot.eff.org/>

Change the domains to your public DNS.

```
sudo certbot certonly \
--manual \
--preferred-challenges=dns \
--email user@nvidia.com \
--server https://acme-v02.api.letsencrypt.org/directory \
--agree-tos \
-d elitang-ovas-streaming.nvsbx.omniverse.nvidia.com \
-d '*.elitang-ovas-streaming.nvsbx.omniverse.nvidia.com'
```

- This process requires access to the zone in order to manually create and verify TXT records
- The certificates will be created in `/etc/letsencrypt/live/ovas-streaming.nvsbx.omniverse.nvidia.com`⁷
- Example output

```
elitang@elitang-mlt ~ % sudo certbot certonly \
--manual \
--preferred-challenges=dns \
--email user@nvidia.com \
--server https://acme-v02.api.letsencrypt.org/directory \
--agree-tos \
-d elitang-ovas-streaming.nvsbx.omniverse.nvidia.com \
-d '*.elitang-ovas-streaming.nvsbx.omniverse.nvidia.com'
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Requesting a certificate for elitang-ovas-streaming.nvsbx.omniverse.nvidia.com
and *.elitang-ovas-streaming.nvsbx.omniverse.nvidia.com
-----
Please deploy a DNS TXT record under the name:
_acme-challenge.elitang-ovas-streaming.nvsbx.omniverse.nvidia.com.

with the following value:
s8oWc1fH-6S0rpXLjzHJcoIkN0jejAiSOuUlb_UWkJ4

Before continuing, verify the TXT record has been deployed. Depending on the
DNS
```

⁷ <http://ovas-streaming.nvsbx.omniverse.nvidia.com>

provider, **this** may take some time, from a few seconds to multiple minutes. You can check **if** it has finished deploying with aid of online tools, such as the Google Admin Toolbox: https://toolbox.googleapps.com/apps/dig/#TXT/_acme-challenge.elitang-ovas-streaming.nvsbx.omniverse.nvidia.com. Look **for** one or more bolded line(s) below the line '**;ANSWER**'. It should show the value(s) you've just added.

- Add the record in Recordsets

The screenshot shows the 'Records' section of the DNS management interface. A new record is being created for the zone '_acme-challenge.elitang-ovas-streaming'. The record type is set to 'TXT', with a TTL of 1 hour. The value is set to 's8oWc1H-650rpXlJzHjcolkNOjejAiSOUUlb_UWkj4'. The 'Value' field also contains the text 'The quick brown fox jumps over the lazy dog.' Below the record details, there is a 'Metadata' section with two empty input fields for 'Key' and 'Value'.

- Verify that the record show up with the link provided in the command output above:

Name
_acme-challenge.elitang-ovas-streaming.nvsbx.omniverse.nvidia.com

A	AAAA	ANY	CAA	CNAME	DNSKEY	DS	MX	NS	PTR	SOA	SRV	TLSA	TSIG	TXT
---	------	-----	-----	-------	--------	----	----	----	-----	-----	-----	------	------	------------

TXT

TTL:
59 minutes 58 seconds

VALUE:
"s8oWc1fH-650rpXLjzHJcoIkN0jejAiS0uUlB_UWkJ4"

 Raw View

```
id 18045
opcode QUERY
rcode NOERROR
flags QR RD RA
;QUESTION
_acme-challenge.elitang-ovas-streaming.nvsbx.omniverse.nvidia.com. IN TXT
;ANSWER
_acme-challenge.elitang-ovas-streaming.nvsbx.omniverse.nvidia.com. 3598 IN TXT "s8oWc1fH-650rpXLjzHJcoIkN0jejAiS0uUlB_UWkJ4"
;AUTHORITY
;ADDITIONAL
```

- Once you have verified that the TXT entry shows up press Enter

Press Enter to Continue

Successfully received certificate.
 Certificate is saved at: /etc/letsencrypt/live/elitang-ovas-streaming.nvsbx.omniverse.nvidia.com/fullchain.pem
 Key is saved at: /etc/letsencrypt/live/elitang-ovas-streaming.nvsbx.omniverse.nvidia.com/privkey.pem
 This certificate expires on 2024-11-07.
 These files will be updated when the certificate renews.

NEXT STEPS:

- This certificate will not be renewed automatically. Autorenewal of --manual certificates requires the use of an authentication hook script (--manual-auth-hook) but one was not provided. To renew **this** certificate, repeat **this** same certbot command before the certificate's expiry date.

If you like Certbot, please consider supporting our work by:

- * Donating to ISRG / Let's Encrypt: <https://letsencrypt.org/donate>
- * Donating to EFF: <https://eff.org/donate-le>

elitang@elitang-mlt ~ %

If this fails, try running the command again or waiting longer.

3.5.4 Create a .pfx from the full chain certificate and private key

Note: When creating this certificate, a password is required.

```
openssl pkcs12 -export -in /etc/letsencrypt/live/elitang-ovas-
streaming.nvsbx.omniverse.nvidia.com/fullchain.pem -inkey /etc/letsencrypt/live/
elitang-ovas-streaming.nvsbx.omniverse.nvidia.com/privkey.pem -out
nvsbx.omniverse.nvidia.com.pfx
```

Your file will be named nvsbx.omniverse.nvidia.com⁸.pfx. Change the value after `-out` to name it differently. Save the location of this file, we will need to upload it to Azure later.

3.6 AKS Cluster

Create new AKS cluster

3.6.1 Basics

- Name
 - `aks-elitang`
- Authentication
 - Microsoft Entra ID with Azure RBAC

Example:

⁸ <http://nvsbx.omniverse.nvidia.com>

Home > Kubernetes services >

Create Kubernetes cluster ...

[Basics](#) [Node pools](#) [Networking](#) [Integrations](#) [Monitoring](#) [Advanced](#) [Tags](#) [Review + create](#)

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

Subscription * [nv-Omniverse-Nucleus](#)

Resource group * [rg-elitang](#) [Create new](#)

Cluster details
Cluster preset configuration * [Dev/Test](#)

To quickly customize your Kubernetes cluster, choose one of the preset configurations above. You can modify these configurations at any time. [Compare presets](#)

Kubernetes cluster name * [aks-elitang](#)

Region * [\(US\) East US](#)

Availability zones [None](#)

AKS pricing tier [Free](#)

Kubernetes version * [1.29.7 \(default\)](#)

Automatic upgrade [Enabled with patch \(recommended\)](#)

Automatic upgrade scheduler
Start on: Fri Aug 09 2024 00:00 +00:00 (Coordinated Universal Time) [Edit schedule](#)

Node security channel type [Node Image](#)

Security channel scheduler
Start on: Fri Aug 09 2024 00:00 +00:00 (Coordinated Universal Time) [Edit schedule](#)

Choose between local accounts or Microsoft Entra ID for authentication and Azure RBAC or Kubernetes RBAC for your authorization needs.

Authentication and Authorization [Microsoft Entra ID authentication with Azure RBAC](#)

This RBAC configuration only applies to data actions on the Kubernetes API and not to actions on the Azure Resource Manager representation of the AKS cluster. [Learn more](#)

3.6.2 Nodepools

- `agentpool` (Default)
 - `Mode system` (optional)
 - `Standard_D2s_v3`
 - Manual scaling
 - 2-3 nodes
 - `Max Pods per node: 30`
 - Note: The name of this node will be applied as a label in AKS. So if you leave it as the default, a label of `agentpool=agentpool` will be created. If you change the name to `agentpool2`, the label will be `agentpool=agentpool2`. Remember this label later, during the OVAS Stack deployment steps.

Example:

Home > Kubernetes services > Create Kubernetes cluster > Update node pool

Node pool name * agentpool

Mode * System

OS SKU * Ubuntu Linux

Availability zones None

Enable Azure Spot instances

Node size * Standard D8s v3

Scale method * Manual

Node count * 2

Optional settings

Max pods per node * 30

Labels

Taints

Update Cancel https://portal.azure.com/#

- cache
 - Standard_D8s_v3
 - Manual scaling
 - 1 node
 - Label: NodeGroup=cache
 - Max Pods per node: 30

Example:

Update node pool

aks-eltang

Node pool name * Cache

Mode * User

OS SKU * Ubuntu Linux

Availability zones None

Enable Azure Spot instances

Node size * Standard D8s v3
8 vcpus, 32 GiB memory

Scale method Manual

Node count * 1

Optional settings

Max pods per node * 30

Labels
Labels are key/value pairs that can be used to categorize or add identifying information to Kubernetes resources such as nodes.
Labels for the node pool will be applied to each node in the node pool. [Learn more](#)

Key	Value
NodeGroup	cache

Taints
Taints are tuples that are used in conjunction with tolerations to determine which pods can be scheduled on which nodes. In order for a pod to be scheduled to a node, it must tolerate all of the taints applied to that node. Taints for the node pool will be applied to each node in the node pool. [Learn more](#)

Key	Value	Effect
gpu	NV36ads_A10_v5	<input checked="" type="checkbox"/>

Update **Cancel**

- gpu
 - NV36ads_A10_v5
 - Manual scaling
 - 1 node
 - Label: NodeGroup=gpu
 - Max Pods per node: 30

Example:

Home > Kubernetes services > Create Kubernetes cluster >

Update node pool

aks-elitang

Node pool name * gpu

Mode * User

OS SKU * Ubuntu Linux

Availability zones None

Enable Azure Spot instances

Node size * Standard NV36ads A10 v5
36 vcpus, 440 GiB memory
Choose a size

Scale method Manual

Node count * 1

Optional settings

Max pods per node * 30

Labels
Labels are key/value pairs that can be used to categorize or add identifying information to Kubernetes resources such as nodes.
Labels for the node pool will be applied to each node in the node pool. [Learn more](#)

Key	Value
NodeGroup	gpu

Taints
Taints are tuples that are used in conjunction with tolerations to determine which pods can be scheduled on which nodes. In order for a pod to be scheduled to a node, it must tolerate all of the taints applied to that node. Taints for the node pool will be applied to each node in the node pool. [Learn more](#)

Key	Value	Effect

Update **Cancel**

3.6.3 Networking

- **Container configuration:** Azure CNI Node Subnet
- Check Bring your own Azure virtual network
 - Choose vnet created in step 1
 - It may autofill with something similar to (New) rg-elitang-vnet, do not select this one, search for your vnet created before.
 - Choose subnet-aks subnet

Home > Kubernetes services >

Create Kubernetes cluster

[Basics](#) [Node pools](#) [Networking](#) [Integrations](#) [Monitoring](#) [Advanced](#) [Tags](#) [Review + create](#)

Enable a private cluster to restrict worker node to API access, enhancing your Kubernetes workload's security and isolation.

Enable private cluster

Public access

Set authorized IP ranges

Container networking

Network configuration

- Azure CNI Overlay
Assigns pod IP addresses from a private IP space. Best for scalability
- Azure CNI Node Subnet
Previously named Azure CNI. Assigns pod IP addresses from your host VNet. Best for workloads where pods must be reachable by other VNet resources
- Kubelet
Older route-table-based Overlay with limited scalability. Not recommended for most clusters
- High pod values may quickly exhaust available IP addresses. [Learn more](#) (if)

Bring your own Azure virtual network

Virtual network elitang-vnet
[Create new](#)

Cluster subnet * subnet-aks (10.2.0.0/24)
[Manage subnet configuration](#)

Kubernetes service address range * 10.0.0/16

Kubernetes DNS service IP address * 10.0.0.10

DNS name prefix * aks-elitang-dns

Network policy *

- None
Allow all ingress and egress traffic to the pods
- Calico
Open-source networking solution. Best for large-scale deployments with strict security requirements
- Azure
Native networking solution. Best for simpler deployments with basic security and networking requirements

Load balancer Standard

•

3.6.4 Review and deploy:

Go to resource when completed.

3.7 Post-deployment

3.7.1 Accessing Kubernetes (RBAC Required to access cluster resources)

This is required to access the AKS cluster and manage resources inside Kubernetes. Add any user here that might need to access the cluster.

- RBAC
 - Click Access control (IAM)
 - Add self to `Azure Kubernetes Service RBAC Cluster Admin`
 - Add self to `Azure Kubernetes Service RBAC Admin`
 - These are done one at a time

Find role:

Home > microsoft.aks-1723155894305 | Overview > aks-elitang | Access control (IAM) >

Add role assignment ...

Role **Members** **Conditions** **Review + assign**

A role definition is a collection of permissions. You can use the built-in roles or you can create your own custom roles. [Learn more](#)

Job function roles **Privileged administrator roles**

Grant access to Azure resources based on job function, such as the ability to create virtual machines.

Name ↑↓	Description ↑↓	Type ↑↓	Category ↑↓	Details
Azure Kubernetes Service RBAC Admin	Lets you manage all resources under cluster/namespace, except update or delete resource quotas an...	BuiltInRole	Containers	View
Azure Kubernetes Service RBAC Cluster Admin	Lets you manage all resources in the cluster.	BuiltInRole	Containers	View

Showing 1 - 2 of 2 results.

- Find your name:

Home > microsoft.aks-1723155894305 | Overview > aks-elitang | Access control (IAM) >

Add role assignment ...

Role **Members** **Conditions** **Review + assign**

Selected role Azure Kubernetes Service RBAC Admin

Assign access to User, group, or service principal Managed identity

Members [+ Select members](#)

Name	Object ID	Type
Elizabeth Tang	4ae7792c-08ee-4ca0-ba6d-a6d68908de...	User

Description

- Review and Assign both roles.

3.7.2 Pull kubeconfig locally and check access

Log in to the `az` CLI with `az login`

Ensure you are in the correct subscription with `az account set --subscription <subscription-id>`

Install kubelogin: <https://azure.github.io/kubelogin/install.html>

Run the following commands:

```

elitang@elitang-mlt ~ % az aks get-credentials --format azure --resource-group rg-elitang --name aks-elitang
Merged "aks-elitang" as current context in /Users/elitang/.kube/config
elitang@elitang-mlt ~ % export KUBECONFIG=/Users/elitang/.kube/config
elitang@elitang-mlt ~ % kubelogin convert-kubeconfig
elitang@elitang-mlt ~ % kubectl get nodes
To sign in, use a web browser to open the page https://microsoft.com/devicelogin and
enter the code BV2FPJZ5E to authenticate.
No resources found in default namespace.
elitang@elitang-mlt ~ % kubectl get nodes
NAME                      STATUS   ROLES      AGE    VERSION
aks-agentpool-35741025-vmss000000  Ready    <none>    20m    v1.29.7
aks-agentpool-35741025-vmss000001  Ready    <none>    20m    v1.29.7
aks-cache-35741025-vmss000000    Ready    <none>    20m    v1.29.7
aks-gpu-35741025-vmss000000     Ready    <none>    19m    v1.29.7

```

Check that the nodes you created appear.

3.8 API Management (APIM) Service

3.8.1 Create the base APIM service:

- **Org name:** nvidia
- **Admin email:** username@nvidia.com⁹
- **Pricing tier:** Developer

Example:

⁹ mailto:username@nvidia.com

Home > API Management services >

Create API Management service

API Management service

Basics Monitor + secure Virtual network Managed identity Tags Review + install

Project details

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

Subscription * ⓘ nv-Omniverse-Nucleus

Resource group * ⓘ rg-elitang Create new

Instance details

Region * ⓘ (US) East US

Resource name * ⓘ APIM-elitang

Organization name * ⓘ nvidia

Administrator email * ⓘ elitang@nvidia.com

Pricing tier

API Management pricing tiers vary in computing capacity per unit and the offered feature set - for example, support for virtual networks, multi-regional deployments, or self-hosted gateways. To accommodate more API requests, consider adding API Management service units instead. [Learn more](#)

⚠️ The Developer tier of API Management does not include SLA and should not be used for production purposes. Your service may experience intermittent outages, for example during upgrades. [Learn more](#)

Pricing tier * ⓘ Developer (no SLA) View all pricing tiers

Virtual Network

- **Connectivity type:** Virtual Network
- **Type:** Internal
- **Vnet:** (created in step 1)
- **Subnet:** subnet-apim

Example:

Home > API Management services >

Create API Management service

API Management service

Basics Monitor + secure **Virtual network** Managed identity Tags Review + install

Network connectivity

You can connect to your Azure API Management instance either publicly, via public IP addresses or service endpoints, or privately, using a private endpoint.

Connectivity type None Virtual network Private endpoint

Virtual network

ℹ️ Securely access resources available in or through your Azure Virtual Network.

Type External Internal

Configure virtual networks

Virtual network * ⓘ elitang-vnet Create new

Subnets * ⓘ subnet-apim (undefined)

Public IP address ⓘ

Review and install

Note: The APIM service can take up to an hour to deploy

3.8.2 Post Deployment

1. Once deployed, add a custom domain:

- Select `Custom Domains` under `Deployment and Infrastructure`
- Add a new custom domain for the gateway:
 - **Type:** Gateway
 - **Hostname:** `apim-gw.elitang-ovas-streaming.net`¹⁰ ← This should be apim-gw.<your private DNS>
 - **Certificate:** Custom
 - **Certificate file:** `ovas-streaming-ssl.pfx`
 - Check `Default SSL binding`
- Add a new custom domain for the management gateway:
 - **Type:** Management
 - **Hostname:** `apim-mgmt.elitang-ovas-streaming.net`¹¹ ← This should be apim-mgmt.<your private DNS>
 - **Certificate:** Custom
 - **Certificate file:** `ovas-streaming-ssl.pfx`

Note: These settings can take up to an hour to apply

2. Add DNS records to private DNS zone

- Add two records for `apim-gw` and `apim-mgmt` to previously created private DNS zone pointing to private IP of the APIM instance created above
- Example:

¹⁰ <http://apim-gw.ovas-streaming.net>

¹¹ <http://apim-mgmt.ovas-streaming.net>

3.9 Application Gateway (with WAF)

3.9.1 Basics

- Tier:** WAFv2
- Autoscaling:** (optional) min 2, max 3
- Policy:** Create new (can use defaults)
- Vnet:** Created in step 1
- Subnet:** subnet-waf

3.9.2 Frontends

- **Type:** both
- **Public IP:** Create new
- **Private IP:** Choose IP in private subnet range (e.g. 10.2.1.10)

Home > Load balancing | Application Gateway >
Create application gateway ...

✓ Basics ② **Frontends** ③ Backends ④ Configuration ⑤ Tags ⑥ Review + create

Traffic enters the application gateway via its front-end IP address(es). An application gateway can use a public IP address, private IP address, or one of each type.¹²

Frontend IP address type: Public Private Both

Public IP address
Public IPv4 address *: (New) elitang-pub
[Add new](#)

Private IP address
Private IPv4 address *: 10.2.1.10

3.9.3 Backends

- Create new
- **Name:** apim
- **Backend target:** apim-gw.elitang-ovas-streaming.net¹² ← This should be apim-gw.<your private DNS>

Home > Load balancing | Application Gateway >
Create application gateway ...

✓ Basics ✓ Frontends ③ **Backends** ④ Configuration ⑤ Tags ⑥ Review + create

A backend pool is a collection of resources to which your application gateway can send traffic. A backend pool can contain virtual machines, virtual machine scale sets, app services, IP addresses, or fully qualified domain names (FQDN).¹²

Add a backend pool

Backend pool	Targets
apim-backend-pool	> 1 target

Add a backend pool.

A backend pool is a collection of resources to which your application gateway can send traffic. A backend pool can contain virtual machines, virtual machine scale sets, app services, IP addresses, or fully qualified domain names (FQDN).¹²

Name *: apim-backend-pool
Add backend pool without targets: Yes No

Backend targets: 1 item

Target type	Target
IP address or FQDN	apim-gw.elitang-ovas-streaming.net
IP address or FQDN	

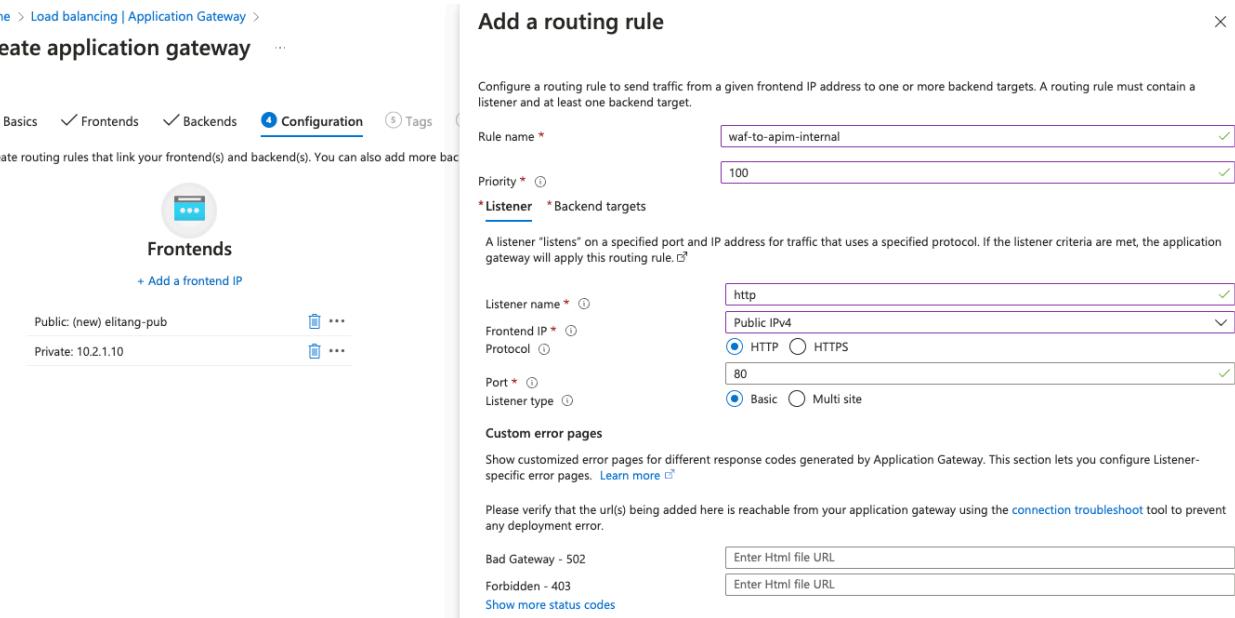
3.9.4 Configuration

- Add new routing rule

12 <http://apim-gw.ovas-streaming.net>

- **Name:** waf-to-apim-internal
- Priority 100
- Listener
 - **Name:** http
 - **Frontend IP:** public IPv4
 - **Protocol:** http
 - **Port:** 80

Example:



The screenshot shows the Azure portal interface for creating an Application Gateway. On the left, there's a navigation bar with 'Home > Load balancing | Application Gateway >' and a 'Create application gateway' button. Below this, there are tabs for 'Basics', 'Frontends', 'Backends', 'Configuration' (which is selected), and 'Tags'. A note says 'Create routing rules that link your frontend(s) and backend(s). You can also add more backends later.' On the right, a 'Frontends' section lists 'Public: (new) elitang-pub' and 'Private: 10.2.1.10'. A 'Listeners' section is partially visible. A large central window titled 'Add a routing rule' is open, showing the configuration for the 'waf-to-apim-internal' rule. It includes fields for 'Priority' (100), 'Listener' (http), 'Frontend IP' (Public IPv4), 'Protocol' (HTTP), 'Port' (80), and 'Listener type' (Basic). There are also sections for 'Custom error pages' and 'Show more status codes'.

- Backend targets

- Target:** apim (backend created above)
- Create new backend setting
 - Name:** https-internal
 - Protocol:** https
 - Server certificate is signed by well-known authority:** No
 - CER certificate:** ovas-streaming-signing-root.cer
 - Override with new host name:** Yes
 - Select Pick hostname from backend target
 - Create custom probes:** Yes

- Example

Add Backend setting X

[← Discard changes and go back to routing rules](#)

Backend settings name *	<input type="text" value="https-internal"/> ✓
Backend protocol	<input type="radio"/> HTTP <input checked="" type="radio"/> HTTPS
Backend port *	<input type="text" value="443"/> ✓
Backend server's certificate is issued by a well-known CA	<input type="radio"/> Yes <input checked="" type="radio"/> No
Upload Root CA certificate	
<p>! You must upload the Root certificate (CER) of the backend server to this Backend Setting, if a Private CA has issued that certificate or if it is a self-generated one. This root certificate allows your application gateway to complete the certificate chain validation.</p> <ul style="list-style-type: none"> To extract the backend server's Root certificate, follow the troubleshooting guide. You can also create your own Root CA and Server certificates. Learn more. 	
CER certificate *	<input type="text" value="ovas-streaming-signing-root.cer"/> ✓
Additional settings	
Cookie-based affinity	<input type="radio"/> Enable <input checked="" type="radio"/> Disable
Connection draining	<input type="radio"/> Enable <input checked="" type="radio"/> Disable
Request time-out (seconds) *	<input type="text" value="20"/>
Override backend path	<input type="text"/>
Host name	
<p>By default, the Application Gateway sends the same HTTP host header to the backend as it receives from the client. If your backend application/service requires a specific host value, you can override it using this setting.</p>	
Override with new host name	<input checked="" type="radio"/> Yes <input type="radio"/> No
<p>! If the backend service is a multi-tenant Azure service such as App Services, Functions, or Portal Apps, we recommend using Custom domain method, instead of overriding the hostname. Using override host name with default domains (azurewebsites.net, azurermicroservices.io, etc.) is good only for the basic tests and operations.</p>	
Host name override	<input type="radio"/> Pick host name from backend target <input checked="" type="radio"/> Override with specific domain name
Create custom probes	<input checked="" type="radio"/> Yes <input type="radio"/> No

3.9.5 Once deployed go to app gateway:

3.9.5.1 Health Probe

Click Settings > Health Probe. Click the test and check that it works.

- **Name:** https
- **Pick hostname from backend settings:** Yes
- **Path:** /status-0123456789abcdef
- **Use probe matching conditions:** Yes
- **Backend settings:** choose existing one
- Test the probe; it should be successful
- Click Add

3.9.5.2 HTTPS Listener

Add a new HTTPS listener (optional; adds TLS termination at AppGw)

- Under **Settings > Listeners**, click **+ Add Listener**
- **Name:** https

- **Frontend IP:** public
- **Protocol:** https
- **Certificate:** Upload a certificate
- **Name:** ovas-streaming
- **PFX Certificate File:** (.pfx file created earlier)
- **Password:**
- **Listener type:** Multi site
- **Host type:** Multiple/Wildcard
- **Host names:**
 - elitang-ovas-streaming.nvsbx.omniverse.nvidia.com¹³
 - *.elitang-ovas-streaming.nvsbx.omniverse.nvidia.com¹⁴

The screenshot shows the 'Add listener' dialog box. Key fields filled in:

- Protocol: https
- Frontend IP: Public
- Port: 443
- Cert name: ovas-streaming
- PFX certificate file: "nvsbx.omniverse.nvidia.com.pfx"
- Host names: elitang-ovas-streaming.nvsbx.omniverse.nvidia.com, *.elitang-ovas-streaming.nvsbx.omniverse.nvidia.com

- If you get an error like this:

13 <http://ovas-streaming.nvsbx.omniverse.nvidia.com>

14 <http://ovas-streaming.nvsbx.omniverse.nvidia.com>

! Failed to save application gateway changes

Failed to save configuration changes to application gateway 'appgw-elitang'. Error: Data or KeyVaultSecretId must be specified for certificate 'rg-elitang/providers/Microsoft.Network/applicationGateways/elitang/sslCertificates/elitang'>appgw-elitang/elitang'. If you are providing certificate data, please upload the certificate again, otherwise please remove the certificate and all references. Subsequent operations will continue to fail with this error message until a valid certificate replaces the existing certificate or the certificate is removed. For secrets/certificates referenced by key vault, please ensure connectivity to key vault is possible and the referenced secret/certificate has not been disabled.

- Change the permissions on the pfx file. An example on Mac/Linux to allow any access to file:

```
elitang@elitang-mlt ~ % sudo chmod 777 nvsbx.omniverse.nvidia.com.pfx
```

3.9.5.3 Routing Rules

- Under `Settings > Rules`, click `+ Routing rule`
 - **Name:** https
 - **Priority:** 10
 - **Listener:** https

Add a routing rule

appgw-elitang

Configure a routing rule to send traffic from a given frontend IP address to one or more backend targets. A routing rule must contain a listener and at least one backend target.

Rule name *	https
Priority *	10
*Listener	https

A listener "listens" on a specified port and IP address for traffic that uses a specified protocol. If the listener criteria are met, the application gateway will apply this routing rule.

- Listener *
- Backend pool
 - **Backend target:** apim
 - **Backend settings:** https-internal

Add a routing rule

Configure a routing rule to send traffic from a given frontend IP address to one or more backend targets. A routing rule must contain a listener and at least one backend target.

Rule name *	https		
Priority *	10		
* Listener	Backend targets		
Choose a backend pool to which this routing rule will send traffic. You will also need to specify a set of Backend settings that define the behavior of the routing rule. ?			
Target type	<input checked="" type="radio"/> Backend pool <input type="radio"/> Redirection		
Backend target *	apim-backend-pool		
Backend settings *	https-internal		
Path-based routing			
You can route traffic from this rule's listener to different backend targets based on the URL path of the request. You can also apply a different set of Backend settings based on the URL path. ?			
Path based rules			
Path	Target name	Backend setting name	Backend pool
No additional targets to display			
Add multiple targets to create a path-based rule			

- Under **Settings > Rules**, click **waf-to-apim-internal**
 - **Backend targets**
 - Change **Target type** from **Backend pool** to **Redirection**
 - **Target listener:** https

waf-to-apim-internal

Configure a routing rule to send traffic from a given frontend IP address to one or more backend targets. A routing rule must contain a listener and at least one backend target.

Rule name	waf-to-apim-internal
Priority *	100
* Listener	Backend targets
Choose a backend pool to which this routing rule will send traffic. You will also need to specify a set of Backend settings that define the behavior of the routing rule. ?	
Target type	<input type="radio"/> Backend pool <input checked="" type="radio"/> Redirection
Redirection type	Permanent
Redirection target	<input checked="" type="radio"/> Listener <input type="radio"/> External site
Target listener *	https
Include query string	<input checked="" type="radio"/> Yes <input type="radio"/> No
Include path	<input checked="" type="radio"/> Yes <input type="radio"/> No

3.9.6 Post Deployment:

If your APIM service is not yet finished deploying with post deployment steps, you will see an error on the AppGW with something like "Unhealthy backend pools". Once AppGW is deployed, you should no longer have this error. Try testing the health probe if this problem continues and make sure that the URLs you added are configured correctly.

3.10 RBAC

1. Assign RBAC permissions to enterprise app registration created by AKS cluster

- Find the vnet created in step 1
- Under `Access control (IAM)`, select the `Network Contributor` role
 - Search for the cluster name under `Managed Identities` and add the managed identity of the AKS cluster

Select managed identities

Some results might be hidden due to your ABAC condition.

Subscription *

nv-Omniverse-Nucleus



Managed identity

All system-assigned managed identities (19)



Select

elitang

4 Kubernetes Services

4.1 Deploy base helm charts

Clone this repository <https://gitlab-master.nvidia.com/omniverse/farm/infrastructure/azure/43083d15-7273-40c1-b7db-39efd9ccc17a/6db1f0c2-73fc-434c-9ebe-7fc0dbcc87a9/ovas-streaming-example#aks-cluster>

Or, a more customized repo: <https://gitlab-master.nvidia.com/wwfo-sae-omniverse/cloud-testing/aws/team-rdarbha/-/tree/azure>

```
git clone https://gitlab-master.nvidia.com/wwfo-sae-omniverse/cloud-testing/aws/team-rdarbha.git
git checkout azure
```

If you do not have access to this, a zip is provided.

4.1.1 Set environment-specific values

- At a minimum, the following values need to be changed to suit your environment:
- Note: Instructions for this are specified in the following steps.
 - `helm/nginx-ingress-controller/values-internal.yaml`
 - `service.beta.kubernetes.io/azure-load-balancer-resource-group`¹⁵ :
name of resource group ex: `rg-elitang`
 - `helm/ov-ov-svc-applications/values.yaml`
 - `host` : This is an arbitrary internal hostname of your choosing used to access the swagger web interface for the application service
 - `streaming.serviceConfig.backend_csp_args.base_domain` : This is the public DNS name at which to access the APIs
 - `helm/ov-ov-svc-streaming/values.yaml`
 - `host` : This is an arbitrary internal hostname of your choosing used to access the swagger web interface for the streaming service

¹⁵ <http://service.beta.kubernetes.io/azure-load-balancer-resource-group>

4.1.2 Internal ingress controller (helm/nginx-ingress-controller)

Check values file; make sure resource group is correct in annotations. File located at `helm/nginx-ingress-controller/values-internal.yaml`

```
...
  service.beta.kubernetes.io/azure-load-balancer-resource-group: rg-elitang
...
```

```
helm repo add bitnami https://charts.bitnami.com/bitnami
helm repo update
helm upgrade -i nginx-ingress-controller-internal -n nginx-ingress-controller --create-namespace -f helm/nginx-ingress-controller/values-internal.yaml bitnami/nginx-ingress-controller
```

- Ensure the Service of type `LoadBalancer` is provisioned with a private external IP (i.e. does not say `<Pending>`; check output of `kubectl get svc -A`)
 - This private IP should be within the range of the `subnet-aks` subnet! If it's not, double-check that the cluster was deployed within your own vnet and not a managed one (see AKS instructions above)

kubectl get svc -n nginx-ingress-controller			
NAME	CLUSTER-IP	EXTERNAL-IP	PORT(S)
nginx-ingress-controller-internal	4.20	10.2.0.120	80:30619/TCP,443:31307/TCP
			LoadBalancer 10.0.3 23h

4.1.3 FluxCD (helm/flux2)

In your flux values file, change `value: system` to `value: <name of your agentpool>`.

You can find the labels by executing `kubectl get nodes --show-labels | grep agentpool` and looking for the label value.

Example:

```
elmController:
  create: true
  affinity:
    nodeAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
```

```

nodeSelectorTerms:
  - matchExpressions:
    - key: agentpool
      operator: In
      values:
        - agentpool    <-- this should be <name of your agentpool>
  create: false
kustomizeController:
  create: false
notificationController:
  create: false
imageReflectionController:
  create: false
policies:
  create: false
rbac:
  create: true
sourceController:
  create: true
affinity:
  nodeAffinity:
    requiredDuringSchedulingIgnoredDuringExecution:
      nodeSelectorTerms:
        - matchExpressions:
          - key: agentpool
            operator: In
            values:
              - agentpool      <-- this should be <name of your agentpool>

```

```

helm repo add fluxcd-community https://fluxcd-community.github.io/helm-charts
helm repo update
helm upgrade --install fluxcd -n omni-system --create-namespace -f helm/flux2/
values.yaml fluxcd-community/flux2

```

4.1.4 GPU Operator (helm/gpu-operator)

```

helm repo add nvidia https://helm.ngc.nvidia.com/nvidia
helm repo update
helm upgrade -i gpu-operator -n gpu-operator --create-namespace -f helm/gpu-
operator/values.yaml nvidia/gpu-operator

```

4.1.5 Memcached (helm/memcached)

```
helm upgrade -i memcached-service-r3 -n omni-streaming --create-namespace bitnami/
memcached --version 7.0.2 -f helm/memcached/values.yml
```

4.1.6 ExternalDNS (scripts/external-dns)

Create a service principal and assign the correct roles via the first script. Edit the `scripts/external-dns/01-create-sp-for-rbac.sh` file with your own:

```
SUBSCRIPTION_ID="YOUR_SUBSCRIPTION_ID"
EXTERNALDNS_NEW_SP_NAME="NvSbxExternalDnsServicePrincipal" # name of the service
principal (This should be unique)
AZURE_DNS_ZONE_RESOURCE_GROUP="fidot_sandbox" # name of resource group where dns zone
is hosted
AZURE_DNS_ZONE="nvsbx.omniverse.nvidia.com" # DNS zone name like example.com or
sub.example.com
```

Execute

```
./scripts/external-dns/01-create-sp-for-rbac.sh
```

Example output:

```
elitang@elitang-mlt ovas-streaming-example % ./scripts/external-dns/01-create-sp-for-
rbac.sh
WARNING: The output includes credentials that you must protect. Be sure that you do
not include these credentials in your code or check the credentials into your source
control. For more information, see https://aka.ms/azadsp-cli
Client ID: <CLIENT ID HERE>
Client secret: <CLIENT SECRET HERE>
...
...
Copy azure.json.template to azure.json and add the above client ID, secret, resource
group and subscription ID
```

Create `azure.json` file with new credentials (note resource group may differ from your cluster RG):

```
{
  "tenantId": "<Your-tenant-ID>",
  "subscriptionId": "<your-subscription-id>",

}
```

```

    "resourceGroup": "<dns-zone-rg>",
    "aadClientId": "<client-id>",
    "aadClientSecret": "<client-secret>"
}

```

Apply

```
kubectl create secret generic azure-config-file --namespace "default" --from-file ./scripts/external-dns/azure.json
```

Edit `scripts/external-dns/03-external-dns-manifest.yaml` edit appropriate values for `--domain-filter` and `--azure-resource-group`

```

...
spec:
  serviceAccountName: external-dns
  containers:
    - name: external-dns
      image: registry.k8s.io/external-dns/external-dns:v0.14.2
      args:
        - --source=service
        - --source=ingress
        - --domain-filter=nvsbx.omniverse.nvidia.com <-- your Public domain here
        - --provider=azure
        - --azure-resource-group=<dns resource group>
        - --txt-prefix=externaldns-
      volumeMounts:
        - name: azure-config-file
          mountPath: /etc/kubernetes
          readOnly: true
...

```

Apply the external-dns manifest

```
kubectl apply -f scripts/external-dns/03-external-dns-manifest.yaml
```

4.1.7 Create required secrets

Get your NGC API token (you must have access to NGC's `nvidian/omniverse` org)

```
export NGC_API_TOKEN=<your-token>
```

Create the `regcred` secret

```
kubectl create secret -n omni-streaming docker-registry regcred \
--docker-server=nvcr.io \
```

```
--docker-username='$oauthtoken' \
--docker-password=$NGC_API_TOKEN \
--dry-run=client -o json | \
kubectl apply -f -
```

Create the nge-omni-user secret

```
kubectl create secret generic -n omni-streaming nge-omni-user \
--from-literal=username='$oauthtoken' \
--from-literal=password="$NGC_API_TOKEN"
```

Add the required NVIDIA helm repositories

```
helm repo add omniverse https://helm.ngc.nvidia.com/nvidia/omniverse --
username='$oauthtoken' --password=$NGC_API_TOKEN
helm repo update
```

4.2 Deploy OVAS services

4.2.1 Streaming (helm/kit-appstreaming-manager)

Check values file and update DNS names accordingly

```
...
ingress:
  host: api.elitang-ovas-streaming.net.      <--- your private domain here ex:
  api.<your-private-domain>
    className: internal-nginx
...
...
```

i To enable/disable WSS follow these directions

```
...
  backend_csp_cls: "nv.svc.streaming._csp.Generic"
  backend_csp_args:
    enable_wss: true      <----- set to true to enable WSS. set to false to
    disable WSS
```

```
base_domain: "ovas-streaming.nvsbx.omniverse.nvidia.com" <--- add
your public domain name here (Leave blank if disabling WSS)
...
```

```
helm upgrade --install \
--namespace omni-streaming \
-f helm/kit-appstreaming-manager/values.yaml --version 1.0.0-ea \
streaming omniverse/kit-appstreaming-manager
```

4.2.2 Applications (helm/kit-appstreaming-applications)

Check values file and update DNS names accordingly

```
...
ingress:
  host: api.elitang-ovas-streaming.net    <--- your private domain here ex:
api.<your-private-domain>
...
```

```
helm upgrade --install \
--namespace omni-streaming \
-f helm/kit-appstreaming-applications/values.yaml --version 1.0.0-ea \
applications omniverse/kit-appstreaming-applications
```

4.2.3 RMCP (helm/kit-appstreaming-rmcp)

Check values file and update DNS names accordingly

Change or comment out affinity from `system` to `agentpool`

```
affinity:
  nodeAffinity:
    requiredDuringSchedulingIgnoredDuringExecution:
      nodeSelectorTerms:
        - matchExpressions:
          - key: agentpool
            operator: In
            values:
              - agentpool <-- this should be <name of your agentpool>
```

```
helm upgrade --install \
--namespace omni-streaming \
-f helm/kit-appstreaming-rmcp/values.yaml --version 1.0.0-ea \
rmcp omniverse/kit-appstreaming-rmcp
```

4.2.4 Deploy the custom streaming resources (manifests/omniverse/azure)



1. To enable WSS: If there is already a file at `manifests/omniverse/azure/application-profile-wss.yaml`, there is no need to create the file. If there is not, create it.
2. Then run `kubectl apply -n omni-streaming -f application-profile-wss.yaml`

```
apiVersion: omniverse.nvidia.com/v1
kind: ApplicationProfile
metadata:
  name: azurelb
spec:
  name: AzureLB example profile
  description: Default profile - uses an AzureLB per stream
  supportedApplications:
    - name: usd-viewer
      versions:
        - '*'
  chartMappings:
    container: streamingKit.image.repository
    container_version: streamingKit.image.tag
    name: streamingKit.name
  chartValues:
    global:
      imagePullSecrets:
        - name: regcred
  streamingKit:
    envoy:
      enabled: true
      secretRef: stream-tls-secret
      config: |
        static_resources:
          listeners:
            - name: webrtc_signaling_listener
              address:
```

```

        socket_address:
          address: 0.0.0.0
          port_value: 49200
      filter_chains:
      - transport_socket:
          name: envoy.transport_sockets.tls
          typed_config:
            "@type": type.googleapis.com/
envoy.extensions.transport_sockets.tls.v3.DownstreamTlsContext
            common_tls_context:
              tls_certificates:
              - certificate_chain: { filename: "/etc/envoy/tls/
tls.crt" }
              private_key: { filename: "/etc/envoy/tls/tls.key" }
        }
        filters:
        - name: envoy.filters.network.tcp_proxy
          typed_config:
            "@type": type.googleapis.com/
envoy.extensions.filters.network.tcp_proxy.v3.TcpProxy
            stat_prefix: tcp
            cluster: service_cluster
            access_log:
            - name: envoy.access_loggers.stream
              typed_config:
                "@type": type.googleapis.com/
envoy.extensions.access_loggers.stream.v3.StdoutAccessLog
                log_format:
                  text_format: "[%START_TIME%] \"%PROTOCOL%\"
connection from %DOWNSTREAM_REMOTE_ADDRESS% to %UPSTREAM_HOST%\n"
            - name: health_listener
              address:
                socket_address:
                  address: 0.0.0.0
                  port_value: 8080
            filter_chains:
            - filters:
              - name: envoy.filters.network.http_connection_manager
                typed_config:
                  "@type": type.googleapis.com/
envoy.extensions.filters.network.http_connection_manager.v3.HttpConnectionM
anager
                  stat_prefix: health_check
                  codec_type: AUTO
                  route_config:
                    name: local_route
                    virtual_hosts:
                    - name: local_service
                      domains: ["*"]
                      routes:
                      - match:

```

```

        prefix: "/health"
        direct_response:
            status: 200
            body:
                inline_string: "OK"
        http_filters:
            - name: envoy.filters.http.router
clusters:
    - name: service_cluster
        connect_timeout: 0.25s
        type: STATIC
        lb_policy: ROUND_ROBIN
        load_assignment:
            cluster_name: service_cluster
            endpoints:
                - lb_endpoints:
                    - endpoint:
                        address:
                            socket_address:
                                address: 127.0.0.1
                                port_value: 49100 # Forwarding to the stream
tls:
    enabled: true
    secretRef: stream-tls-secret
image:
    repository: nvcr.io/omniverse/prerel/usd-viewer
    pullPolicy: Always
    tag: 0.2.0
sessionId: session_id
service:
    signalingPort: 31000
    mediaPort: 31001
    healthPort: 31002
annotations:
    # NLB configuration
    service.beta.kubernetes.io/azure-load-balancer-internal: "false"
    external-dns.alpha.kubernetes.io/ttl: "5"
    # Security stuff - may need to specify this in
`loadBalancerSourceRanges`
    # service.beta.kubernetes.io/load-balancer-source-ranges: 216.228.1
25.128/30,216.228.127.128/30,206.223.160.26/32,216.228.125.131/32,216.228.1
25.128/30,216.228.127.128/30
    # Health check
    service.beta.kubernetes.io/port_8080_health-probe_protocol: HTTP
    service.beta.kubernetes.io/port_8080_health-probe_port: "8080"
    service.beta.kubernetes.io/port_8080_health-probe_request-path: /
health
    type: LoadBalancer
    name: kit-app
    resources:
        limits:

```

```

cpu: "3"
memory: 20Gi
nvidia.com/gpu: "1"
requests:
  nvidia.com/gpu: "1"
env:
  - name: USD_PATH
    value: /app/data/Forklift_A/Forklift_A01_PR_V_NVD_01.usd

```

- i** To disable WSS, make the following changes in `manifests/omniverse/azure/application-profile-azurelb.yaml`

```

envoy:
  tls:
    enabled: true <---- set this to false
    secretRef: stream-tls-secret

```

Then run

```
kubectl apply -n omni-streaming -f application-profile-azurelb.yaml
```

```
kubectl apply -n omni-streaming -f application.yaml
kubectl apply -n omni-streaming -f application-version.yaml
```

4.2.5 Deploy HelmRepository (`manifests/helm-repositories`)

```
kubectl apply -n omni-streaming -f manifests/helm-repositories/ngc-omniverse.yaml
```

This should (eventually) show READY: True in the output of `kubectl get helmrepositories -n omni-streaming:`

```
> kubectl get helmrepositories -n omni-streaming
NAMESPACE      NAME          URL                           AGE
READY   STATUS
omni-streaming  ngc-omniverse  https://helm.ngc.nvidia.com/nvidia/omniverse  13m
True    stored artifact: revision
'sha256:5766710ed3052d8a247ea8aeb4d093bc9395bd82afbfaeda41546af2a22d850'
```

4.2.6 Create private DNS record for ingress controller

Go to the Private DNS Zone you created. Create the following recordset:

```
api.elitang-ovas-streaming.net16 -> private external ip of ingress controller LB
service (e.g. 10.2.0.120 shown below)
```

```
elitang@elitang-mlt ovas-streaming-example % kubectl get svc -n nginx-ingress-controller
NAME                                     TYPE        CLUSTER-IP
EXTERNAL-IP     PORT(S)           AGE
nginx-ingress-controller-internal          LoadBalancer  10.0.100.71
10.2.0.120    80:32710/TCP,443:31053/TCP  13h
nginx-ingress-controller-internal-default-backend ClusterIP  10.0.166.232
<none>        80/TCP            13h
elitang@elitang-mlt ovas-streaming-example %
```

Example of recordset to create:

The screenshot shows the Azure portal interface for creating a DNS record. The top navigation bar has 'api' selected. Below it, the domain 'elitang-ovas-streaming.net' is chosen. A modal window titled 'Users' is open, showing a single record set entry. The 'Name' field contains 'api.elitang-ovas-streaming.net'. The 'Type' field is set to 'A'. The 'TTL *' field is set to '1'. The 'TTL unit' dropdown is set to 'Hours'. Under the 'IP address' section, there are two input fields: the first contains '10.2.0.120' and the second contains '0.0.0.0'. There is also a small trash icon next to the first IP address field.

4.2.7 Create public DNS entry for App Gateway

- Navigate to the Public DNS Zone
- Create the following RecordSet
 - elitang-ovas-streaming.nvsbx.omniverse.nvidia.com
 - Points to IP address of APIGW public IP
 - To find this navigate to the APIGW
 - Should match the value of `streaming.serviceConfig.backend_csp_args.base_domain` in the streaming service specified previously.

¹⁶ <http://api.ovas-streaming.net>

Add record set

nvsbx.omniverse.nvidia.com

Name	<input type="text" value="elitang-ovas-streaming"/>	.nvsbx.omniverse.nvidia.com
Type	<input type="text" value="A – Address record"/>	
Alias record set ⓘ	<input type="text" value="No"/>	
TTL *	<input type="text" value="1"/>	
TTL unit	<input type="text" value="Hours"/>	
IP address	<input type="text" value="20.253.91.113"/> <input type="text" value="0.0.0.0"/>	

4.2.8 Add API route to ingress controller in APIM

Navigate to APIs in the APIM

Home > APIM-elitang | APIs

APIM-elitang | APIs

API Management service

Search Developer portal Send us your feedback

Overview Activity log Access control (IAM) Tags Diagnose and solve problems Events Settings Workspaces APIs Products

Define a new API

+ Add API All APIs Echo API ...

HTTP Manually define an HTTP API

WebSocket Streaming, full-duplex communication with a WebSocket server

GraphQL Access the full capabilities of your data from a single endpoint.

gRPC High performance, universal Remote Procedure Call framework

Create from definition

- Create a new **HTTP API**

Create an HTTP API

Basic Full

* Display name

* Name

Web service URL

API URL suffix

Base URL

Create **Cancel**

- Create new `GET` operation
 - URL: `/*`
 - Under Responses tab, add `200`

The screenshot shows the APIM-eliteang API Management service interface. On the left, there's a navigation sidebar with options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Events, Settings, APIs, Workspaces, and APIs. The APIs section is currently selected. In the main pane, a 'REVISION 1' card is displayed with the creation date 'CREATED Aug 9, 2024, 11:14:55 AM'. The 'Design' tab is selected, showing the path 'http get > Get > Frontend'. A 'Frontend' operation is being configured with a display name 'Get', name 'get', and URL pattern 'GET /*'. The 'Responses' tab is active, showing a '200 OK' response template placeholder: 'Please select a response'.

- Under Backend > HTTP(s) endpoint, add the DNS name given to the streaming services, `api.elitang-ovas-streaming.net`
 - Check the `Override` box next to `Service URL`
 - Select `HTTP(s) endpoint` instead of `Azure logic app`

The screenshot shows the APIM-elitang API Management service interface. On the left, the navigation menu is visible with options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Events, Settings, APIs, Workspaces, APIs (selected), Products, Subscriptions, Named values, Backends, and Policy fragments. The main panel displays a 'Backend' configuration for the 'http get > Get > Backend' operation. The 'Target' section is set to 'HTTP(s) endpoint' with the URL 'http://api.elitang-ovas-streaming.net'. The 'Service URL' field contains 'http://api.elitang-ovas-streaming.net'. Under 'Gateway credentials', 'None' is selected. The 'Test' tab is active, showing the configuration details.

- Repeat this for POST and DELTE operations. Result below:

The screenshot shows the APIM-elitang API Management service interface. The navigation menu is identical to the previous screenshot. The main panel displays the 'All APIs' list, which includes the 'Echo API' and the 'http get' operation. The 'http get' operation is currently selected. The 'Test' tab is active, showing the configuration details for this specific operation. The 'All operations' section lists 'DEL delete', 'GET Get', and 'POST post' operations.

5 Testing

5.1 Testing From CLI

- From CLI with subscription key required:
 - To get subscription key, go to APIM and navigate to APIs > Subscriptions

The screenshot shows the APIM Management service interface. The left sidebar has a tree view with Home, APIM-elitang selected, and sub-options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Events, Settings, APIs, Workspaces, APIs, Products, Subscriptions (which is selected and highlighted in yellow), and Named values.

Display name	Primary key	Secondary key	Scope	State	Owner	Allow tracing
Product Starter	XXXXXXXXXX	XXXXXXXXXX	Product Starter	Active	Administrator	<input checked="" type="checkbox"/>
Product Unlimited	XXXXXXXXXX	XXXXXXXXXX	Product Unlimited	Active	Administrator	<input checked="" type="checkbox"/>
Built-in all-access su...	XXXXXXXXXX	XXXXXXXXXX	Service	Active		<input checked="" type="checkbox"/>

- Right click on the row that has Built in all-access subscription...
- Click Show/hide keys
- Copy primary key

```
curl -H 'Ocp-Apim-Subscription-Key: <your-key>' https://appgw.ovas-streaming.nvsbx.omniverse.nvidia.com/streaming/docs`  
curl -H 'Ocp-Apim-Subscription-Key: <your-key>' https://appgw.ovas-streaming.nvsbx.omniverse.nvidia.com/applications/docs
```

- Example Output:

```
elitang@elitang-mlt ~ % curl -H 'Ocp-Apim-Subscription-Key: <KEY>'  
https://elitang-ovas-streaming.nvsbx.omniverse.nvidia.com/streaming/  
docs  
  
<!DOCTYPE html>  
<html>  
<head>  
<link type="text/css" rel="stylesheet" href="https://  
cdn.jsdelivr.net/npm/swagger-ui-dist@5.9.0/swagger-ui.css">
```

```

<link rel="shortcut icon" href="https://fastapi.tiangolo.com/img/favicon.png">
<title>Application Streaming API – Swagger UI</title>
</head>
<body>
<div id="swagger-ui">
</div>
<script src="https://cdn.jsdelivr.net/npm/swagger-ui-dist@5.9.0/dist/swagger-ui-bundle.js"></script>
<!-- `SwaggerUIBundle` is now available on the page --&gt;
&lt;script&gt;
const ui = SwaggerUIBundle({
    url: '/streaming/openapi.json',
    "dom_id": "#swagger-ui",
    "layout": "BaseLayout",
    "deepLinking": true,
    "showExtensions": true,
    "showCommonExtensions": true,
    oauth2RedirectUrl: window.location.origin + '/streaming/docs/oauth2-redirect',
    presets: [
        SwaggerUIBundle.presets.apis,
        SwaggerUIBundle.SwaggerUIStandalonePreset
    ],
})
&lt;/script&gt;
&lt;/body&gt;
&lt;/html&gt;
%
elitang@elitang-mlt ~ %
</pre>

```

5.2 Testing with swagger docs

5.2.1 Disabling Subscription Key

From browser (requires disabling subscription key without a service in front to acquire one): navigate to above URLs and confirm you can get to swagger docs, run commands, etc.

Go to APIM, APIs > APIs > http get > settings. Uncheck the box that says `subscription required`

Swagger docs for Streaming and Application can be accessed here:

Edit with your public domain.

<https://elitang-ovas-streaming.nvsbx.omniverse.nvidia.com/streaming/docs>¹⁷

<https://elitang-ovas-streaming.nvsbx.omniverse.nvidia.com/application/docs>¹⁸

5.2.2 Creating a stream

Go to streaming swagger, ex: <https://elitang-ovas-streaming.nvsbx.omniverse.nvidia.com/streaming/docs>¹⁹

Click the POST /Stream

Click Post, then click "Try it out"

Enter the following as the request body:

```
{
  "id": "usd-viewer",
  "profile": "azurelb",
  "version": "0.2.0"
}
```

¹⁷ https://elitang-ovas-streaming.nvsbx.omniverse.nvidia.com/streaming/docs#/Streaming%20APIs/start_stream_stream_post

¹⁸ https://elitang-ovas-streaming.nvsbx.omniverse.nvidia.com/streaming/docs#/Streaming%20APIs/start_stream_stream_post

POST /stream Create a streaming session

Create a streaming session using the provided metadata information.

Parameters

No parameters

Request body required

```
{
  "id": "usd-viewer",
  "profile": "azuriteb",
  "version": "0.2.0"
}
```

Execute

Once executed, you should be able to see a pod being created with `kubectl get pods -A`

Additionally, you can use the Get /stream operation on the swagger page and see that the stream is created.

5.2.3 Verification

Checking that external DNS created entries in Public DNS Zone

externaldns-a-rfetvw.elitang-ovas-streaming	TXT	300	"heritage=external-dns,external-dns/owner=default,extern al-dns/resource=service/omi-streaming/kit-app-ebfb6863-90bc-42a4-af9-faa88999bc58"		
externaldns-rfetvw.elitang-ovas-streaming	TXT	300	"heritage=external-dns,external-dns/owner=default,extern al-dns/resource=service/omi-streaming/kit-app-ebfb6863-90bc-42a4-af9-faa88999bc58"		

You can also check that a new rule is created on your load balancer.

The screenshot shows two side-by-side Azure management interface pages. The left page is titled 'Load balancing | Load Balancer' under 'kubernetes'. It lists several load balancers: 'apollo-tunnel-bridge-prod02-lb', 'apollo-tunnel-bridge-prod1-lb', 'apollo-tunnel-prod-eu2-lb', 'dummy-lb', and multiple entries for 'kubernetes'. The right page is titled 'kubernetes | Load balancing rules' under 'Load balancer'. It shows a table of load-balancing rules with columns for Name, Protocol, Backend pool, and Health probe. Three rules are listed: one for UDP port 31001 and two for TCP port 31000, all pointing to the 'kubernetes' backend pool.

5.3 Testing with local client app (Locally installed certificates)

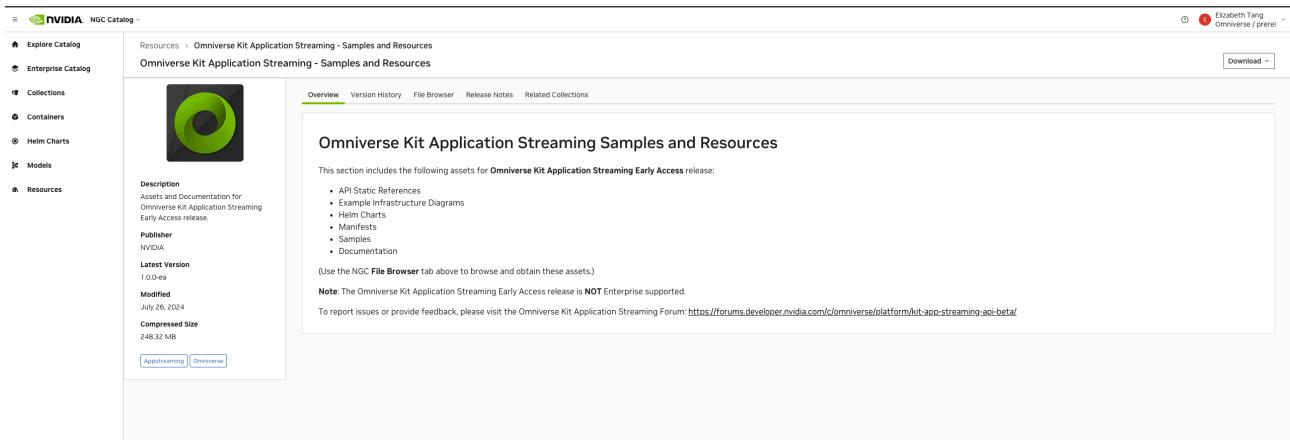
Make sure the the subscription key is disabled in the step above.

Go to NGC and search for streaming

The screenshot shows the NVIDIA NGC Catalog interface. On the left is a sidebar with categories like Explore Catalog, Enterprise Catalog, Collections, Containers, Helm Charts, Models, and Resources. A search bar at the top contains the query 'streaming'. Below the search bar, it says 'Displaying 20 results'. Two items are listed: 'Omniverse Kit Application Streaming - Early Access' and 'Omniverse Kit SDK - Application Streaming Container'. Both items have a green circular icon with a white play button symbol. Below each item is a brief description and 'View Labels' and 'Learn More' buttons.

Click on Omniverse Kit Application Streaming - Early Access

Scroll to the bottom and click on Resources



Click File Browser and click `usd-viewer-streaming-sample.zip` to download this zip. Unzip this.

5.3.1 Client Application On Local Machine

Edit `stream.config.json` with the following file and change endpoints for streaming and applications.

Make sure these are HTTPS endpoints

"appServer": "<https://elitang-ovas-streaming.nvsbx.omniverse.nvidia.com/applications>²⁰", "streamServer": "<https://elitang-ovas-streaming.nvsbx.omniverse.nvidia.com/streaming>²¹"

```
{
  "usd_stage_uri": "/app/data/Forklift_A/Forklift_A01_PR_V_NVD_01.usd",
  "defaultPrimPath": "/World/Geometry",
  "$comment": "source can be either 'stream', 'gfn' or 'local'",
  "source": "stream",

  "stream": {
    "$comment": "Required props if source is set to 'stream'.",
    "applicationId": "usd-viewer",
    "applicationVersion": "0.2.0",
    "appServer": "https://elitang-ovas-streaming.nvsbx.omniverse.nvidia.com/applications",
    "streamServer": "https://elitang-ovas-streaming.nvsbx.omniverse.nvidia.com/streaming"
  },
  "gfn": {
    "$comment": "Required props if source is set to 'gfn'.",
    "catalogClientId": "",
    "clientId": "",
    "cmsId": 0
  },
  "local": {
}}
```

²⁰ <http://elitang-ovas-streaming.nvsbx.omniverse.nvidia.com/applications>

²¹ <http://elitang-ovas-streaming.nvsbx.omniverse.nvidia.com/streaming>

```
        "$comment": "Required props if source is set to 'local'.",
        "server": "127.0.0.1"
    }
}
```

5.3.1.1 Install NPM

Install NPM and specify version (running `apt install npm` may install a version too old or too new)

```
curl -sL https://deb.nodesource.com/setup_18.x22 -o nodesource_setup.sh
sudo bash nodesource_setup.sh
sudo apt install nodejs
```

Install npm (this may take some time)

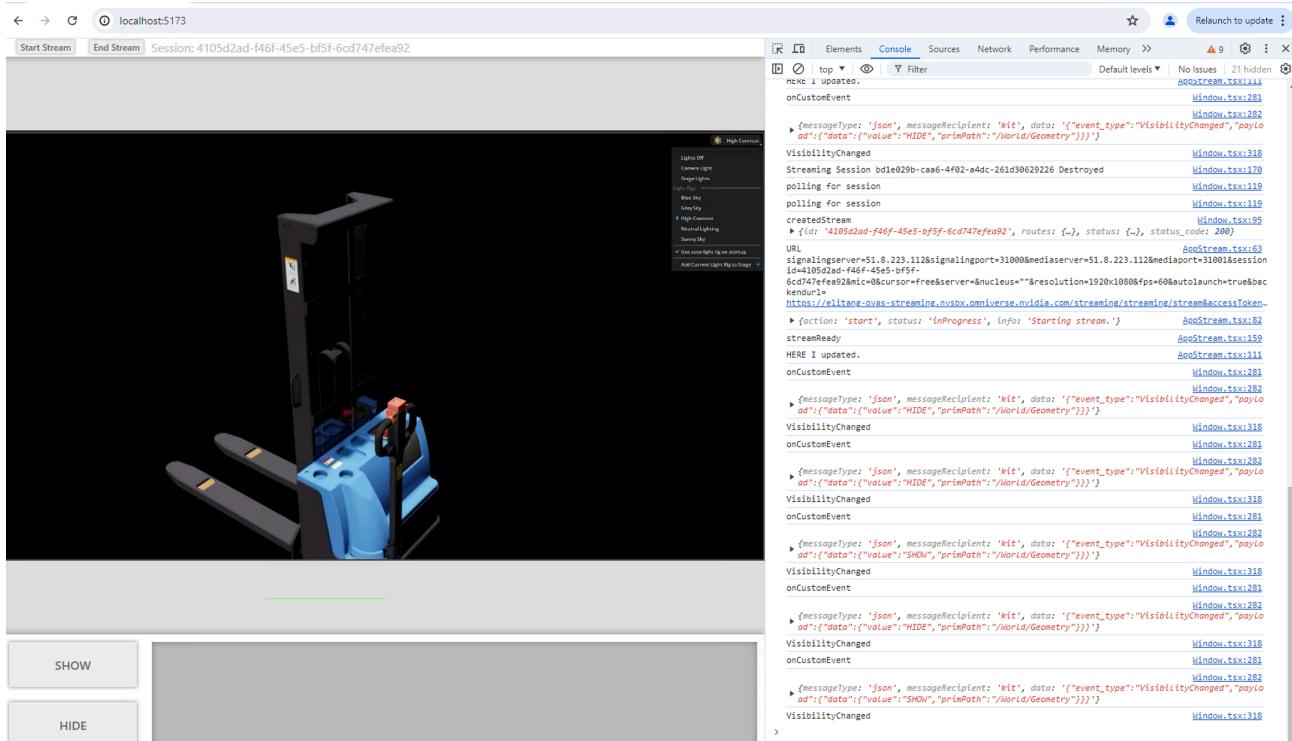
```
npm install
```

```
npm run dev --host
```

If WSS is not enabled, navigate the the URL provided. You should see a forklift load. This may take some time the first stream, as objects need to be loaded into the cache. Wait about 10 minutes. If the stream fails, try once more. Sometimes the stream times out before the objects are loaded. Once you see the forklift, OVAS is fully validated.

If WSS is enabled, the stream will not start, follow steps below.

²² http://deb.nodesource.com/setup_18.x



5.3.2 Enabling WSS:

Download and install Mkcrt: <https://github.com/FiloSottile/mkcrt>

Run the command

```
mkcert -install
```

Create a self signed certificate for the base domain specified in your `nv-ov-svc-streaming/values.yaml`

Locate the domain under `base_domain`. Example:

```
...
backend_csp_cls: "nv.svc.streaming._csp.Generic"
backend_csp_args:
  enable_wss: true
  base_domain: "dev.ovsa.kddccorp.com"
...
```

On your local machine, execute this command

```
mkcert *.dev.ovsa.kddccorp.com
```

Now you can access the stream at <http://localhost:5173/>. You should see in the networking tab, a WSS connection established. Once you see the forklift, OVAS is fully validated with end to end encryption.

You are using an unsupported command-line flag: --disable-web-security. Stability and security will suffer.

Start Stream End Stream Session: c62b55be-0ef7-4ee0-8a15-8f3fdad58e30

localhost:5173

Request Headers

- Request URL: wss://sxzxdb.dev.ovsa.kddccorp.com:31000/sign_in?peer_id=peer-9509302460&version=2&reconnect=1
- Request Method: GET
- Status Code: 101 Switching Protocols

Response Headers

- Connection: Upgrade
- Date: Thu, 09 Aug 2024 17:11:27 GMT
- Sec-WebSocket-Accept: rz/4MEEPu/w0uaXv2YEuCKXPQ=
- Sec-WebSocket-Protocol: x-nv-sessionid.c62b55be-0ef7-4ee0-8a15-8f3fdad58e30
- Upgrade: websocket

Request Headers

- Accept-Encoding: gzip, deflate, br, zstd
- Accept-Language: en-US,en;q=0.9
- Cache-Control: no-cache
- Connection: Upgrade
- Host: sxzxdb.dev.ovsa.kddccorp.com:31000
- Origin: http://localhost:5173
- Pragma: no-cache
- Sec-WebSocket-Extensions: permessage-deflate: client_max_window_bits
- Sec-WebSocket-Key: o7g5OgvrfNw8gNM3Q/GvA==
- Sec-WebSocket-Protocol: x-nv-sessionid.c62b55be-0ef7-4ee0-8a15-8f3fdad58e30
- Sec-WebSocket-Version: 13
- Upgrade: websocket
- User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.0.0 Safari/537.36

68 requests | 12.0 kB transferred

SHOW

HIDE

Path: /World/Geometry/SM_Forklift_Rudder_A01_01
Part ID: PID8296
Name: Rudder_A01_01
URL: https://www.nvidia.com/en-us/omniverse/8296/10_0_2
Version: 10.0.2

6 Client App in Azure

Clone the branch `feat/appstream_noauth_azure_configmap`:https://gitlab-master.nvidia.com/omniverse/samples/web/usd-viewer-streaming-sample/-/tree/feat/appstream_noauth_azure_configmap?ref_type=heads

6.1 Creating a NGINX Ingress Controller

Create a file called `nginx-ingress-external.yaml`

Add the following code to the file:

```

nodeAffinityPreset:
  type: hard
  key: kubernetes.azure.com/mode
  values:
  - system

config:
  proxy-body-size: 0m
  proxy-request-buffering: "off"
  ssl-ciphers: HIGH:!aNULL:!eNULL:!EXPORT:!DES:!RC4:!MD5:!PSK:@SECLEVEL=1
  ssl-protocols: TLSv1.2 TLSv1.3
  use-http2: "false"
metrics:
  enabled: false
  serviceMonitor:
    enabled: false
replicaCount: 2
service:
  annotations:
    service.beta.kubernetes.io/azure-load-balancer-health-probe-request-path: /
healthz
  service.beta.kubernetes.io/azure-load-balancer-resource-group: rg-elitang
  externalTrafficPolicy: Local

```

Apply this file with:

```
helm upgrade -i nginx-ingress-controller -n nginx-ingress-controller --create-namespace -f nginx-ingress-external.yaml bitnami/nginx-ingress-controller
```

This will create a new nginx ingress controller, with an exposed Public IP. This may take a few minutes to come up. Find this IP with:

```
elitang@elitang-mlt usd-viewer-streaming-sample 2 % kubectl get svc -n nginx-ingress-controller
NAME                                TYPE        CLUSTER-IP
EXTERNAL-IP   PORT(S)           AGE
nginx-ingress-controller               LoadBalancer  10.0.132.23
51.8.220.1    80:30761/TCP,443:31981/TCP  3d18h  <----- Copy the external IP
nginx-ingress-controller-default-backend   ClusterIP    10.0.180.252
<none>      80/TCP            3d18h
nginx-ingress-controller-internal       LoadBalancer  10.0.71.172
10.2.0.149    80:32690/TCP,443:32315/TCP  12d
nginx-ingress-controller-internal-default-backend   ClusterIP    10.0.163.197
<none>      80/TCP            12d
```

6.2 Adding Entry to Public DNS Zone

Create a recordset with a new subdomain prefix ex: `viewer.elitang-ovas-streaming`

viewer.elitang-ovas-streaming ×

`nvsbx.omniverse.nvidia.com`

User	
Name	viewer.elitang-ovas-streaming.nvsbx.omniverse.nvidia.com
Type	A
Alias record set ⓘ	No
TTL *	1
TTL unit	Hours
IP address	51.8.220.1 0.0.0.0
Metadata	
Key	Value
<input type="text" value="Enter Key"/>	<input type="text" value="Enter Value"/>

Edit the file: `helm/streaming-sample/internal/azure/dev-b/values.yaml`

```
---
```

```

image:
  repository: nvcr.io/nvidian/omniverse/streaming-sample-viewer           #<----
point to your client app image
  pullPolicy: Always
  # Overrides the image tag whose default is the chart appVersion.
  tag: 1.0.0-beta.3-azure-configmap
imagePullSecrets:
  - name: regcred                                         #<---- In
this tutorial regcred is in the namespace omni-streaming. Therefore to access this
secret, this chart must be deployed with -n omni-streaming

serviceConfig:
  appServerUri: https://elitang-ovas-streaming.nvsbx.omniverse.nvidia.com/
applications          #<---- add your endpoints, these will be added to the
application with a configmap
  streamServerUri: https://elitang-ovas-streaming.nvsbx.omniverse.nvidia.com/
streaming

ingress:
  enabled: true
  className: nginx
  tls:
    - hosts:
        - viewer.elitang-ovas-streaming.nvsbx.omniverse.nvidia.com
          #<---- The public DNS record you created above should match this
          secretName: stream-tls-secret
  hosts:
    - host: viewer.elitang-ovas-streaming.nvsbx.omniverse.nvidia.com
      paths:
        - path: /
          pathType: Prefix
  annotations:
    # Route53 DNS name
    external-dns.alpha.kubernetes.io/ingress-hostname-source: annotation-only
    #external-dns.alpha.kubernetes.io/hostname: viewer.dev-
b.nvsbx.omniverse.nvidia.com
    # Certificate and TSL redirection
    kubernetes.io/ingress.class: nginx

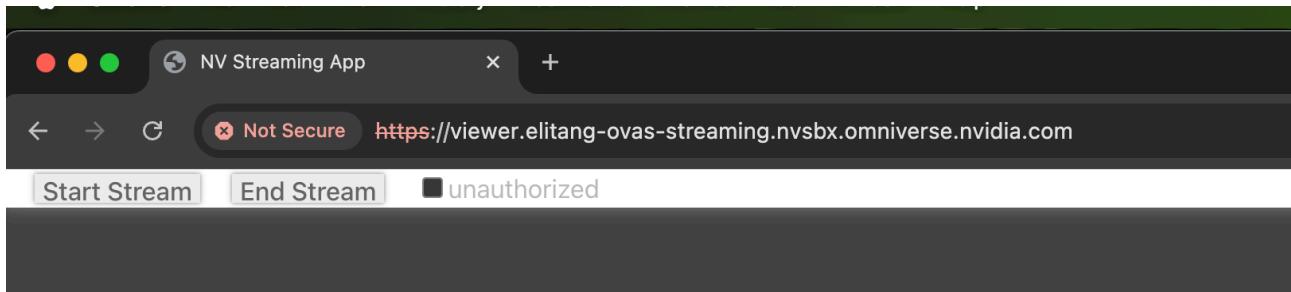
```

Apply this with:

```
helm upgrade -i viewer-sample -f helm/streaming-sample/internal/azure/dev-b/
values.yaml helm/streaming-sample -n omni-streaming
```

If you see that both pods are in a RUNNING state, go to the domain you created. Ex: viewer.elitang-ovas-streaming.nvsbx.omniverse.nvidia.com²³

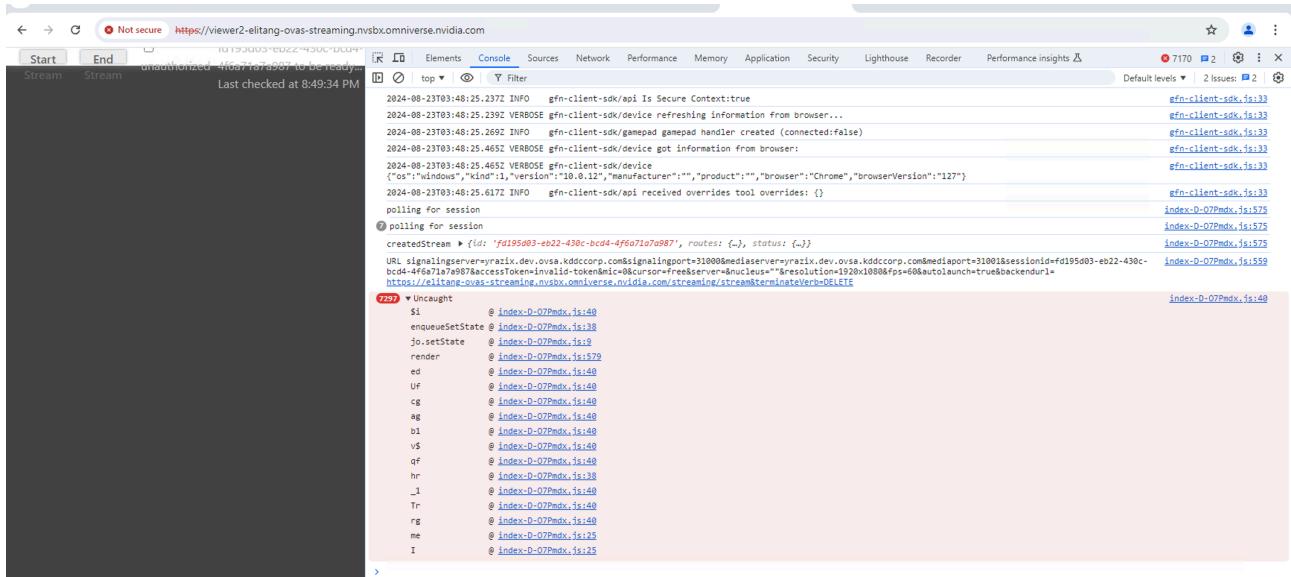
²³ <http://viewer.elitang-ovas-streaming.nvsbx.omniverse.nvidia.com>



If the domain hangs and does not resolve, try troubleshooting with nslookup. Recordsets may not update instantly.

If a NGINX code appears, there may be a problem with your client application.

6.3 Current Bug (Tracked on Jira)



6.4 Changing Application Used

If you need to use a custom application, there may be some changes needed in helm/streaming-sample/templates/configmap.yaml

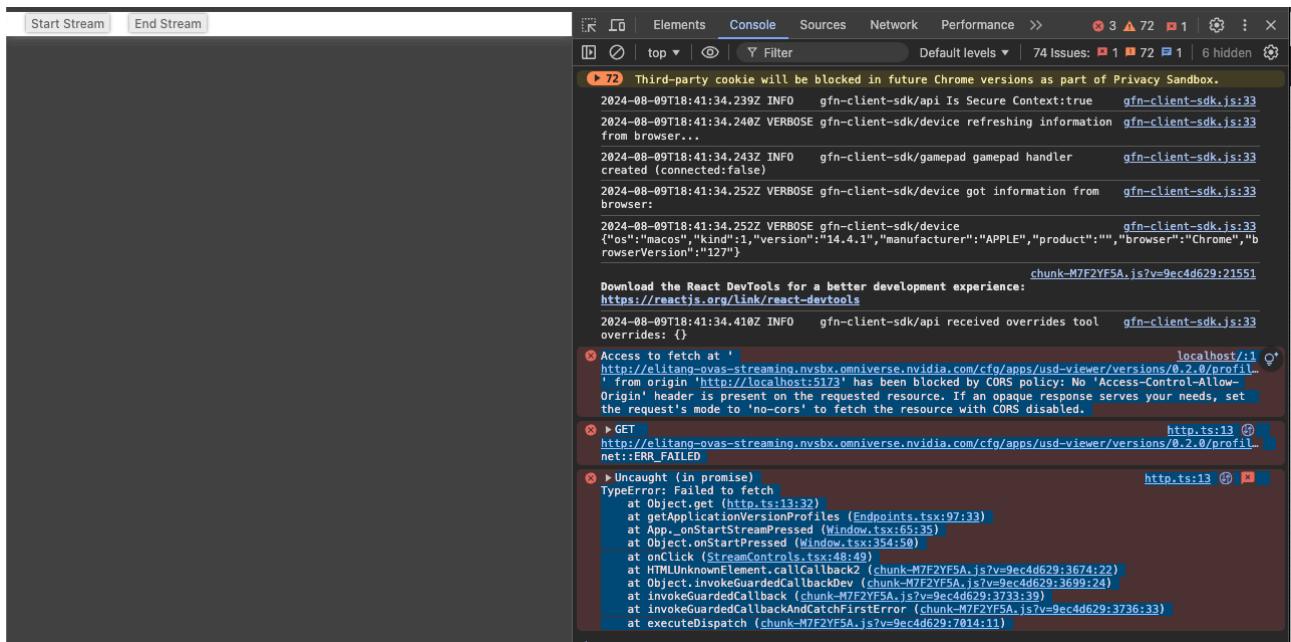
Such as: applicationID, applicationversion, applicatinprofile

7 Deleting resources and environment

Find your resource group and delete the resource group

8 Troubleshooting

8.1 CORS Policy Error



<https://stackoverflow.com/questions/3102819/disable-same-origin-policy-in-chrome>

On Windows:

1. Search for Run in Windows search
2. Type the command `chrome.exe --user-data-dir="C://Chrome dev session" --disable-web-security`
3. This should open a chrome browser. Access the stream from this browser instead.

8.2 NSG Rules

Each tenant may be different and there may be additional NSG rules that must be created. Search for the name of your resources or in NSG on Azure to see if any additional configurations are necessary. If allowed, try opening NSG rules to all temporarily for testing.

8.3 Forbidden access to Kubernetes

If you get an error like

```
dsmarsh@dsmarsh-linux-vm:~$ az aks get-credentials --format azure --resource-group rg-elitang --name aks-elitang
Merged "aks-elitang" as current context in /home/dsmarsh/.kube/config
dsmarsh@dsmarsh-linux-vm:~$ kubelogin convert-kubeconfig
dsmarsh@dsmarsh-linux-vm:~$ kubectl get pods
Error from server (Forbidden): pods is forbidden: User "dsmarsh@nvidia.com" cannot
list resource "pods" in API group "" in the namespace "default": User does not have
access to the resource in Azure. Update role assignment to allow access.
```

Or you see forbidden in the Azure Portal,

The screenshot shows the Azure portal interface for an AKS cluster named 'aks-elitang'. The user is navigating through the 'Namespaces' section under 'Kubernetes resources'. A search bar at the top is empty. Below it, there's a 'Create' button and a 'Refresh' button. A 'Filter by namespace name' input field contains the placeholder 'Enter the full namespace name'. To the right of the input field is a 'Add label filter' link. The main area displays a large 'Forbidden (403)' error message with a cloud icon containing a 'no' symbol. Below the error message, a detailed error description reads: 'namespaces is forbidden: User "dsmarsh@nvidia.com" cannot list resource "namespaces" in API group "" at the cluster scope: User does not have access to the resource in Azure. Update role assignment to allow access.. 'dsmarsh@nvidia.com' does not have the required Kubernetes permissions to view this resource. Ensure you have the correct role/role binding for this user or group.' At the bottom of the error message is a 'Try again' button.

You need to add yourself to the AKS RBAC with these steps:

[AKS-RBAC \(see page 29\)](#)

9 Enabling WSS Troubleshooting (Problem resolving DNS)

One issue in our environment is that there's currently a wildcard DNS entry for `*.nvsbx.omniverse.nvidia.com`²⁴. That means if you do an nslookup for `<anything>.nvsbx...` it'll resolve to that default ip. The problem is because the client first checks to make sure it can resolve the hostname of the kit app pod (that gets created with external-dns), but the wildcard throws that off and (a) starts trying to connect before the actual record is created and (b) resolves to the wildcard ip instead of the ip of the load balancer.

The workaround implemented here is to switch the external-dns configuration to use a different DNS host (route53 for example), replace the stream-tls-secret with certs for that zone, so that the zone that doesn't have a default entry.

If your environment does not have a wildcard DNS entry, this may not be necessary.

Our alternative zone in Route53 is dev.ovsa.kddccorp.com²⁵

Follow the directions from above on creating a certificate for the public DNS zone. Enter the DNS TXT in your zone and obtain the certificate.

9.1 Obtain Public DNS Certificate

```
elitang@elitang-mlt ~ % sudo certbot certonly \
    --manual \
    --preferred-challenges=dns \
    --email user@nvidia.com \
    --server https://acme-v02.api.letsencrypt.org/directory \
    --agree-tos \
    -d dev.ovsa.kddccorp.com \
    -d '*.*.dev.ovsa.kddccorp.com'

Saving debug log to /var/log/letsencrypt/letsencrypt.log
Requesting a certificate for dev.ovsa.kddccorp.com and *.dev.ovsa.kddccorp.com

-----
Please deploy a DNS TXT record under the name:

_acme-challenge.dev.ovsa.kddccorp.com.
```

with the following value:

`qlJq309uMTHIRELNLq7gzBgmrsijIc24EdBFqjmb15I`

Before continuing, verify the TXT record has been deployed. Depending on the DNS provider, **this** may take some time, from a few seconds to multiple minutes. You can check **if** it has finished deploying with aid of online tools, such as the Google

²⁴ <http://nvsbx.omniverse.nvidia.com/>

²⁵ <http://dev.ovsa.kddccorp.com>

```
Admin Toolbox: https://toolbox.googleapps.com/apps/dig/#TXT/_acme-challenge.dev.ovsa.kddccorp.com.
```

Look **for** one or more bolded line(s) below the line '**;ANSWER**'. It should show the value(s) you've just added.

Press Enter to Continue

Successfully received certificate.

Certificate is saved at: /etc/letsencrypt/live/dev.ovsa.kddccorp.com/fullchain.pem

Key is saved at: /etc/letsencrypt/live/dev.ovsa.kddccorp.com/privkey.pem

This certificate expires on 2024-11-12.

These files will be updated when the certificate renews.

NEXT STEPS:

- This certificate will not be renewed automatically. Autorenewal of --manual certificates requires the use of an authentication hook script (--manual-auth-hook) but one was not provided. To renew **this** certificate, repeat **this** same certbot command before the certificate's expiry date.

If you like Certbot, please consider supporting our work by:

- * Donating to ISRG / Let's Encrypt: <https://letsencrypt.org/donate>
- * Donating to EFF: <https://eff.org/donate-le>

elitang@elitang-mlt ~ %

Once this certificate is obtained, create a secret in Azure with the following command, specifying where your certs are located.

```
sudo kubectl create secret tls stream-tls-secret --cert=/etc/letsencrypt/live/dev.ovsa.kddccorp.com/fullchain.pem --key=/etc/letsencrypt/live/dev.ovsa.kddccorp.com/privkey.pem -n omni-streaming
```

9.2 Change ExternalDNS to Use Alternative Zone

Since our alternative zone existed in AWS, we had to reconfigure externalDNS.

First, remove the previous externalDNS

```
kubectl delete -f /scripts/external-dns/03-external-dns-manifest.yaml
```

9.3 Setting up ExternalDNS (Required for TLS)

External DNS was installed through these docs <https://github.com/kubernetes-sigs/external-dns/blob/master/docs/tutorials/aws.md>.

1. Create IAM credentials

a. Create file with the following contents called `policy.json`

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "route53:ChangeResourceRecordSets"
      ],
      "Resource": [
        "arn:aws:route53:::hostedzone/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "route53>ListHostedZones",
        "route53>ListResourceRecordSets",
        "route53>ListTagsForResource"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

b. Apply with

```
aws iam create-policy --policy-name "AllowExternalDNSUpdates" --policy-document file://policy.json

# example: arn:aws:iam::XXXXXXXXXXXXX:policy/AllowExternalDNSUpdates
export POLICY_ARN=$(aws iam list-policies \
--query 'Policies[?PolicyName==`AllowExternalDNSUpdates`].Arn' --output text)
```

Verify that this was applied by looking in the AWS Console:

IAM > Policies > AllowExternalDNSUpdates

AllowExternalDNSUpdates Info

Policy details

Type Customer managed	Creation time June 28, 2024, 16:20 (UTC-04:00)
--------------------------	---

Permissions Entities attached Tags Policy versions (1) Access Advisor

Permissions defined in this policy Info

Permissions defined in this policy document specify which actions are allowed or denied. To define permissions for an IAM identity (user, user group, or role), attach a policy to it.

Allow (1 of 420 services)

Service	Access level	Resource	Request condition
Route 53	Limited: List, Read, Write	Multiple	None

2. Create permissions to modify DNS Zone (static credential steps followed)

a. **9.3.1 Create IAM user and attach the policy**

```
# create IAM user
aws iam create-user --user-name "externaldns"

# attach policy arn created earlier to IAM user
aws iam attach-user-policy --user-name "externaldns" --policy-arn
$POLICY_ARN
```

b. **9.3.2 Create the static credentials**

```
SECRET_ACCESS_KEY=$(aws iam create-access-key --user-name "externaldns")
ACCESS_KEY_ID=$(echo $SECRET_ACCESS_KEY | jq -r '.AccessKey.AccessKeyId')

cat <<-EOF > credentials

[default]
aws_access_key_id = $(echo $ACCESS_KEY_ID)
aws_secret_access_key = $(echo $SECRET_ACCESS_KEY | jq -r
'.AccessKey.SecretAccessKey')
```

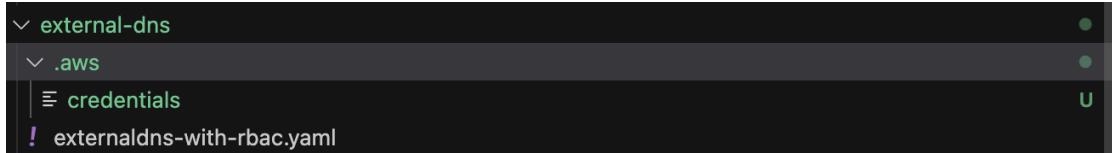
EOF

c. 9.3.3 Create Kubernetes secret from credentials

```
kubectl create secret generic external-dns \
--namespace ${EXTERNALDNS_NS:-"default"} --from-file /local/path/to/
credentials
```

3. Create `externaldns-with-rbac.yaml`. This YAML example file below:

- Make sure that the credentials file location matches the location specified in the yaml below. Currently the yaml is expecting this file to be in the local path of `/ .aws/credentials`. Create this folder and move the credentials file there. Ex:



```
# comment out sa if it was previously created
apiVersion: v1
kind: ServiceAccount
metadata:
  name: external-dns
  labels:
    app.kubernetes.io/name: external-dns
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: external-dns
  labels:
    app.kubernetes.io/name: external-dns
rules:
- apiGroups: [""]
  resources: ["services","endpoints","pods","nodes","ingresses"]
  verbs: ["get","watch","list"]
- apiGroups: ["extensions","networking.k8s.io"]
  resources: ["ingresses"]
  verbs: ["get","watch","list"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: external-dns-viewer
  labels:
    app.kubernetes.io/name: external-dns
```

```

roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: external-dns
subjects:
  - kind: ServiceAccount
    name: external-dns
    namespace: default # change to desired namespace: externaldns, kube-addons
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: external-dns
  labels:
    app.kubernetes.io/name: external-dns
spec:
  strategy:
    type: Recreate
  selector:
    matchLabels:
      app.kubernetes.io/name: external-dns
  template:
    metadata:
      labels:
        app.kubernetes.io/name: external-dns
  spec:
    serviceAccountName: external-dns
    containers:
      - name: external-dns
        image: registry.k8s.io/external-dns/external-dns:v0.14.2
        # image: registry.k8s.io/external-dns/external-dns:v1.14.5
        args:
          - --source=service
          - --source=ingress
          - --domain-filter=ovsa.kddccorp.com # will make ExternalDNS see only the
hosted zones matching provided domain, omit to process all available hosted zones
          - --provider=aws
          # - --policy=upsert-only # would prevent ExternalDNS from deleting any
records, omit to enable full synchronization
          - --registry=txt
          - --txt-owner-id=external-dns
        env:
          - name: AWS_DEFAULT_REGION
            value: us-east-1 # change to region where EKS is installed
        # Uncommend below if using static credentials
          - name: AWS_SHARED_CREDENTIALS_FILE
            value: ./aws/credentials
    volumeMounts:
      - name: aws-credentials
        mountPath: ./aws
        readOnly: true
  volumes:
    - name: aws-credentials

```

```
secret:
  secretName: external-dns
```

There is an optional args that will restrict `externaldns` from only manipulating the filtered domain to avoid issues where other DNS entries are incorrectly edited. Edit the `externaldns-with-rbac.yaml` file and add this entry in the `args:` section:

```
spec:
...
  template:
    ...
      spec:
        serviceAccountName: external-dns
        containers:
          - name: external-dns
            image: registry.k8s.io/external-dns/external-dns:v0.14.2
            # image: registry.k8s.io/external-dns/external-dns:v1.14.5
            args:
              - --source=service
              - --source=ingress
            ###here--> - --domain-filter=ovsa.kddccorp.com # will make ExternalDNS see only the
            hosted zones matching provided domain, omit to process all available hosted zones
              - --provider=aws
              # - --policy=upsert-only # would prevent ExternalDNS from deleting any
            records, omit to enable full synchronization
              - --registry=txt
              - --txt-owner-id=external-dns
...

```

Apply with:

```
kubectl apply --filename externaldns-with-rbac.yaml --namespace ${EXTERNALDNS_NS:-"default"}
```

Now externalDNS is configured to use the alternative DNS Zone in AWS.

10 Resources Deployed in This Tutorial

10.1 Azure subscription nv-Omniverse-Nucleus

Resources deployed in the rg-elitang resource group

NAME	TYPE	LOCATION
aks-elitang	Kubernetes service	East US
APIM-elitang	API Management service	East US
appgw-elitang	Application gateway	East US
appgw-waf-elitang	Application Gateway WAF policy	East US
CPU Usage Percentage - aks-elitang	Metric alert rule	Global
elitang-aks-template	Template spec	East US
elitang-apim-template	Template spec	East US
elitang-appgateway-template	Template spec	East US
elitang-nsg	Network security group	East US
elitang-ovas-streaming.net²⁶	Private DNS zone	Global
elitang-pub	Public IP address	East US
elitang-vnet	Virtual network	East US

26 <http://elitang-ovas-streaming.net>

kubernetes-ac3d328687b004cd5a68c4bf9981f6cc	Public IP address	East US	
Memory Working Set Percentage - aks-elitang	Metric alert rule	Global	
RecommendedAlertRules-AG-1	Action group	Global	

In order to access AKS from the CLI or see Kubernetes resources in the portal, you must add RBAC to your user:

To see more go to [RBAC for AKS \(see page 29\)](#)

Recordset entries in the nvsbx.omniverse.nvidia.com²⁷ Public DNS Zone. This belongs to `fidot_sandbox` resource group.

The screenshot shows the Azure DNS zone management interface for the domain `nvsbx.omniverse.nvidia.com`. The left sidebar shows navigation options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings, Properties, Locks, DNS Management, and Recordsets. The `Recordsets` option is selected and highlighted in grey. The main pane displays a table of record sets with the following data:

Record Name	Type	TTL	Value	Actions
appgw.dev-c	A	5	135.237.33.116	edit delete
ecfben	A	5	57.151.11.143	edit delete
elitang-ovas-streaming	A	3600	48.216.179.29	edit delete
viewer.elitang-ovas-streaming	A	3600	51.8.220.1	edit delete
externaldns-a-ecfben	TXT	300	"heritage=external-dns,external-	edit delete

27 <http://nvsbx.omniverse.nvidia.com>

10.2 AWS

10.2.1 Drew Como's sandbox account: omni-aws-sandbox

Sign in as IAM user

Account ID (12 digits) or account alias

IAM user name

Password

Remember this account

Sign in

[Sign in using root user email](#)

[Forgot password?](#)

The screenshot shows the AWS Route 53 service interface. On the left, there is a navigation sidebar with various options like Dashboard, Hosted zones, Health checks, Profiles, IP-based routing, Traffic flow, Domains, Resolver, VPCs, Inbound endpoints, Outbound endpoints, Rules, Query logging, and Outposts. The 'Hosted zones' option is selected.

In the main content area, the URL is 'Route 53 > Hosted zones > ovsa.kddccorp.com'. The page title is 'Public ovsa.kddccorp.com Info'. There are three buttons at the top right: 'Delete zone', 'Test record', and 'Configure query logging'. Below this is a 'Hosted zone details' section with a 'Edit hosted zone' button.

A tab bar at the bottom of this section shows 'Records (5)', 'DNSSEC signing', and 'Hosted zone tags (0)'. The 'Records (5)' tab is selected. A table below lists five records:

Record name	Type	Alias	Value/Route traffic to	TTL (s...)	Health ...	Evaluat...		
ovsa.kddccorp.com	NS	Simple	-	No	ns-681.awsdns-21.net. ns-462.awsdns-57.com. ns-1157.awsdns-16.org. ns-1745.awsdns-26.co.uk.	172800	-	-
ovsa.kddccorp.com	SOA	Simple	-	No	ns-681.awsdns-21.net.awsdns-16.org.awsdns-26.co.uk.	900	-	-
_668120396cd4dd95cc3cb...	CNAME	Simple	-	No	ns-318.awsdns-39.com. ns-1152.awsdns-16.org. ns-845.awsdns-41.net. ns-1760.awsdns-28.co.uk.	300	-	-
dev.ovsa.kddccorp.com	NS	Simple	-	No	ns-681.awsdns-21.net. ns-462.awsdns-57.com. ns-1157.awsdns-16.org. ns-1745.awsdns-26.co.uk.	300	-	-
test.ovsa.kddccorp.com	A	Simple	-	No	74.208.236.206	300	-	-

10.2.2 AWSOS-AD-Admin

- I created a Hosted Zone which uses Drew Como's sandbox domain
- The entry in ovsa.kddccorp.com which is dev.ovsa.kssccorp.com is necessary for the below hosted zone to function. This hosted zone is where my current ExternalDNS makes changes

The screenshot shows the AWS Route 53 console with the following details:

- Route 53 > Hosted zones > dev.ovsa.kddccorp.com**
- Records (7)**: The table lists the following records:

Record name	Type	Value	TTL
dev.ovsa.kddccorp.com	NS	ns-318.awsdns-39.com, ns-1152.awsdns-16.org, ns-845.awsdns-41.net, ns-1760.awsdns-28.co.uk	172800
_570a4e41d78b80ffcb03966b...	CNAME	qJUq309uMTHIRELNlq7gzB...	300
_acme-challenge.dev.ovsa.kddc...	TXT	"qJUq309uMTHIRELNlq7gzB..."	300
a-yhoqtq.dev.ovsa.kddccorp.com	TXT	"heritage=external-dns,exter..."	300
yhoqtq.dev.ovsa.kddccorp.com	A	135.237.21.18	5
yhoqtq.dev.ovsa.kddccorp.com	TXT	"heritage=external-dns,exter..."	300