Installing vThunder ADC using AWS CFT Templates

v1.0.0

January, 2023

Information in this document is subject to change without notice.

PATENT PROTECTION

A10 Networks, Inc. products are protected by patents in the U.S. and elsewhere. The following website is provided to satisfy the virtual patent marking provisions of various jurisdictions including the virtual patent marking provisions of the America Invents Act. A10 Networks, Inc. products, including all Thunder Series products, are protected by one or more of U.S. patents and patents pending listed at: a10-virtual-patent-marking.

TRADEMARKS

A10 Networks, Inc. trademarks are listed at: a10-trademarks

CONFIDENTIALITY

This document contains confidential materials proprietary to A10 Networks, Inc. This document and information and ideas herein may not be disclosed, copied, reproduced or distributed to anyone outside A10 Networks, Inc. without prior written consent of A10 Networks, Inc.

DISCLAIMER

This document does not create any express or implied warranty about A10 Networks, Inc. or about its products or services, including but not limited to fitness for a particular use and non-infringement. A10 Networks, Inc. has made reasonable efforts to verify that the information contained herein is accurate, but A10 Networks, Inc. assumes no responsibility for its use. All information is provided "as-is." The product specifications and features described in this publication are based on the latest information available; however, specifications are subject to change without notice, and certain features may not be available upon initial product release. Contact A10 Networks, Inc. for current information regarding its products or services. A10 Networks, Inc. products and services are subject to A10 Networks, Inc. standard terms and conditions.

ENVIRONMENTAL CONSIDERATIONS

Some electronic components may possibly contain dangerous substances. For information on specific component types, please contact the manufacturer of that component. Always consult local authorities for regulations regarding proper disposal of electronic components in your area.

FURTHER INFORMATION

For additional information about A10 products, terms and conditions of delivery, and pricing, contact your nearest A10 Networks, Inc. location, which can be found by visiting www.a10networks.com.

Table of Contents

Introduction

vThunder is a fully operational, software-based Application Delivery Controller (ADC) solution that can run on Amazon Web Services (AWS) cloud. vThunder provides a robust, flexible, and easy-to-deploy application delivery and server load balancing service.

ACOS uses the CloudFormation Templates (CFT) to quickly deploy the vThunder instance on the AWS cloud. Table 1 lists the available AWS CFT for deploying vThunder ADC on AWS cloud:

Table 1 : Available AWS CloudFormation Templates

Template	Description	Configuration
A10-vThunder_ ADC-2NIC-1VM	• Creates one vThunder instance with two Network Interface Cards (NICs). • Deploys a Certificate Authority SSL Certificate and Server Load Balancer (SLB).	• 2 NICs (1 Management + 1 Data) • BYOL (Bring Your Own License) • 1 VM (vThunder Virtual Instance) • SLB (vThunder Server Load Balancer) • SSL (Apply SSL Certificate)
A10-vThunder_ ADC-2NIC-1VM-GLM	• Creates one vThunder instance with two Network Interface Cards and A10 Global License Manager (GLM) integration. • Deploys a Certificate Authority SSL Certificate and Server Load Balancer.	• 2 NICs (1 Management + 1 Data) • BYOL (Bring Your Own License) • 1 VM (vThunder Virtual Instance) • SLB (vThunder Server Load Balancer) • SSL (Apply SSL Certificate) • GLM (Auto apply A10

Template	Description	Configuration
		license)
A10-vThunder_ ADC-3NIC-2VM-HA	• Creates two vThunder instances with High Availability (HA) setup, each vThunder contains three Network Interface Cards. • Deploys a Certificate Authority SSL Certificate and Server Load Balancer.	• 3 NICs (1 Management + 2 Data) • BYOL (Bring Your Own License) • 2 VMs (vThunder Virtual Instances) • SLB (vThunder Server Load Balancer) • SSL (Apply SSL Certificate) • HA (High Availability with auto switchover with next available vThunder VM using VRRP)
A10-vThunder_ ADC-3NIC-2VM-HA-GLM-PVTVIP	• Creates two vThunder instances with High Availability setup and an A10 Global License Manager integration, each vThunder has three Network Interface Cards. • Deploys a Certificate Authority SSL Certificate, and a Server Load Balancer.	• 3 NICs (1 Management + 2 Data) • BYOL (Bring Your Own License) • 2 VMs (vThunder Virtual Instances) • SLB (vThunder Server Load Balancer) • SSL (Apply SSL Certificate) • GLM (Auto apply A10 license) • HA (High Availability with auto switchover with available VM using VRRP) • VIP (Private Interface)

Template	Description	Configuration
A10-vThunder_ ADC-3NIC-2VM-HA-GLM-PUBVIP-BACKAUTO	• Creates two vThunder instances with High Availability (HA) setup and GLM integration, each vThunder contains three Network Interface Cards. • Deploys a Certificate Authority SSL Certificate, Server Load Balancer, and backend server autoscaling support.	• 3 NICs (1 Management + 2 Data) • BYOL (Bring Your Own License) • 2 VMs (vThunder Virtual Instances) • SLB (vThunder Server Load Balancer) • SSL (Apply SSL Certificate) • GLM (Auto apply A10 license) • HA (High Availability with auto switchover for the available VM using VRRP) • VIP (Public Interface) • BACKAUTO (Lambda function to apply SLB config into vThunder for newly added/deleted web/app servers via autoscaling group)
A10-vThunder_ ADC-3NIC-6VM-2RG-GSLB	• Creates two Global Server Load Balancer (GSLB) regions, one GSLB controller and two site devices in each of the two regions. • Creates two real servers (Ubuntu 16.04.0-LTS) in each region.	• 3 NICs (1 Management + 2 Data) • BYOL (Bring Your Own License) • 6 VMs (Three vThunder Virtual Instances in each Region) • 2 RGs (Regions) • GSLB (vThunder SLB in

Template	Description	Configuration
		multiple regions)

This documentation helps you to deploy the vThunder instance/s on the AWS cloud after completion of following tasks:

- Downloading the required template from GitHub on your local machine
- Configuring the vThunder installation parameters in the template
- Executing required commands in the AWS terminal window.

AWS Terminology

Below is the AWS glossary that explains the most common terms used while using AWS.

- **Access control list (ACL)** — A firewall or a security layer on the subnet level. For more information, see https://docs.aws.amazon.com/AmazonS3/latest/userguide/acls.html
- **CloudWatch** — A service that allows you to monitor various elements of your AWS account. For more information, see https://docs.aws.amazon.com/cloudwatch/index.html
- **Lambda** — A serverless computing that will replace Elastic Compute Cloud (EC2) instances, for most of the functionality of EC2. For more information, see https://docs.aws.amazon.com/lambda/latest/dg/welcome.html
- **Security group (SG)** — The firewall or security layer on the server or instance level. For more information, see https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-security-groups.html.
- **Subnet** — A subsection of a network that generally includes all the computers in a specific location. For more information, see https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-ec2-subnet.html.

- **Virtual Private Cloud (VPC)** — A private subsection of AWS you control and in which you can place AWS resources. For more information, see https://docs.aws.amazon.com/vpc/latest/userguide/what-is-amazon-vpc.html.

Prerequisites

To deploy vThunder on AWS cloud using any of the available CFT, you must ensure the following prerequisites are met:

- AWS account with a valid subscription and access keys (Required)
  - Create the access keys *(access key ID* and *secret access key)* if you don't have them already. For more information, see https://docs.aws.amazon.com/powershell/latest/userguide/pstools-appendix-sign-up.html
  - Create an SSH key from **AWS Management Console** > **EC2 Formation** > **Key Pairs** > **Create key pair** with the following:
    - **Name**: *your key name*
    - **Key pair type**: **RSA**
    - **Private key file format**: **.pem**
      Save this SSH key for future use. For more information, see https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/create-key-pairs.html
  - Download the following AWS tool to create and manage resources:
    - AWS Management Console — A web console to create and monitor AWS resources.
    - **AWS Command Line Interface (CLI)** — An interface that can be launched to start a CLI session. The interface can be launched using one of the following:

      - **Linux shells** — Use programs such as bash, zsh, and tcsh to run commands in Linux or macOS.

      - **Windows command line** — Use Windows command prompt or PowerShell to run commands in Windows.

- **Remotely** — Use Amazon Elastic Compute Cloud (Amazon EC2) instances through a remote terminal program such as MobaXterm, PuTTY, SSH, or with AWS Systems Manager to run commands in AWS. For more information, see http//docs.aws.amazon.com/cli/index.html?nc2=h_ql_doc_cli

- AWS User
  A user with predefined and custom policies applied. For more information, see Security Policy for AWS User.
- Configure the following parameters if you are deploying Backend server autoscaling CFT:
  - Create AWS Secrets Manager secret from **AWS Management Console** > **Secrets Manager** > **Store a new secret** with the following:

    - **Secret Type**: Other type of secret

    - **Key/value pairs**: Provide the following values:

    Table 2 : Key value pair

Key	Value
aws_access_key_id	*your aws access key id*
aws_secret_access_key	*your aws secret access key*

    - **Encryption key**: aws/secretsmanager

    - **Secret name**: *your_AWS_keys_secret_manager_name*

    For more information, see https://docs.aws.amazon.com/secretsmanager/latest/userguide/create_secret.html
  - Lambda Function Execution Role ARN
    This is required for the Autoscaling template. For more information, see https://docs.aws.amazon.com/lambda/latest/dg/lambda-intro-execution-role.html#permissions-executionrole-console

- Valid SSL certificate to apply on vThunder (Optional).
- Python [3.8.5 and later] with PIP.
- Text editor (Notepad++, Notepad or any other text editor application).

- A10 GLM account access with valid licenses.
  This access is required for the templates using GLM. For more information, see Global License Manager User Guide.
- CloudFormation Template (CFT)
  Go to GitHub [Branch: release/v1.0.0] and download the required CFT folder to your local machine. For example, C:\Users\TestUser\Templates. The template folder contains the json parameter files and Python scripts for the deployment of the respective template.
- A10 vThunder default user credentials
  Send a request to A10 Networks Support for A10 vThunder login default user credentials.

Image Repository

CFT templates are tested with the following A10 vThunder images:

- 64-bit Advanced Core OS (ACOS) version 5.2.1-P6, build 74
- 64-bit Advanced Core OS (ACOS) version 5.2.1-P5, build 113

Configure Operating System

Before deploying CFT, perform the following steps depending on your operating system:

- Windows
- Linux, macOS, Unix

Windows

If you are deploying the CFT using Windows, perform the following steps:

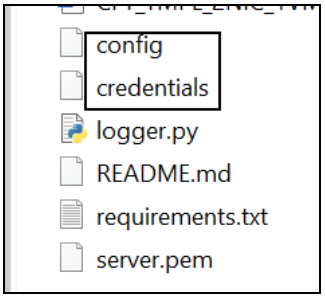
1. Verify if the recommended Python version is installed correctly.
   Launch the command prompt from Windows search, and enter the following command:
   ```
   C:\Users\TestUser> python
   ```

If Python is installed, the version details are displayed. Ensure that the version is 3.8.5 or higher.

2. Verify if PIP is installed.
   In the command prompt, enter the following command:
   ```
   C:\Users\TestUser> pip
   ```
   If PIP is installed, the pip command usage, commands and other general options are displayed.

3. Install all Python dependencies.
   a. From the command prompt, navigate to the downloaded CFT folder path and enter the following command:
      ```
      C:\Users\TestUser\Templates> pip install -r requirements.txt
      ```
      A .aws folder is automatically created under C:\Users\TestUser.
   b. From the downloaded CFT folder, locate and open the **credentials** file with a text editor.

      Figure 1 : CFT folder

   c. Update the access key ID and secret access key as per your AWS account and then save the changes.
      ```
      [default]
      aws_access_key_id = your_aws_access_key_id
      aws_secret_access_key = your_aws_secret_access_key
      ```
   d. Copy this file to the C:\Users\TestUser\.aws folder.
   e. From the downloaded CFT template folder, locate and open the **config** file with a text editor.

f. Update region with the your working region and then save the changes.

```
[default]
region = you_working_region
output = json
```

g. Copy this file to the C:\Users\TestUser\.aws folder.

Linux, macOS, Unix

If you are deploying the CFT using either Linux, macOS, or Unix, perform the following steps:

1. Verify if the recommended Python version is installed.
   Open the Shell terminal, enter the following command:

   ```
   testuser@localhost:~$ python
   ```

   If Python is installed, the version details are displayed.

2. Verify if PIP is installed.
   In the command prompt, enter the following command:

   ```
   testuser@localhost:~$ pip
   ```

   If PIP is installed, the pip command usage, available commands, and general options are displayed.

3. Install all Python dependencies.

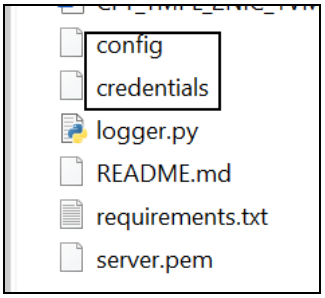
   a. From the command prompt, navigate to the downloaded CFT folder path and enter the following command:

   ```
   testuser@localhost:~$ pip install -r requirements.txt
   ```

   A ~/.aws folder is auto created under /home/username.

   b. From the downloaded CFT folder, locate and open the **credentials** file with a text editor.

Figure 2 : CFT folder



c. Update the access key and secret access key as per your AWS account and then save the changes.

```
[default]
aws_access_key_id = your_aws_access_key_id
aws_secret_access_key = your_aws_secret_access_key
```

d. Copy this file to the ~/.aws folder.

e. From the downloaded CFT folder, locate and open **credentials** file with a text editor.

f. Update region with logged-in-user region and then save the changes.

```
[default]
region = logged_in_user_region
output = json
```

g. Copy this file to the ~/.aws folder.

For more information, see https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-credentials.html

CloudFormation Templates

To implement infrastructure as a code for your AWS solution, use CFT. The template is a json native file that defines the infrastructure and configuration for your project. The template uses declarative syntax to specify the resources that are to be deployed and the properties for those resources without having to write the sequence of programming commands to create it.

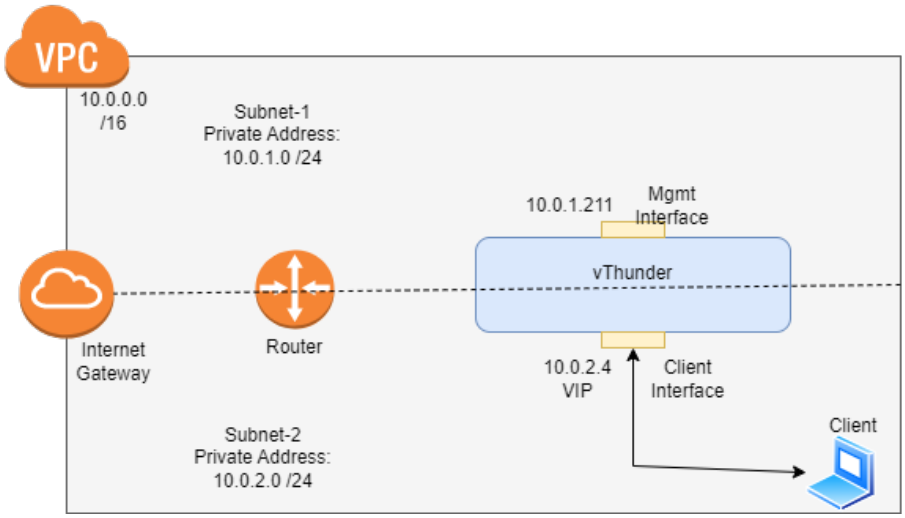
The following templates are available:

- Deploy CFT A10-vThunder_ADC-2NIC-1VM
- Deploy CFT A10-vThunder_ADC-2NIC-1VM-GLM
- Deploy CFT A10-vThunder_ADC-3NIC-2VM-HA
- Deploy CFT A10-vThunder_ADC-3NIC-2VM-HA-GLM-PVTVIP
- Deploy CFT A10-vThunder_ADC-3NIC-2VM-HA-GLM-PUBVIP-BACKAUTO
- Deploy CFT A10-vThunder_ADC-3NIC-6VM-2RG-GSLB

Deploy CFT A10-vThunder_ADC-2NIC-1VM

Using this template, one vThunder instance containing one management interface and one data interface can be deployed.

Figure 3 : 2NIC-1VM Deployment Topology



The following topics are covered:

System Requirements

The CFT displays the default values when you download and save the files on your local machine. You can modify the default values based on your deployment.

You need to configure the following resources to deploy vThunder on the AWS cloud:

Table 3 : System Requirements

Resource Name	Description	Default Value
Stack	A new stack is created with the specified name and location.	Here, the stack name used is **vth**.
Network Interface Card [NIC]	Two types of interfaces are created for a vThunder instance: • One management interface • One data interface	**vth-inst1-mgmt-nic1** **vth-inst1-data-nic1**
Subnet	Two subnets are created with an address prefix each.	**vth-vpc-mgmt-subnet1** **vth-vpc-data-subnet1**
Virtual Private Network [VCN]	A virtual private network is assigned to the virtual machine instance.	**vth-vpc** Address prefix for virtual network: 10.0.0.0/16
Elastic Public IP	Elastic Public IP is created and attached to the management interface of the vThunder instance.	**vth-inst1-mgmt-nic1-ip**
Security Group	One security group is created for the management interface. One security group is created for the data interface. **Logical name**: • vThunderSecurityGroupMgmt	Here, the tag names are: **vth-sg-mgmt** **vth-sg-data**

Resource Name	Description	Default Value
	• vThunderSecurityGroupData	
vThunder Instance	A vThunder EC2 instance is created. **Default type**: m4.xlarge (40 Gb memory) Table 4 lists the supported instance types.	`vth-inst1`
Ubuntu Server Instance	One Ubuntu Server instance is created. **Default size**: t2.micro	`vth-server`

Supported Instance Types

Table 4 : List of Supported Instance Types

Instance	vCPU	Memory	Number of Network Interfaces
c4.xlarge	4	7680	4
c4.4xlarge	16	30720	8
c4.8xlarge	36	61440	8
d2.xlarge	4	31232	4
d2.2xlarge	8	62464	4
d2.4xlarge	16	124928	8
d2.8xlarge	36	249856	8
m4.xlarge	4	16384	4
m4.2xlarge	8	32768	4
m4.4xlarge	16	65536	8
m4.10xlarge	40	163840	8
i2.xlarge	4	31232	4

Instance	vCPU	Memory	Number of Network Interfaces
i2.2xlarge	8	62464	4
i2.4xlarge	16	124928	8
i2.8xlarge	32	249856	8
c5d.large	2	4096	3
c5d.9xlarge	36	73728	8
c5d.2xlarge	8	32768	4
c5d.4xlarge	16	73728	8
c5.xlarge	4	8192	4
c5.2xlarge	8	16384	4
c5.4xlarge	16	32768	8
c5.9xlarge	36	73728	8
g3.4xlarge	16	124928	8
g3.8xlarge	32	249856	8
i3.large	2	15616	3
i3.xlarge	4	31232	4
i3.2xlarge	8	62464	4
i3.4xlarge	16	124928	8
i3.8xlarge	32	249856	8
m5d.large	2	8192	3
m5d.xlarge	4	16384	4
m5d.2xlarge	8	32768	4
m5d.4xlarge	16	65536	8
m5.large	2	8192	3
m5.xlarge	4	16384	4
m5.2xlarge	8	32768	4
m5.4xlarge	16	65536	8
r5d.large	2	16384	3
r5d.xlarge	4	32768	4

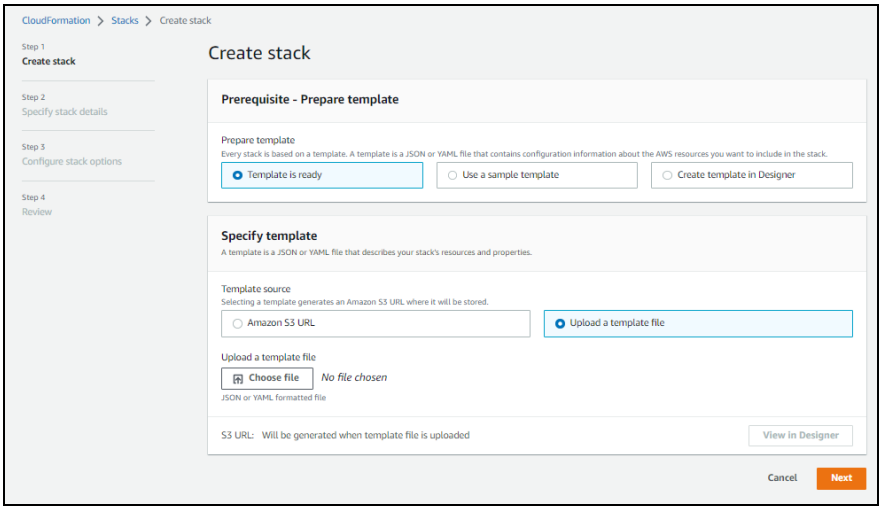
Instance	vCPU	Memory	Number of Network Interfaces
r5d.2xlarge	8	65536	4
r5d.4xlarge	16	131072	8
r5.large	2	16384	3
r5.xlarge	4	32768	4
r5.2xlarge	8	65536	4
r5.4xlarge	16	131072	8
r4.large	2	15616	3
r4.xlarge	4	31232	4
r4.2xlarge	8	62464	4
r4.4xlarge	16	124928	8
r4.8xlarge	32	249856	8
t3.medium	2	4096	3
t3.large	2	8192	3
t3.xlarge	4	16384	4
t3.2xlarge	8	32768	4
z1d.large	2	16384	3
z1d.xlarge	4	32768	4
z1d.2xlarge	8	65536	4
z1d.3xlarge	12	98304	8
z1d.6xlarge	24	196608	8

Create vThunder Instance

To create a vThunder instance, perform the following steps:

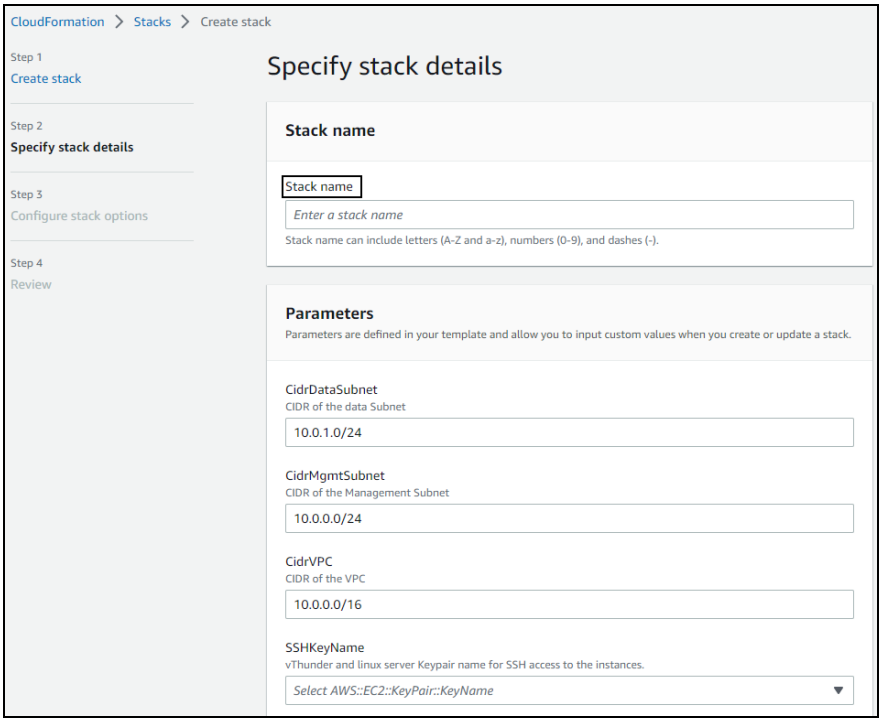
1. From **AWS Management Console**, navigate to **CloudFormation** > **Stacks** > **Create Stack** > **With new resources (standard)**.
   The **Create stack** window is displayed.

Figure 4 : Create stack window



2. In the **Prerequisite - Prepare template** section, select **Template is ready** option. After the selecting this option, the **Specify template** section is displayed.
3. In the **Specify template** section, select **Upload a template file** and click **Choose file** to browse and upload the following template file from the downloaded CFT folder:
   CFT_TMPL_2NIC_1VM_1.json
   The selected template file name is displayed as the chosen file.
4. Click **Next**.
   The **Specify stack details** window is displayed.

Figure 5 : Specify stack details window



5. In the **Specify stack details** window, enter or select the following:
   a. In the **Stack name** section, enter a **Stack name**.

      Here, **`vth`** is the stack name.

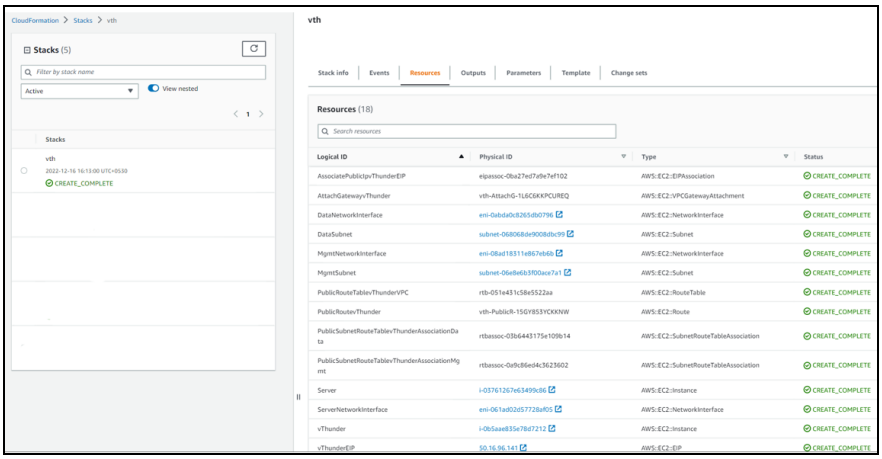
   b. In the **Parameters** section, enter or select the required value in the following fields:
      - **KeyPairName**: *your SSH key*
      - **TagValue**: **`a10-vthunder-adc`**
      - **Zone:** *your availability zone*
   c. Verify the other fields and change the values as required. (Optional)
6. Click **Next**.
   The **Configure stack options** window is displayed.
7. Verify the fields and change the values as required. (Optional)
8. Click **Next**.
   The **Review** window is displayed.

9. Verify if all the stack configurations are correct and then click **Submit**.

NOTE: The system may take a few minutes to create the resources. So, wait until the stack status is **CREATE_COMPLETE**.

10. Verify if all the above resources are created in the **AWS Management Console** > **CloudFormation** > **Stacks** > <stack_name> > **Resources** tab.

Figure 6 : Resource listing in the resource group



The vThunder instance is listed under **Resources** tab.

Access vThunder using GUI or CLI

vThunder instance can be accessed using any of the following ways:

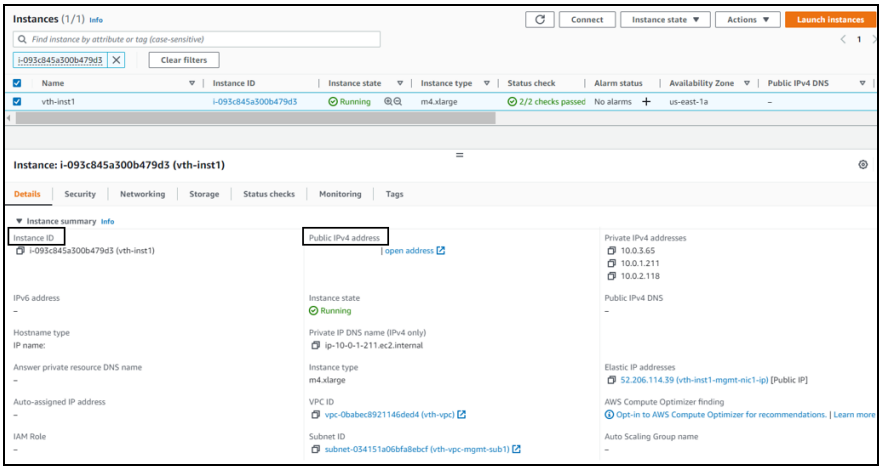
- Access vThunder using GUI
- Access vThunder using CLI

Access vThunder using GUI

To access vThunder instance using GUI, perform the following steps:

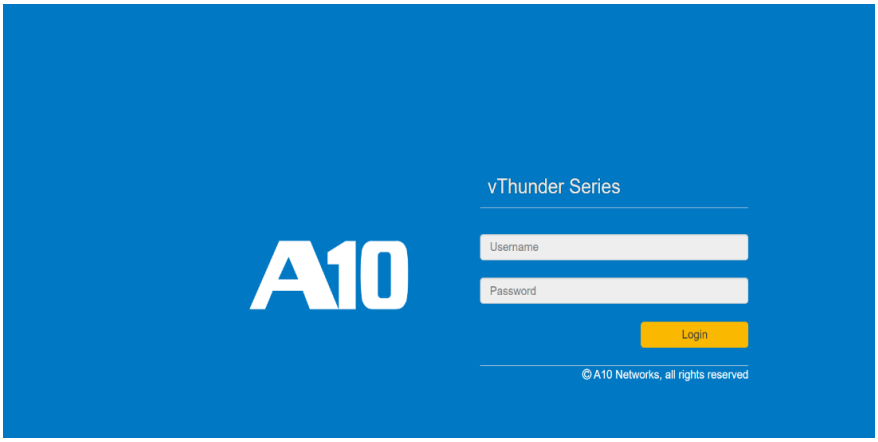
1. From **AWS Management Console**, navigate to **EC2** > **Instances**.
2. Select your instance name.
   Here, **`vth-inst1`** is the vThunder instance.

Figure 7 : vThunder instance



3. Copy the **Public IPv4 address** from the **Details** tab and replace the IP address in the below link:
   http://<vThunder_public_IPv4_address>
4. Open the updated link in any browser.
   The vThunder login window is displayed.

Figure 8 : vThunder GUI



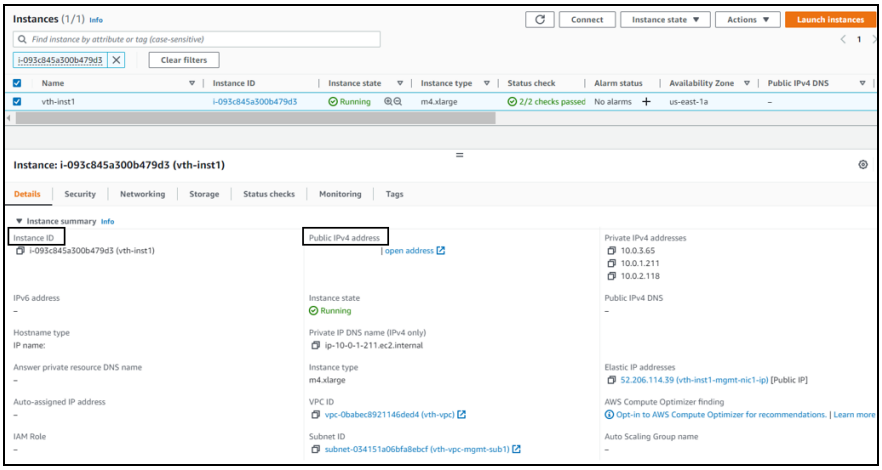
5. Enter the following credentials:
   - Username – admin
   - Password – *EC2 Instance ID*
     The home page is displayed if the entered credentials are correct.

Access vThunder using CLI

To access vThunder instance using CLI, perform the following steps:

1. From **AWS Management Console**, navigate to **EC2** > **Instances**.
2. Select your instance name.
   Here, **`vth-inst1`** is the vThunder instance.

   Figure 9 : vThunder instance



3. Copy the **Public IPv4 address** from the **Details** tab.
4. Open any SSH client and provide the following to establish a connection:
   - Hostname: Public IPv4 address
   - Username: admin
   - Key: SSH Key
5. Connect to the session.
6. In the SSH client session, enter the following commands:

```
vThunder(NOLICENSE)>enable <---Execute command--->
Password:                  <---just press Enter key--->
vThunder(NOLICENSE)#config <---Configuration mode--->
vThunder(config)(NOLICENSE)#
```

The vThunder instance is ready to use.

Configure Server and Client Machine

To test the traffic flow via vThunder, create and configure a server machine and a client machine:

Configure Server Machine

To configure a server machine, perform the following steps:

1. From **AWS Management Console**, navigate to **EC2** > **Instances**.
2. Select your server instance name.
   Here, **`vth-server`** is the server instance name.
3. Click **Connect**.
   A **Connect to instance** window with **EC2 Instance Connect** tab is displayed.
4. Click **Connect**.
   A **Terminal** window is displayed.
5. Run the following command to create an Apache Server virtual machine:
   ```
   sudo apt install apache2
   ```
   While the Apache server is getting installed, you get a prompt to continue further. Enter 'Y' to continue. After the installation is complete, a newline prompt is displayed.

Configure a Client Machine

To configure a client machine, perform the following steps:

1. From **AWS Management Console**, navigate to **EC2** > **Instances**.
2. Click **Launch Instances**.
   A **Launch an instance** window is displayed.
3. In the **Name and tags** section, enter an instance name.
   Here, enter **`vth-client`** as the client instance name.

4. In the **Application and OS Images** section, select **Ubuntu**.
5. In the **Instance type** section, select the required instance type.
6. In the **Key pair (login)** field, select your SSH key.
7. In the **Network settings** section, click **Edit** to edit the following:
   - VPC: *your VPC*
     Here, `vth-vpc` is the VPC.
   - Subnet: Data subnet
     Here, `10.0.1.0/24` is the data subnet value.
   - Auto-assign public IP: Enable
   - Firewall (security groups): Select existing security group
   - Common security groups: *your data security group*
     Here, `vth-vThunderSecurityGroupData` is the security group.
8. Click **Launch instance**.

NOTE: The system may take a few minutes to launch the instance.

The client instance is displayed in the **Instances** list with the status as **Running**.

Configure vThunder as an SLB

The following topics are covered:

- Initial Setup
- Deploy vThunder as an SLB

Initial Setup

Before deploying vThunder on AWS cloud as an SLB, you need to configure the corresponding parameters in the CFT.

To configure the parameters, perform the following steps:

1. Navigate to the folder where you have downloaded the CFT, and open the CFT_TMPL_2NIC_1VM_CONFIG_SLB_SSL_PARAM.json with a text editor.

NOTE: Each parameter has a default value mentioned in the parameter file.

2. Configure the stack name created using CFT.

```
"stackDetails": {
  "value": [
    {
      "stackName": "vth"
    }
  ]},
```

3. Configure SLB server ports.

```
"server-list": {
  "value": [
    {
      "port-list": [
        {
          "port-number": 53,
          "protocol": "udp"
        },
        {
          "port-number": 80,
          "protocol": "tcp"
        },
        {
          "port-number": 443,
          "protocol": "tcp"
        }
      ]
    }
  ]
},
```

4. Configure Service Group List ports.
   The Service group name by default is “sg+port_number”. If you may want to change the service group name then after changing the name, change the names in the Virtual servers as well.

```
"serviceGroupList": {
  "value": [
    {
      "name": "sg443",
      "protocol": "tcp",
      "member-list": [
        {
          "port": 443
        }
      ]
    },
    {
      "name": "sg53",
      "protocol": "udp",
      "member-list": [
        {
          "port": 53
        }
      ]
    },
    {
      "name": "sg80",
      "protocol": "tcp",
      "member-list": [
        {
          "port": 80
        }
      ]
    }
  ]
},
```

5. Configure a Virtual Server.
   The virtual server default name is “vip”. It is the Private IP address of Ethernet1.

```
"virtualServerList": {
  "virtual-server-name": "vip",
  "metadata": {
    "description": "virtual server is using ethernet 1 ip address"
```

```
    },
    "value": [
      {
        "port-number": 53,
        "protocol": "udp",
        "auto": 1,
        "service-group": "sg53"
      },
      {
        "port-number": 80,
        "protocol": "http",
        "auto": 1,
        "service-group": "sg80"
      },
      {
        "port-number": 443,
        "protocol": "https",
        "auto": 1,
        "service-group": "sg443"
      }
    ]
  },
```

6. Configure SSL.

```
    "sslConfig": {
        "requestTimeOut": 40,
        "Path": "server.pem",
        "File": "server",
        "CertificationType": "pem"
      }
```

NOTE: By default, SSL configuration is disabled i.e., no SSL configuration is applied.

The server.pem file is available in the downloaded CFT folder. You can edit this file as required or use a different certificate file, but the path should be changed accordingly.

7. Verify if all the configurations in the CFT_TMPL_2NIC_1VM_CONFIG_SLB_SSL_PARAM.json file are correct and then save the changes.

Deploy vThunder as an SLB

To deploy vThunder on AWS cloud as an SLB, perform the following steps:

1. From your command prompt, navigate to the folder where you have downloaded the CFT.
2. Run the following command to create vThunder SLB instance:

```
python ./CFT_TMPL_2NIC_1VM_CONFIG_SLB_SSL_2.py
```

A message is prompted to upload the SSL certificate.

```
Do you want to upload ssl certificate(yes/no)?yes
configured ethernet ip
Configured server vth-server
Configured virtual servers
SSL Configured.
Configurations are saved on partition: shared
The certificate available in the sslConfig path is uploaded.
```

If the SSL Certificate upload is successful, a message 'SSL Configured' is displayed.

Verify Deployment

To verify vThunder SLB deployment using CFT, perform the following steps:

1. Run the following command on vThunder:

```
vThunder(config)(NOLICENSE)#show running-config
```

If the deployment is successful, the following SLB configuration is displayed:

```
!Current configuration: 0 bytes
!Configuration last updated at 17:36:35 GMT Wed Nov 30 2022
!Configuration last saved at 17:35:40 GMT Wed Nov 30 2022
!64-bit Advanced Core OS (ACOS) version 5.2.1, build 153 (Dec-11-
2020,14:34)
!
!
```

```
interface ethernet 1
  enable
  ip address dhcp
!
!
slb server vth-server 10.0.1.120
  port 53 udp
  port 80 tcp
  port 443 tcp
!
slb service-group sg443 tcp
  member vth-server 443
!
slb service-group sg53 udp
  member vth-server 53
!
slb service-group sg80 tcp
  member vth-server 80
!
slb virtual-server vip use-if-ip ethernet 1
  port 53 udp
    source-nat auto
    service-group sg53
  port 80 http
    source-nat auto
    service-group sg80
  port 443 https
    source-nat auto
    service-group sg443
!
!
end
```

2. Run the following command on vThunder:

```
vThunder(config)(NOLICENSE)#show pki cert
```

If the deployment is successful, the following SSL configuration is displayed:

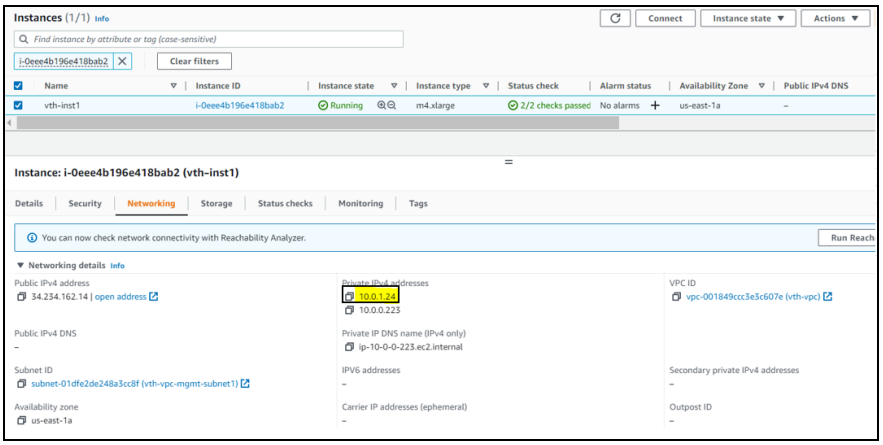
```
Name     Type                          Expiration   Status
-------------------------------------------------------------
server certificate Jan 28 12:00:00 2028 GMT [Unexpired, Bound]
```

Verify Traffic Flow

To verify the traffic flow from client machine to server machine via vThunder, perform the following:

1. From **AWS Management Console**, navigate to **EC2** > **Instances**.
2. Select the vThunder instance name and then click the **Networking** tab.
   Here, **`vth-inst1`** is the vThunder instance name.
3. Copy the IP address of the data subnet under the **Private IPv4 address**.
   Here, **`10.0.1.24`** is the data subnet value.

   Figure 10 : vThunder instance



4. Select your client instance from the **Instances** list.
   Here, **`vth-client`** is the client instance name.
5. Click **Connect**.
   A **Connect to instance** window with **EC2 Instance Connect** tab is displayed.
6. Click **Connect**.
   A **Terminal** window is displayed.
7. Run the following command in the Terminal window to send the traffic from the client machine:

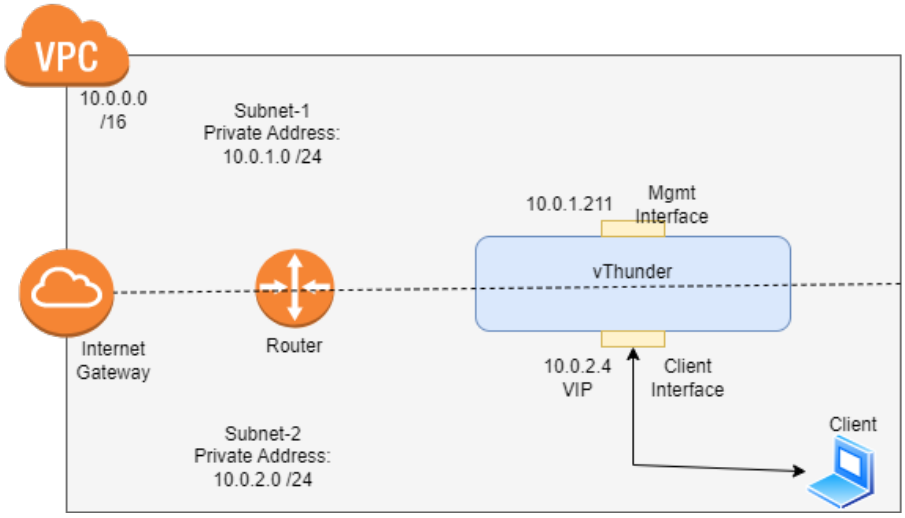
```
curl <vThunder_instance_data_private_IPv4_Address>
```

8. Verify if a response is received.

Deploy CFT A10-vThunder_ADC-2NIC-1VM-GLM

Using this template, one vThunder instance containing one management interface and one data interface with GLM integration can be deployed.

Figure 11 : 2NIC-1VM-GLM Deployment Topology



The following topics are covered:

System Requirements

The CFT displays the default values when you download and save the files on your local machine. You can modify the default values based on your deployment.

You need to configure the following resources to deploy vThunder on the AWS cloud:

Table 5 : System Requirements

Resource Name	Description	Default Value
Stack	A new stack is created with the specified name and location.	Here, the stack name used is **vth**.
Network Interface Card [NIC]	Two types of interfaces are created for a vThunder instance: • One management interface • One data interface	**vth-inst1-mgmt-nic1** **vth-inst1-data-nic1**
Subnet	Two subnets are created with an address prefix each.	**vth-vpc-mgmt-subnet1** **vth-vpc-data-subnet1**
Virtual Private Network [VCN]	A virtual private network is assigned to the virtual machine instance.	**vth-vpc** Address prefix for virtual network: 10.0.0.0/16
Elastic Public IP	Elastic Public IP is created and attached to the management interface of the vThunder instance.	**vth-inst1-mgmt-nic1-ip**
Security Group	One security group is created for the management interface. One security group is created for the data interface. **Logical name**: • vThunderSecurityGroupMgmt	Here, the tag names are: **vth-sg-mgmt** **vth-sg-data**

Resource Name	Description	Default Value
	• vThunderSecurityGroupData	
vThunder Instance	A vThunder EC2 instance is created. **Default type**: m4.xlarge (40 Gb memory) Table 6 lists the supported instance types.	`vth-inst1`
Ubuntu Server Instance	One Ubuntu Server instance is created. **Default size**: t2.micro	`vth-server`

Supported Instance Types

Table 6 : List of Supported Instance Types

Instance	vCPU	Memory	Number of Network Interfaces
c4.xlarge	4	7680	4
c4.4xlarge	16	30720	8
c4.8xlarge	36	61440	8
d2.xlarge	4	31232	4
d2.2xlarge	8	62464	4
d2.4xlarge	16	124928	8
d2.8xlarge	36	249856	8
m4.xlarge	4	16384	4
m4.2xlarge	8	32768	4
m4.4xlarge	16	65536	8
m4.10xlarge	40	163840	8
i2.xlarge	4	31232	4

Instance	vCPU	Memory	Number of Network Interfaces
i2.2xlarge	8	62464	4
i2.4xlarge	16	124928	8
i2.8xlarge	32	249856	8
c5d.large	2	4096	3
c5d.9xlarge	36	73728	8
c5d.2xlarge	8	32768	4
c5d.4xlarge	16	73728	8
c5.xlarge	4	8192	4
c5.2xlarge	8	16384	4
c5.4xlarge	16	32768	8
c5.9xlarge	36	73728	8
g3.4xlarge	16	124928	8
g3.8xlarge	32	249856	8
i3.large	2	15616	3
i3.xlarge	4	31232	4
i3.2xlarge	8	62464	4
i3.4xlarge	16	124928	8
i3.8xlarge	32	249856	8
m5d.large	2	8192	3
m5d.xlarge	4	16384	4
m5d.2xlarge	8	32768	4
m5d.4xlarge	16	65536	8
m5.large	2	8192	3
m5.xlarge	4	16384	4
m5.2xlarge	8	32768	4
m5.4xlarge	16	65536	8
r5d.large	2	16384	3
r5d.xlarge	4	32768	4

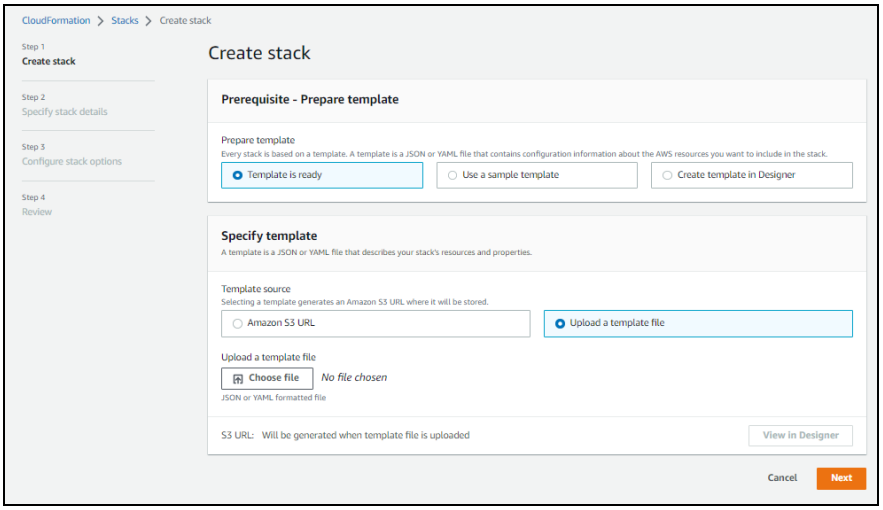
Instance	vCPU	Memory	Number of Network Interfaces
r5d.2xlarge	8	65536	4
r5d.4xlarge	16	131072	8
r5.large	2	16384	3
r5.xlarge	4	32768	4
r5.2xlarge	8	65536	4
r5.4xlarge	16	131072	8
r4.large	2	15616	3
r4.xlarge	4	31232	4
r4.2xlarge	8	62464	4
r4.4xlarge	16	124928	8
r4.8xlarge	32	249856	8
t3.medium	2	4096	3
t3.large	2	8192	3
t3.xlarge	4	16384	4
t3.2xlarge	8	32768	4
z1d.large	2	16384	3
z1d.xlarge	4	32768	4
z1d.2xlarge	8	65536	4
z1d.3xlarge	12	98304	8
z1d.6xlarge	24	196608	8

Create vThunder Instance

To create a vThunder instance, perform the following steps:

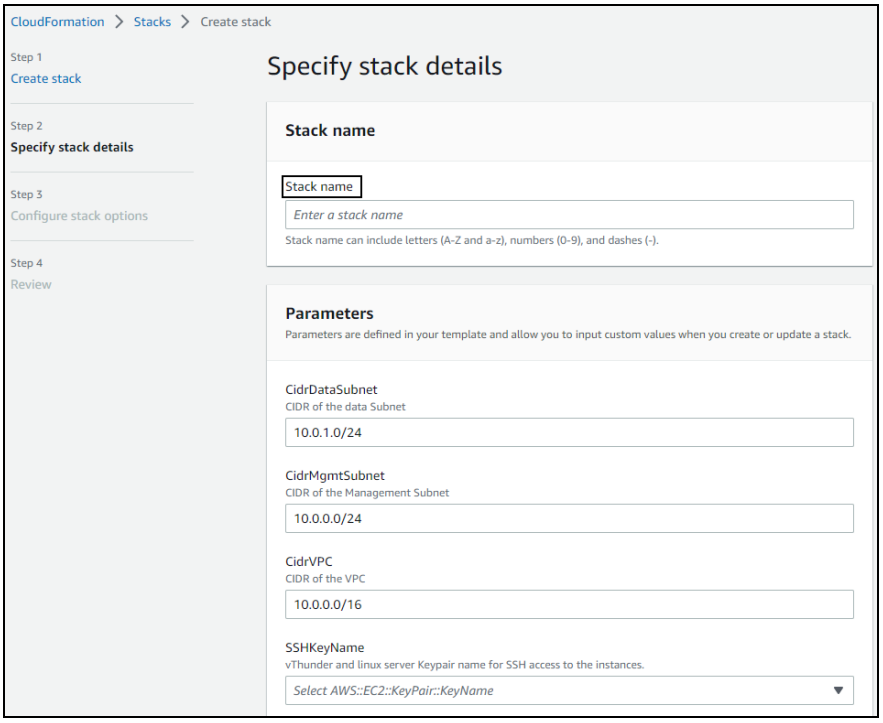
1. From **AWS Management Console**, navigate to **CloudFormation** > **Stacks** > **Create Stack** > **With new resources (standard)**.
   The **Create stack** window is displayed.

Figure 12 : Create stack window



2. In the **Prerequisite - Prepare template** section, select **Template is ready** option. After the selecting this option, the **Specify template** section is displayed.
3. In the **Specify template** section, select **Upload a template file** and click **Choose file** to browse and upload the following template file from the downloaded CFT folder:
   CFT_TMPL_2NIC_1VM_GLM_1.json
   The selected template file name is displayed as the chosen file.
4. Click **Next**.
   The **Specify stack details** window is displayed.

Figure 13 : Specify stack details window



5. In the **Specify stack details** window, enter or select the following:
   a. In the **Stack name** section, enter a **Stack name**.

      Here, **`vth`** is the stack name.

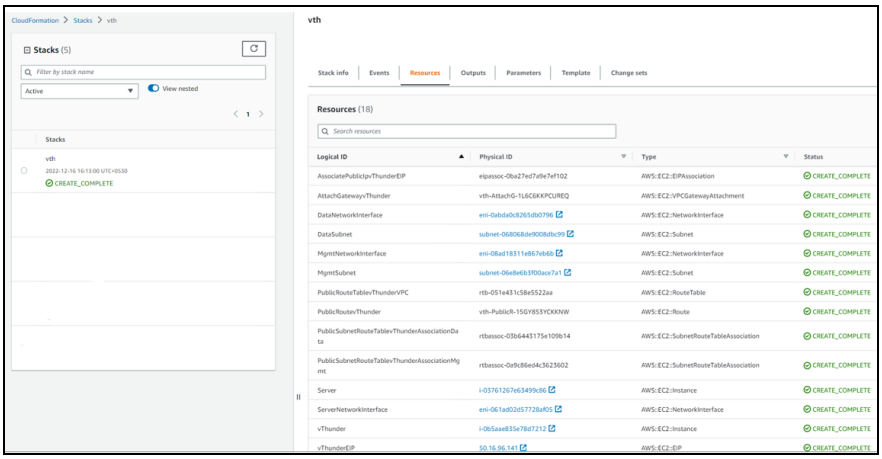
   b. In the **Parameters** section, enter or select the required value in the following fields:
      - **KeyPairName**: *your SSH key*
      - **TagValue**: **`a10-vthunder-adc`**
      - **Zone:** *your availability zone*
   c. Verify the other fields and change the values as required. (Optional)
6. Click **Next**.
   The **Configure stack options** window is displayed.
7. Verify the fields and change the values as required. (Optional)
8. Click **Next**.
   The **Review** window is displayed.

9. Verify if all the stack configurations are correct and then click **Submit**.

NOTE: The system may take a few minutes to create the resources. So, wait until the stack status is **CREATE_COMPLETE**.

10. Verify if all the above resources are created in the **AWS Management Console** > **CloudFormation** > **Stacks** > <stack_name> > **Resources** tab.

Figure 14 : Resource listing in the resource group



The vThunder instance is listed under **Resources** tab.

Access vThunder using GUI or CLI

vThunder instance can be accessed using any of the following ways:

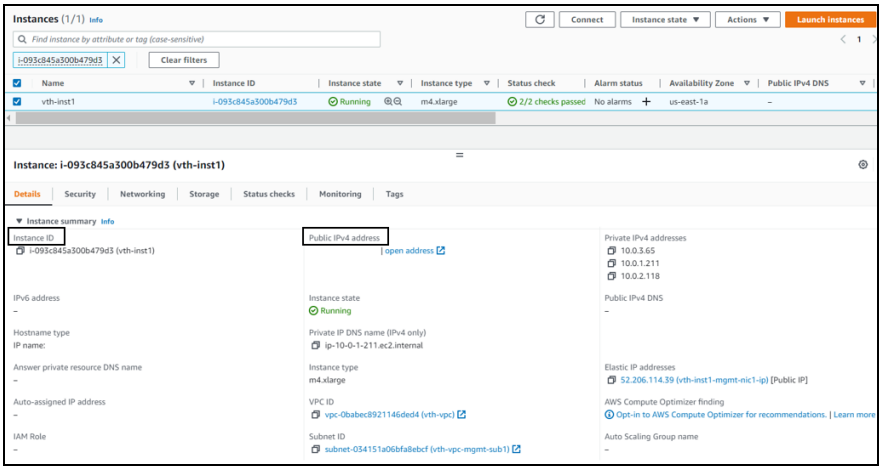
- Access vThunder using GUI
- Access vThunder using CLI

Access vThunder using GUI

To access vThunder instance using GUI, perform the following steps:

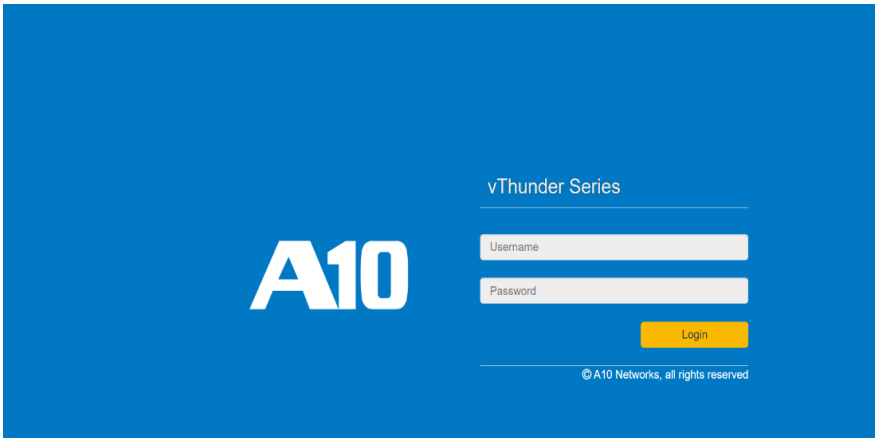
1. From **AWS Management Console**, navigate to **EC2** > **Instances**.
2. Select your instance name.
   Here, **`vth-inst1`** is the vThunder instance.

Figure 15 : vThunder instance



3. Copy the **Public IPv4 address** from the **Details** tab and replace the IP address in the below link:
   http://<vThunder_public_IPv4_address>
4. Open the updated link in any browser.
   The vThunder login window is displayed.

   Figure 16 : vThunder GUI



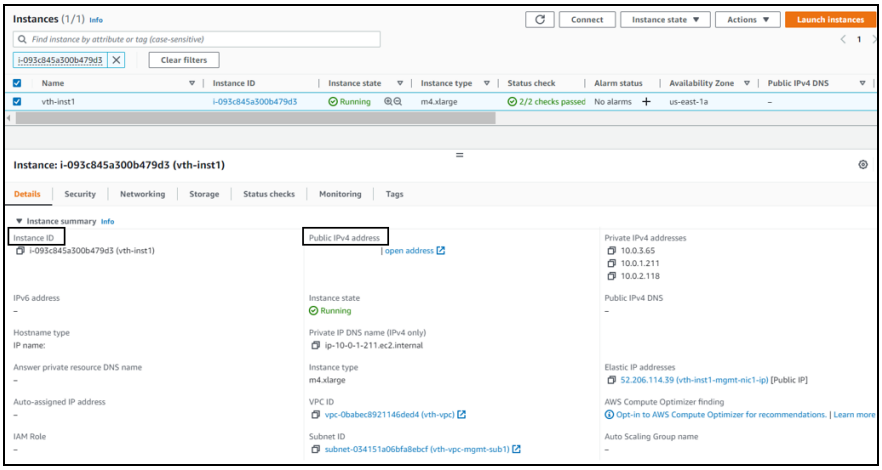
5. Enter the following credentials:
   - Username – admin
   - Password – *EC2 Instance ID*
     The home page is displayed if the entered credentials are correct.

Access vThunder using CLI

To access vThunder instance using CLI, perform the following steps:

1. From **AWS Management Console**, navigate to **EC2** > **Instances**.
2. Select your instance name.
   Here, **`vth-inst1`** is the vThunder instance.

   Figure 17 : vThunder instance



3. Copy the **Public IPv4 address** from the **Details** tab.
4. Open any SSH client and provide the following to establish a connection:
   - Hostname: Public IPv4 address
   - Username: admin
   - Key: SSH Key
5. Connect to the session.
6. In the SSH client session, enter the following commands:

```
vThunder(NOLICENSE)>enable <---Execute command--->
Password:                  <---just press Enter key--->
vThunder(NOLICENSE)#config <---Configuration mode--->
vThunder(config)(NOLICENSE)#
```

   The vThunder instance is ready to use.

Configure Server and Client Machine

To test the traffic flow via vThunder, create and configure a server machine and a client machine:

- Configure Server and Client Machine
- Configure Server and Client Machine

Configure Server Machine

To configure a server machine, perform the following steps:

1. From **AWS Management Console**, navigate to **EC2** > **Instances**.
2. Select your server instance name.
   Here, **`vth-glm-server`** is the server instance name.
3. Click **Connect**.
   A **Connect to instance** window with **EC2 Instance Connect** tab is displayed.
4. Click **Connect**.
   A **Terminal** window is displayed.
5. Run the following command to create an Apache Server virtual machine:
   ```
   sudo apt install apache2
   ```
   While the Apache server is getting installed, you get a prompt to continue further. Enter 'Y' to continue. After the installation is complete, a newline prompt is displayed.

Configure a Client Machine

To configure a client machine, perform the following steps:

1. From **AWS Management Console**, navigate to **EC2** > **Instances**.
2. Click **Launch Instances**.
   A **Launch an instance** window is displayed.
3. In the **Name and tags** section, enter an instance name.
   Here, enter **`vth-glm-client`** as the client instance name.

4. In the **Application and OS Images** section, select **Ubuntu**.
5. In the **Instance type** section, select the required instance type.
6. In the **Key pair (login)** field, select your SSH key.
7. In the **Network settings** section, click **Edit** to edit the following:
   - VPC: *your VPC*
     Here, **`vth-glm-vpc`** is the VPC.
   - Subnet: Data subnet
     Here, **`10.0.1.0/24`** is the data subnet value.
   - Auto-assign public IP: Enable
   - Firewall (security groups): Select existing security group
   - Common security groups: *your data security group*
     Here, **`vth-glm-vThunderSecurityGroupData`** is the security group.
8. Click **Launch instance**.

NOTE: The system may take a few minutes to launch the instance.

The client instance is displayed in the **Instances** list with the status as **Running**.

Configure vThunder as an SLB

The following topics are covered:

- Initial Setup
- Deploy vThunder as an SLB

Initial Setup

Before deploying vThunder on AWS cloud as an SLB, you need to configure the corresponding parameters in the CFT.

To configure the parameters, perform the following steps:

1. Navigate to the folder where you have downloaded the CFT, and open the CFT_TMPL_2NIC_1VM_GLM_CONFIG_SLB_SSL_GLM_PARAM.json with a text editor.

NOTE: Each parameter has a default value mentioned in the parameter file.

2. Configure the stack name created using CFT.

```
"stackDetails": {
  "value": [
    {
      "stackName": "vth"
    }
  ]},
```

3. Configure SLB server ports.

```
"server-list": {
  "value": [
    {
      "port-list": [
        {
          "port-number": 53,
          "protocol": "udp"
        },
        {
          "port-number": 80,
          "protocol": "tcp"
        },
        {
          "port-number": 443,
          "protocol": "tcp"
        }
      ]
    }
  ]
},
```

4. Configure Service Group List ports.
   The Service group name by default is “sg+port_number”. If you may want to change the service group name then after changing the name, change the names in the Virtual servers as well.

```
"serviceGroupList": {
  "value": [
    {
      "name": "sg443",
      "protocol": "tcp",
      "member-list": [
        {
          "port": 443
        }
      ]
    },
    {
      "name": "sg53",
      "protocol": "udp",
      "member-list": [
        {
          "port": 53
        }
      ]
    },
    {
      "name": "sg80",
      "protocol": "tcp",
      "member-list": [
        {
          "port": 80
        }
      ]
    }
  ]
},
```

5. Configure a Virtual Server.
   The virtual server default name is “vip”. It is the Private IP address of Ethernet1.

```
"virtualServerList": {
  "virtual-server-name": "vip",
  "metadata": {
    "description": "virtual server is using ethernet 1 ip address"
```

```
    },
    "value": [
      {
        "port-number": 53,
        "protocol": "udp",
        "auto": 1,
        "service-group": "sg53"
      },
      {
        "port-number": 80,
        "protocol": "http",
        "auto": 1,
        "service-group": "sg80"
      },
      {
        "port-number": 443,
        "protocol": "https",
        "auto": 1,
        "service-group": "sg443"
      }
    ]
  },
```

6. Configure SSL.

```
    "sslConfig": {
        "requestTimeOut": 40,
        "Path": "server.pem",
        "File": "server",
        "CertificationType": "pem"
      }
```

NOTE: By default, SSL configuration is disabled i.e., no SSL configuration is applied.

The server.pem file is available in the downloaded CFT folder. You can edit this file as required or use a different certificate file, but the path should be changed accordingly.

7. Configure GLM account details.
   You can get the **Entitlement Token** from GLM > **Licenses** > select your license > **Overview** > **Info** tab.

```
    "user_name": {
        "value": "xxxxxxxx@a10networks.com"
      },
      "user_password": {
        "value": "xxxxxxxx"
      },
      "entitlement_token": {
        "value": "xxxxxxxx"
      }
  }
```

8. Verify if all the configurations in the CFT_TMPL_2NIC_1VM_GLM_CONFIG_SLB_SSL_GLM_PARAM.json file are correct and then save the changes.

Deploy vThunder as an SLB

To deploy vThunder on AWS cloud as an SLB, perform the following steps:

1. From your command prompt, navigate to the folder where you have downloaded the CFT.
2. Run the following command to create vThunder SLB instance:

```
python ./CFT_TMPL_2NIC_1VM_GLM_CONFIG_SLB_SSL_GLM_2.py
```

   A message is prompted to upload the SSL certificate.

```
Do you want to upload ssl certificate(yes/no)?yes
configured ethernet ip
Configured server vth-server
Configured virtual servers
SSL Configured.
Configurations are saved on partition: shared
The certificate available in the sslConfig path is uploaded.
```

   If the SSL Certificate upload is successful, a message 'SSL Configured' is displayed.

Verify Deployment

To verify vThunder SLB deployment using CFT using CLI, perform the following steps:

1. Run the following command on vThunder:

```
vThunder(config)#show running-config
```

If the deployment is successful, the following SLB with GLM configuration is displayed:

```
!Current configuration: 0 bytes
!Configuration last updated at 17:36:35 GMT Wed Nov 30 2022
!Configuration last saved at 17:35:40 GMT Wed Nov 30 2022
!64-bit Advanced Core OS (ACOS) version 5.2.1, build 153 (Dec-11-
2020,14:34)
!
!
glm use-mgmt-port
glm enable-requests
glm token A10f771cecbe
!
interface ethernet 1
  enable
  ip address dhcp
!
!
slb server vth-server 10.0.1.120
  port 53 udp
  port 80 tcp
  port 443 tcp
!
slb service-group sg443 tcp
  member vth-server 443
!
slb service-group sg53 udp
  member vth-server 53
!
slb service-group sg80 tcp
```

```
  member vth-server 80
!
slb virtual-server vip use-if-ip ethernet 1
  port 53 udp
    source-nat auto
    service-group sg53
  port 80 http
    source-nat auto
    service-group sg80
  port 443 https
    source-nat auto
    service-group sg443
!
!
end
```

2. Run the following command on vThunder:

```
vThunder(config)#show pki cert
```

If the deployment is successful, the following SSL configuration is displayed:

```
Name    Type                        Expiration  Status
-----------------------------------------------------------------
server certificate Jan 28 12:00:00 2028 GMT [Unexpired, Bound]
```

Verify GLM

The application of license can be verified using any of the following ways:

- Verify License using GUI
- Verify License using CLI

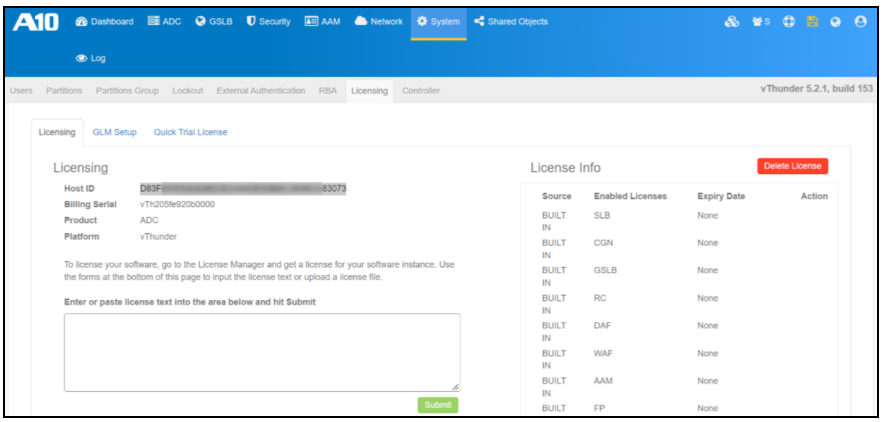
Verify License using GUI

To verify license using GUI, perform the following steps:

1. Log in to your vThunder instance using Public IPv4 address.
   Here, **vth-inst1** is the vThunder instance.

2. Navigate to **Profile** > **Setting** > **Licensing**.
3. Click the **Licensing** tab.
   If the license is successfully applied on vThunder, the Host ID is displayed.

   Figure 18 : GLM > Licensing window



Verify License using CLI

To verify license using CLI, perform the following steps:

1. Verify if NOLICENSE is removed from the vThunder prompt.
2. Run the following command on vThunder:

```
vThunder(config)#show license
```

   If the license is successfully applied on vThunder, the Host ID is displayed:

```
Host ID          : D83F****82EB633D****067DB84136****83073
```

3. Run the following command on vThunder instance:

```
vThunder(config)#show license-info
```

   If the license is successfully applied on vThunder, the following GLM configuration is displayed:

```
Host ID          : D83F****82EB633D****067DB84136****83073
USB ID           : Not Available
Billing Serials: vTh205fe920b0000
Token            : Not Available
Product          : ADC
Platform         : vThunder
```

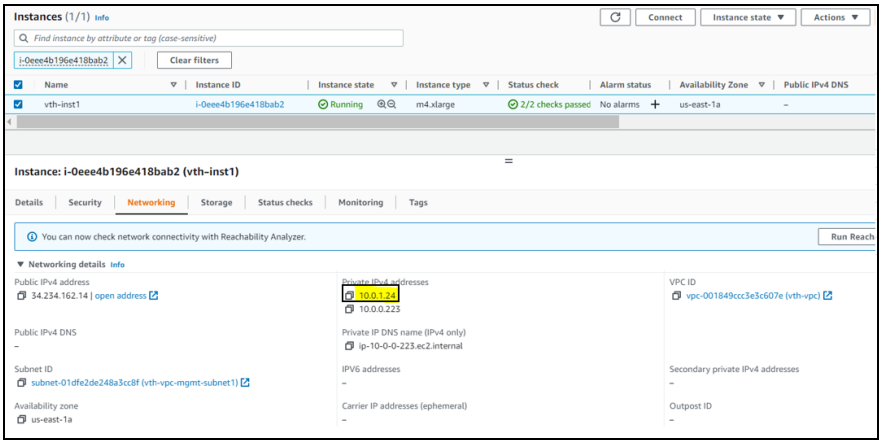
```
Burst           : Disabled
GLM Ping Interval In Hours : 24
-------------------------------------------------------------------------
-------
Enabled Licenses   Expiry Date (UTC)         Notes
-------------------------------------------------------------------------
-------
SLB                None
CGN                None
GSLB               None
RC                 None
DAF                None
WAF                None
AAM                None
FP                 None
WEBROOT            N/A              Requires an additional Webroot
license.
THREATSTOP         N/A              Requires an additional ThreatSTOP
license.
QOSMOS             N/A              Requires an additional QOSMOS
license.
WEBROOT_TI         N/A              Requires an additional Webroot
Threat Intel license.
IPSEC_VPN          N/A              Requires an additional IPsec VPN
license.
500 Mbps Bandwidth 20-January-2023
```

Verify Traffic Flow

To verify the traffic flow from client machine to server machine via vThunder, perform the following:

1. From **AWS Management Console**, navigate to **EC2** > **Instances**.
2. Select the vThunder instance name and then click the **Networking** tab. Here, **`vth-inst1`** is the instance name.
3. Copy the IP address of the data subnet under the **Private IPv4 address**. Here, **`10.0.1.24`** is the data subnet value.

Figure 19 : vThunder instance



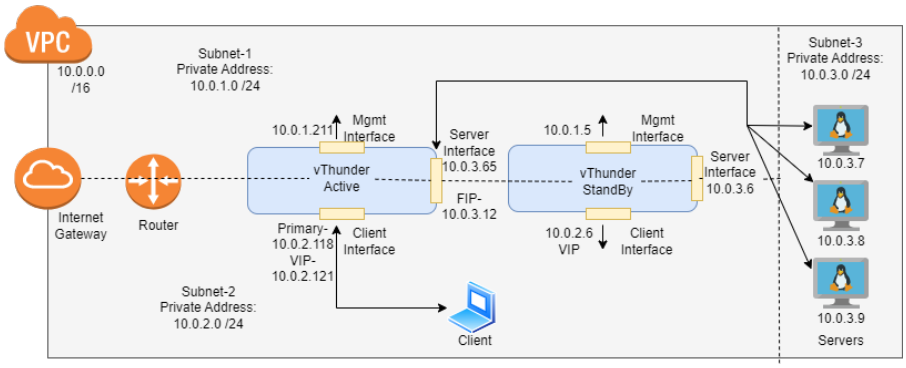
4. Select your client instance from the **Instances** list.
   Here, **`vth-client`** is the client instance name.
5. Click **Connect**.
   A **Connect to instance** window with **EC2 Instance Connect** tab is displayed.
6. Click **Connect**.
   A **Terminal** window is displayed.
7. Run the following command in the Terminal window to send the traffic from the client machine:
   ```
   curl <vThunder_instance_data_private_IPv4_Address>
   ```
8. Verify if a response is received.

Deploy CFT A10-vThunder_ADC-3NIC-2VM-HA

Using the A10-vThunder_ADC-3NIC-2VM-HA, two vThunder instances can be deployed containing:

- One management interface and two data interfaces each
- HA support

Figure 20 : 3NIC-2VM-HA Topology



The following topics are covered:

System Requirements

The CFT displays the default values when you download and save the files on your local machine. You can modify the default values based on your deployment.

You need to configure the following resources to deploy vThunder on the AWS cloud:

Table 7 : System Requirements

Resource Name	Description	Default Value
Stack	A new stack is created with the specified name and location.	Here, the stack name used is **vth**.
Network Interface Card [NIC]	Two types of interfaces are created for each vThunder instance: • One management interface • Two data interfaces	**vth-inst1-mgmt-nic1** **vth-inst1-data-nic2** **vth-inst1-data-nic3** **vth-inst2-mgmt-nic1** **vth-inst2-data-nic2** **vth-inst2-data-nic3**
Subnet	Three subnets are created with an address prefix each.	**vth-vpc-mgmt-sub1** **vth-vpc-data-sub1** **vth-vpc-data-sub2**
Virtual Private Network [VCN]	A virtual private network is assigned to the virtual machine instance.	**vth-vpc** Address prefix for virtual network: 10.0.0.0/16
Elastic Public IP	Elastic Public IPs are created and attached to the management interface of the vThunder instances.	**vth-inst1-mgmt-nic1-ip** **vth-inst2-mgmt-nic1-ip**
Security Group	Two security groups are created: • One security group for the	Here, the tag names are: **vth-sg-mgmt**

Resource Name	Description	Default Value
	management interface. • One security group for the data interface. **Logical name**: • vThunderSecurityGroupMgmt • vThunderSecurityGroupData	`vth-sg-data`
vThunder Instance	Two vThunder EC2 instances are created: • One active • One standby **Default type**: m4.xlarge (40 Gb memory) Table 8 lists the supported instance types.	`vth-inst1` `vth-inst2`
Server Instance	One Ubuntu Server instance is created. **Default Size**: t2.micro	`vth-server`

Supported Instance Types

Table 8 : List of Supported Instance Types

Instance	vCPU	Memory	Number of Network Interfaces
c4.xlarge	4	7680	4
c4.4xlarge	16	30720	8
c4.8xlarge	36	61440	8
d2.xlarge	4	31232	4
d2.2xlarge	8	62464	4
d2.4xlarge	16	124928	8
d2.8xlarge	36	249856	8

Instance	vCPU	Memory	Number of Network Interfaces
m4.xlarge	4	16384	4
m4.2xlarge	8	32768	4
m4.4xlarge	16	65536	8
m4.10xlarge	40	163840	8
i2.xlarge	4	31232	4
i2.2xlarge	8	62464	4
i2.4xlarge	16	124928	8
i2.8xlarge	32	249856	8
c5d.large	2	4096	3
c5d.9xlarge	36	73728	8
c5d.2xlarge	8	32768	4
c5d.4xlarge	16	73728	8
c5.xlarge	4	8192	4
c5.2xlarge	8	16384	4
c5.4xlarge	16	32768	8
c5.9xlarge	36	73728	8
g3.4xlarge	16	124928	8
g3.8xlarge	32	249856	8
i3.large	2	15616	3
i3.xlarge	4	31232	4
i3.2xlarge	8	62464	4
i3.4xlarge	16	124928	8
i3.8xlarge	32	249856	8
m5d.large	2	8192	3
m5d.xlarge	4	16384	4
m5d.2xlarge	8	32768	4
m5d.4xlarge	16	65536	8
m5.large	2	8192	3

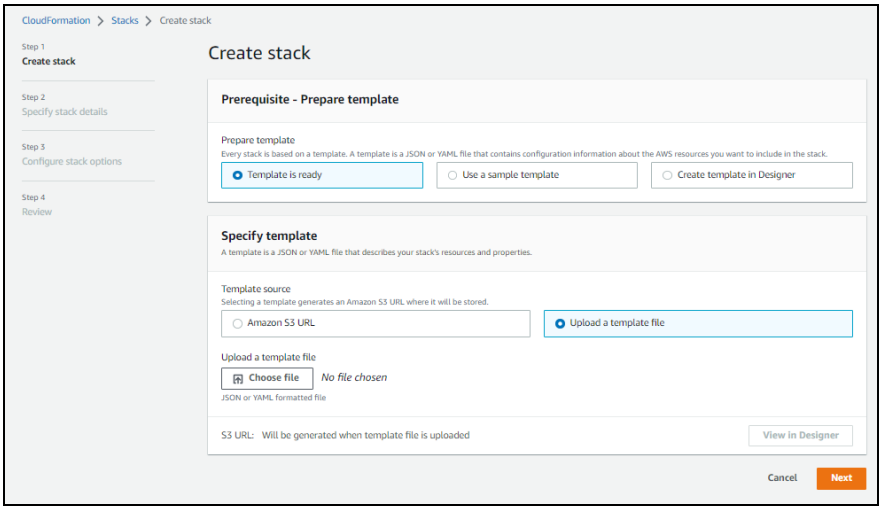
Instance	vCPU	Memory	Number of Network Interfaces
m5.xlarge	4	16384	4
m5.2xlarge	8	32768	4
m5.4xlarge	16	65536	8
r5d.large	2	16384	3
r5d.xlarge	4	32768	4
r5d.2xlarge	8	65536	4
r5d.4xlarge	16	131072	8
r5.large	2	16384	3
r5.xlarge	4	32768	4
r5.2xlarge	8	65536	4
r5.4xlarge	16	131072	8
r4.large	2	15616	3
r4.xlarge	4	31232	4
r4.2xlarge	8	62464	4
r4.4xlarge	16	124928	8
r4.8xlarge	32	249856	8
t3.medium	2	4096	3
t3.large	2	8192	3
t3.xlarge	4	16384	4
t3.2xlarge	8	32768	4
z1d.large	2	16384	3
z1d.xlarge	4	32768	4
z1d.2xlarge	8	65536	4
z1d.3xlarge	12	98304	8
z1d.6xlarge	24	196608	8

Create vThunder Instances

To create vThunder instances, perform the following steps:

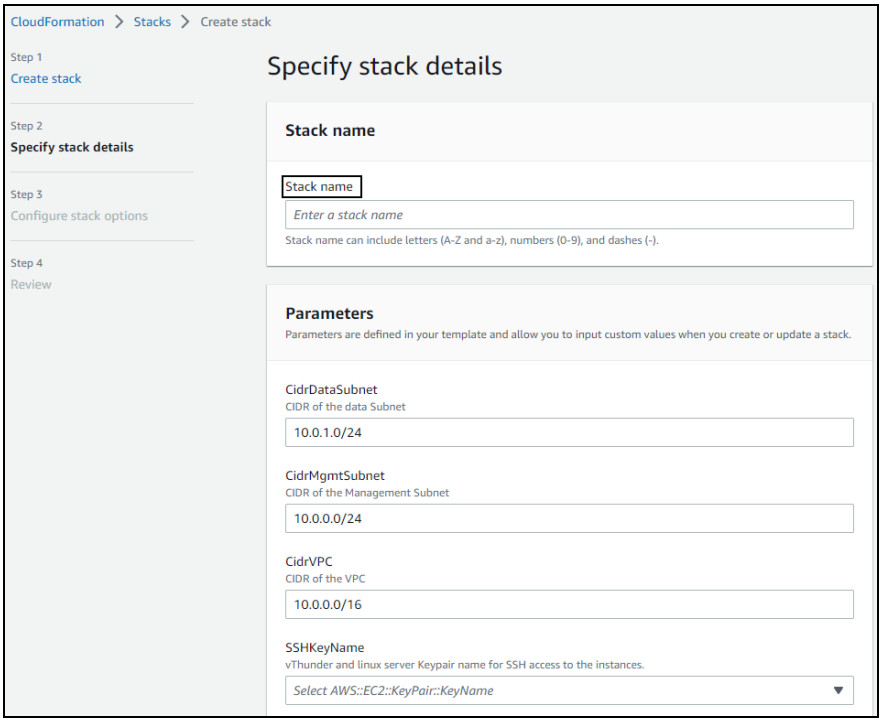
1. From **AWS Management Console**, navigate to **CloudFormation** > **Stacks** > **Create Stack** > **With new resources (standard)**.
   The **Create stack** window is displayed.

   Figure 21 : Create stack window



2. In the **Prerequisite - Prepare template** section, select **Template is ready** option. After the selecting this option, the **Specify template** section is displayed.
3. In the **Specify template** section, select **Upload a template file** and click **Choose file** to browse and upload the following template file from the downloaded CFT folder:
   CFT_TMPL_3NIC_2VM_HA_1.json
   The selected template file name is displayed as the chosen file.
4. Click **Next**.
   The **Specify stack details** window is displayed.

Figure 22 : Specify stack details window



5. In the **Specify stack details** window, enter or select the following:
   a. In the **Stack name** section, enter a **Stack name**.

      Here, **`vth`** is the stack name.

   b. In the **Parameters** section, enter or select the required value in the following fields:

      - **KeyPairName**: *your SSH key*
      - **TagValue**: **`a10-vthunder-adc`**
      - **Zone:** *your availability zone*

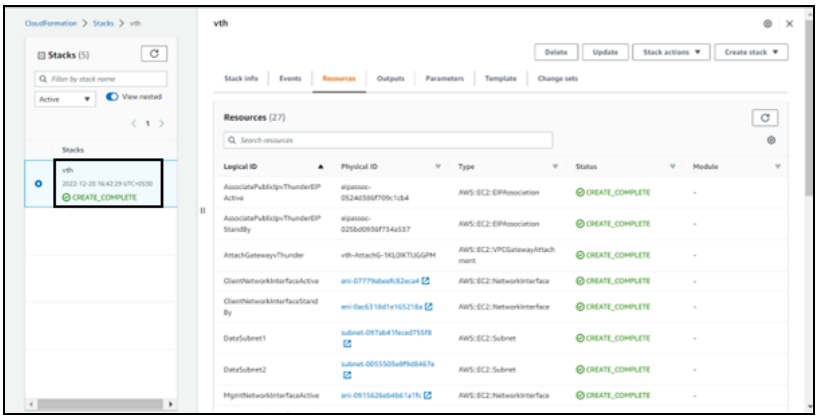
   c. Verify the other fields and change the values as required. (Optional)
6. Click **Next**.
   The **Configure stack options** window is displayed.
7. Verify the fields and change the values as required. (Optional)
8. Click **Next**.
   The **Review** window is displayed.

9. Verify if all the stack configurations are correct and then click **Submit**.

NOTE: The system may take a few minutes to create the resources. So, wait until the stack status is **CREATE_COMPLETE**.

10. Verify if all the above resources are created in the **AWS Management Console** > **CloudFormation** > **Stacks** > <stack_name> > **Resources** tab.

Figure 23 : Resource listing in the resource group



The two vThunder instances are listed under **Resources** tab.

Access vThunder using GUI or CLI

vThunder instances can be accessed using any of the following ways:

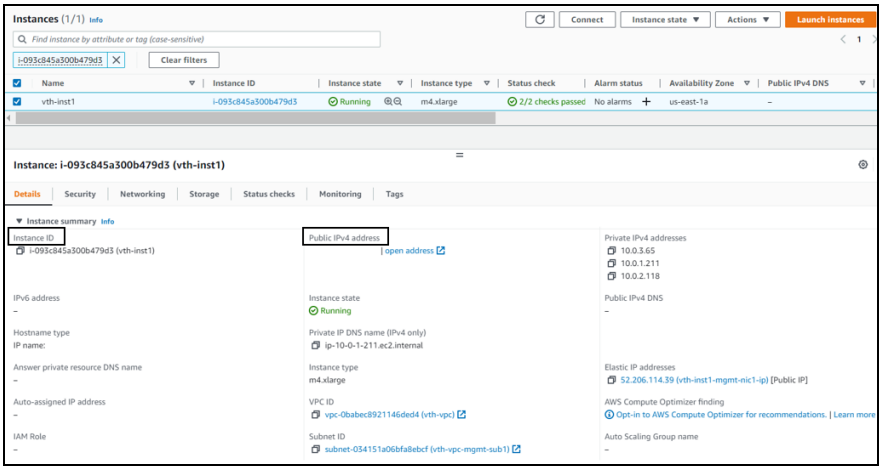
- Access vThunder using GUI
- Access vThunder using CLI

Access vThunder using GUI

To access vThunder instances using GUI, perform the following steps:

1. From **AWS Management Console**, navigate to **EC2** > **Instances**.
2. Select your instance name.
   Here, **`vth-inst1`**, **`vth-inst2`** are the vThunder instances.

Figure 24 : vThunder instance

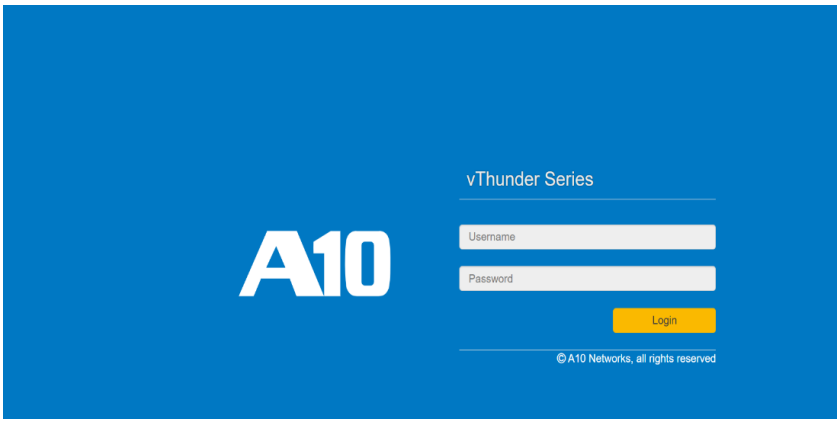


3. For each vThunder instance, perform the following steps:

   a. Copy the **Public IPv4 address** from the **Details** tab and replace the IP address in the below link:
   http://<vThunder_public_IPv4_address>

   b. Open the updated link in any browser.
   The vThunder login window is displayed.

   Figure 25 : vThunder GUI



   c. Enter the following credentials:

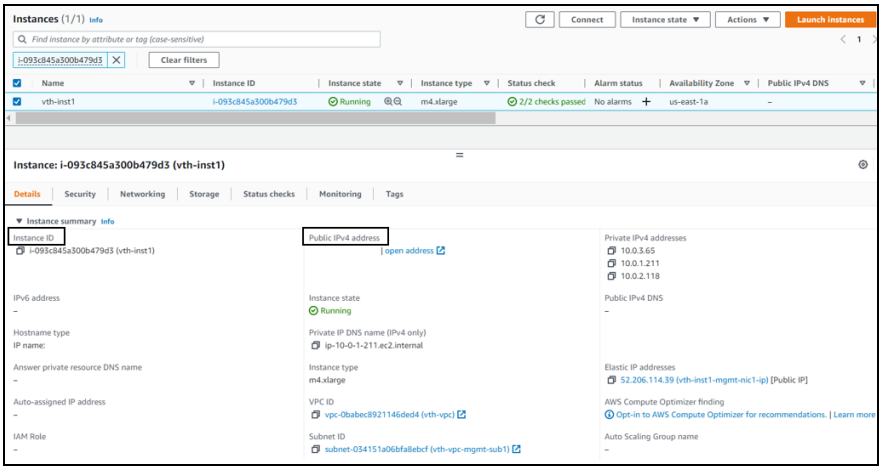
   - Username – admin
   - Password – *EC2 Instance ID*
     The home page is displayed if the entered credentials are correct.

Access vThunder using CLI

To access vThunder instances using CLI, perform the following steps:

1. From **AWS Management Console**, navigate to **EC2** > **Instances**.
2. Select your instance name.
   Here, **`vth-inst1, vth-inst2`** are the vThunder instances.

   Figure 26 : vThunder instance



3. For each vThunder instance, perform the following steps:
   a. Copy the **Public IPv4 address** from the **Details** tab.
   b. Open any SSH client and provide the following to establish a connection:
      - Hostname: Public IPv4 address
      - Username: admin
      - Key: SSH Key
   c. Connect to the session.
   d. In the SSH client session, run the following commands:

```
vThunder(NOLICENSE)>enable <---Execute command--->
Password:                  <---just press Enter key--->
vThunder(NOLICENSE)#config <---Configuration mode--->
vThunder(config)(NOLICENSE)#
```

   The vThunder instances are ready to use.

Configure Server and Client Machine

To test the traffic flow via vThunder, create and configure a server machine and a client machine:

Configure Server Machine

To configure a server machine, perform the following steps:

1. From **AWS Management Console**, navigate to **EC2** > **Instances**.
2. Select your server instance name.
   Here, **`vth-server`** is the server instance name.
3. Click **Connect**.
   A **Connect to instance** window with **EC2 Instance Connect** tab is displayed.
4. Click **Connect**.
   A **Terminal** window is displayed.
5. Run the following command in the Terminal window to create an Apache Server virtual machine:
   ```
   sudo apt install apache2
   ```
   While the Apache server is getting installed, you get a prompt to continue further. Enter 'Y' to continue. After the installation is complete, a newline prompt is displayed.

Configure a Client Machine

To configure a client machine, perform the following steps:

1. From **AWS Management Console**, navigate to **EC2** > **Instances**.
2. Click **Launch Instances**.
   A **Launch an instance** window is displayed.
3. In the **Name and tags** section, enter an instance name.
   Here, enter **`vth-client`** as the client instance name.

4. In the **Application and OS Images** section, select **Ubuntu**.
5. In the **Instance type** section, select the required instance type.
6. In the **Key pair (login)** field, select your SSH key.
7. In the **Network settings** section, click **Edit** to edit the following:
   - VPC: *your VPC*
     Here, `vth-vpc` is the VPC.
   - Subnet: Data subnet
     Here, `10.0.2.0/24` is the data subnet value.
   - Auto-assign public IP: Enable
   - Firewall (security groups): Select existing security group
   - Common security groups: *your data security group*
     Here, `vth-vThunderSecurityGroupData` is the security group.
8. Click **Launch instance**.

NOTE: The system may take a few minutes to launch the instance.

The client instance is displayed in the **Instances** list with the status as **Running**.

Import AWS Access Keys

In high availability configuration, IP switching happens between the two vThunder instances. To enable the IP switching, the AWS keys should be imported on these vThunder instances.

For importing AWS keys on a vThunder instance, perform the following steps:

- Add 'All TCP' rule in Security Group
- Create AWS access keys file
- Create a new FTP server
- Upload access keys on FTP server
- Upload access keys on vThunder instances

Add 'All TCP' rule in Security Group

To add 'All TCP' rule in the security group, perform the following steps:

1. From **AWS Management Console**, navigate to **EC2** > **Security Groups**.
2. Select your security group which got created for data interface.
   Here, **`vth-sg-data`** (**`vThunderSecurityGroupData`**) is the security group created for data interface.
   The **Security Groups** window for data interface is displayed.
3. Select the **Inbound rules** tab.
4. Click **Edit inbound rules**.
   The Edit inbound rules window is displayed.
5. Click **Add rule**.
   A rule row is added.
6. Select **All TCP** in the **Type** field.
7. Specify **0.0.0.0/0** in CIDR notation field.
8. Click **Save rules**.
   The rule is added in the security group.

NOTE: As the 'All TCP' rule is only used to transfer the AWS access keys to a vThunder instance, you can remove it from the security group after the AWS access keys are uploaded on both vThunder instances.

Create AWS access keys file

To create AWS access keys file, perform the following steps:

1. Open a text document on your local machine and copy the following information to this document:

```
[default]
aws_access_key_id = your_aws_access_key_id
aws_secret_access_key = your_aws_secret_access_key
```

2. Update the access key ID and secret access key as per your AWS account.
3. Save the text file with a name.
   Here, **aws_access_key.txt** is the access keys file name.

Create a new FTP server

If you don't have an existing FTP server, create a new FTP server.

To create FTP server, perform the following steps:

1. From **AWS Management Console**, navigate to **EC2** > **Instances**.
2. Click **Launch Instances**.
   A **Launch an instance** window is displayed.
3. In the **Name and tags** section, enter an instance name.
   Here, enter `vth-stack-keys` as the FTP server name.
4. In the **Application and OS Images** section, select **Ubuntu**.
5. In the **Instance type** section, select the required instance type.
6. In the **Key pair (login)** field, select your SSH key.
7. In the **Network settings** section, click **Edit** to edit the following:
   - VPC: *your VPC*
     Here, `vth-vpc` is the VPC.
   - Subnet: Data subnet 1
     Here, `10.0.2.0/24` is the data subnet 1 value.
   - Auto-assign public IP: Enable
   - Firewall (security groups): Select existing security group
   - Common security groups: *your data security group*
     Here, `vth-vThunderSecurityGroupData` is the security group.
8. Click **Launch instance**.

   NOTE: It may take the system a few minutes to launch the instance.

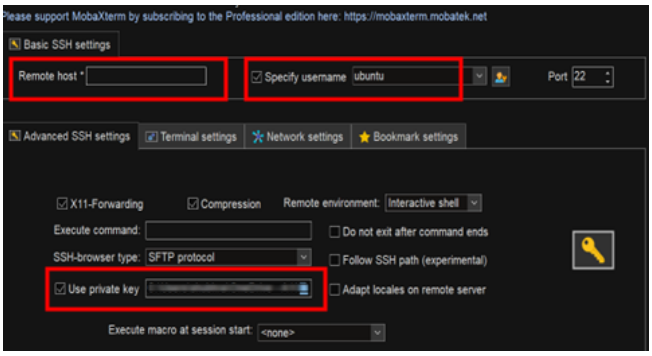
   The FTP server instance is displayed in the **Instances** list with status as **Running**.
9. Select your FTP server instance name.
   Here, you can enter `vth-stack-keys` is the FTP server instance.
10. Copy the **Public IPv4 address** from the **Details** tab.
11. Open any SSH client and enter the login details.
    For example: If you are using MobaXterm, perform the following steps:

a. Enter the public IPv4 address in the **Remote host** field.

b. Enter **ubuntu** in the **Specify username** field.

c. Select SSH key in the **Use private key** field.

Figure 27 : MobaXterm Basic SSH settings



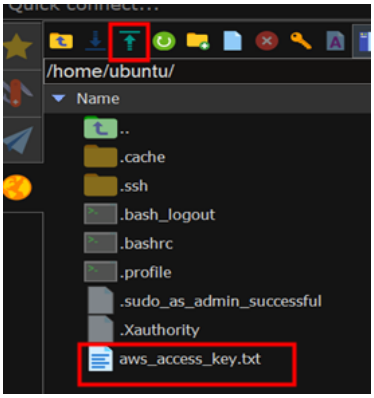
12. Connect to the FTP server instance.
If the connection is established, the FTP server instance is created.

Upload access keys on FTP server

To upload access keys on the FTP server, perform the following steps:

1. Upload the access keys file at the '/home/ubuntu' location of the FTP server. Here, **aws_access_key.txt** is the access key file.

Figure 28 : MobaXterm Upload



2. Verify if the file had been uploaded successfully.
3. Run the following command in the Terminal window to provide the uploaded file

with read and write permission:

```
sudo chmod 777 <access_key_filename>
```

Example:

```
sudo chmod 777 aws_access_key.txt
```

4. Run the following commands to update all the package information:

```
sudo apt update && sudo apt upgrade
```

NOTE: If a message "Daemons using outdated libraries Which services should be restarted?" is prompted, click **Ok** and press **Enter**.

5. Run the following command to install FTP server:

```
sudo apt-get install vsftpd
```

6. Run the following commands to start and enable the FTP server:

```
sudo systemctl start vsftpd
sudo systemctl enable vsftpd
```

7. Run the following commands to add a new user and a password:

```
sudo useradd -m <username>
sudo passwd <username>
New password: <----Enter password---->
Retype new password: <----Re-enter password---->
passwd: password updated successfully
```

Example:

```
sudo useradd -m vth-user
sudo passwd vth-user
New password: <----Enter password---->
Retype new password: <----Re-enter password---->
passwd: password updated successfully
```

8. Navigate to the 'cd /etc/' folder and run the following command to open the **`vsftpd.conf`** file:

```
sudo vi vsftpd.conf
```

9. Press **Esc** and enter **i** to enable edit/insert mode for **`vsftpd.conf`** file.
10. Provide write permission to the uploaded file:

```
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
```

11. After the changes, press **Esc** then type **:wq** to save the changes and exit.
12. Run the following commands to restart the FTP server instance:

```
sudo systemctl restart vsftpd
cd /home/ubuntu
ftp localhost
Connected to localhost.
220 (vsFTP 3.0.5)
Name (localhost:ubuntu): <username>
331 Please specify the password.
Password: <----Enter password---->
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Example:

```
sudo systemctl restart vsftpd
cd /home/ubuntu
ftp localhost vth-user vth-user
Connected to localhost.
220 (vsFTP 3.0.5)
Name (localhost:ubuntu): vth-user
331 Please specify the password.
Password: <----Enter password---->
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

13. Put the access keys in the FTP server instance:

```
ftp> put <access_key_filename>
```

Example:

```
ftp> put aws_access_key.txt
```

The AWS access keys are uploaded on the FTP server instance.

Upload access keys on vThunder instances

To upload access keys on both vThunder instances, perform the following steps:

1. Run the following commands on each vThunder instance using CLI to import the access keys:

```
vThunder(config)#admin admin
vThunder(config-admin:admin)#aws-accesskey import use-mgmt-port
ftp://username@publicipoftheinstancewithkeys
               <---Update with the username and the Public IP of
FTP server instance---->
Password []?    <---Provide the password set for FTP server instance
user---->
File name [/]?aws_access_key.txt <---Provide the access keys file name-
--->
vThunder(config-admin:admin)#
```

2. Run the following commands on each vThunder instance to verify the access keys upload:

```
vThunder(config-admin:admin)#aws-accesskey show
[default]
aws_access_key_id = your_aws_access_key_id
aws_secret_access_key = your_aws_secret_access_key
vThunder(config-admin:admin)#exit
vThunder(config)#
vThunder(config)#
vThunder-Active(config)#
```

NOTE: Once the AWS access keys are transferred to vThunder, make sure to terminate the FTP server from **AWS Management Console** > **EC2** > **Instances**.

Configure vThunder as an SLB with HA

The following topics are covered:

Initial Setup

Before deploying vThunder on AWS cloud as an SLB with HA, you need to configure the corresponding parameters in the CFT.

To configure the parameters, perform the following steps:

1. Navigate to the folder where you have downloaded the CFT, and open the CFT_TMPL_3NIC_2VM_HA_CONFIG_SLB_SSL_HA_PARAM.json with a text editor.

   NOTE: Each parameter has a default value mentioned in the parameter file.

2. Configure the stack name created using CFT.

```
"stackDetails": {
  "value": [
    {
      "stackName": "vth"
    }
  ]},
```

3. Configure SLB server ports.

```
"server-list": {
  "value": [
    {
      "metadata": {
        "description": "SLB server host/fqdn-name."
      },
      "port-list": [
        {
          "port-number": 53,
          "protocol": "udp"
```

```
            },
            {
              "port-number": 80,
              "protocol": "tcp"
            },
            {
              "port-number": 443,
              "protocol": "tcp"
            }
          ]
        }
      ]
    },
```

4. Configure Service Group ports.
   The Service group name by default is “sg+port_number”. If you may want to change the service group name then after changing the name, change the names in the Virtual servers as well.

```
    "serviceGroupList": {
      "value": [
        {
          "name": "sg443",
          "protocol": "tcp",
           "member-list": [
            {
              "port": 443
            }
          ]
        },
        {
          "name": "sg53",
          "protocol": "udp",
          "member-list": [
            {
              "port": 53
            }
          ]
        },
```

```
        {
          "name": "sg80",
          "protocol": "tcp",
          "member-list": [
            {
              "port": 80
            }
          ]
        }
      ]
    },
```

5. Configure a Virtual Server and its ports.
   The virtual server default name is “vip”. It is the Private IP address of Ethernet1.

```
     "virtualServerList": {
        "virtual-server-name": "vip",
        "metadata": {
          "description": "virtual server is using ethernet 1 ip
address"
        },
        "value": [
          {
            "port-number": 53,
            "protocol": "udp",
            "auto": 1,
            "service-group": "sg53"
          },
          {
            "port-number": 80,
            "protocol": "http",
            "auto": 1,
            "service-group": "sg80"
          },
          {
            "port-number": 443,
            "protocol": "https",
            "auto": 1,
            "service-group": "sg443"
```

```
            }
        ]
    },
```

6. Configure DNS.

```
        "dns": {
            "value": "8.8.8.8"
        },
```

7. Configure a Network Gateway IP.
   The default value of network gateway IP address is 10.0.1.1 as this is the first IP address of the data subnet 1 configuration.

```
        "rib-list": [
            {
            "ip-dest-addr":"0.0.0.0",
            "ip-mask":"/0",
            "ip-nexthop-ipv4": [
                {
              "ip-next-hop": "10.0.1.1"
            },
            {
              "ip-next-hop": "10.0.2.1"
            }
            ]
            }
        ],
```

8. Set VRRP-A.

```
        "vrrp-a": {
            "set-id":1
        },
```

9. Set a Terminal Idle Timeout.

```
        "terminal": {
            "idle-timeout":0
        },
```

10. Set VRID.
    For vThunder instance 1, the default priority is 100, and for vThunder instance 2, it is 99 (100-1).

```
        "vrid-list": [
       {
         "vrid-val": 0,
         "blade-parameters": {
           "priority": 100
         }
       }
     ],
```

11. Configure SSL.

```
    "sslConfig": {
        "requestTimeOut": 40,
        "Path": "server.pem",
        "File": "server",
        "CertificationType": "pem"
      }
```

NOTE: By default, SSL configuration is disabled i.e., no SSL configuration is applied.

The server.pem file is available in the downloaded CFT folder. You can edit this file as required or use a different certificate file, but the path should be changed accordingly.

12. Verify if all the configurations in the CFT_TMPL_3NIC_2VM_HA_CONFIG_SLB_SSL_HA_PARAM.json file are correct and then save the changes.

Deploy vThunder as an SLB

To deploy vThunder on AWS cloud as an SLB, perform the following steps:

1. From your command prompt, navigate to the folder where you have downloaded the CFT.
2. Run the following command to create vThunder SLB instances:

```
python ./CFT_TMPL_3NIC_2VM_HA_CONFIG_SLB_SSL_HA_2.py
```

A message is prompted to upload the SSL certificate and configure HA.

```
Do you want to upload ssl certificate(yes/no)?yes
Do you want to configure HA(yes/no)?yes
```

```
configured ethernet ip
Configured server vth-server
Configured virtual servers
SSL Configured.
Configured primary dns
Configured IP Route
Configured Vrrp A common
Configured idle timeout
Configured vrrp rid
Configured peer group
Configurations are saved on partition: shared
The certificate available in the sslConfig path is uploaded.
```

If the SSL Certificate upload is successful, a message 'SSL Configured' is displayed.

Verify Deployment

To verify vThunder SLB deployment using CFT, perform the following steps:

1. Run the following command on vThunder instance 1 using CLI:

```
vThunder-Active(config)(NOLICENSE)#show running-config
```

If the deployment is successful, the following SLB configuration is displayed:

```
!Current configuration: 349 bytes
!Configuration last updated at 10:56:58 GMT Fri Jan 6 2023
!Configuration last saved at 10:53:34 GMT Fri Jan 6 2023
!64-bit Advanced Core OS (ACOS) version 5.2.1, build 153 (Dec-11-
2020,16:36)
!
vrrp-a common
  device-id 1
  set-id 1
  enable
!
terminal idle-timeout 0
!
ip dns primary 8.8.8.8
```

```
!
!
interface ethernet 1
  enable
  ip address dhcp
!
interface ethernet 2
  enable
  ip address dhcp
!
vrrp-a vrid 0
  floating-ip 10.0.3.12
  blade-parameters
    priority 99
!
vrrp-a peer-group
  peer 10.0.2.117
  peer 10.0.2.173
!
ip route 0.0.0.0 /0 10.0.1.1
ip route 0.0.0.0 /0 10.0.2.1
!
slb server vth-server 10.0.3.23
  port 53 udp
  port 80 tcp
  port 443 tcp
!
slb service-group sg443 tcp
  member vth-server 443
!
slb service-group sg53 udp
  member vth-server 53
!
slb service-group sg80 tcp
  member vth-server 80
!
slb virtual-server vip 10.0.2.121
```

```
  port 53 udp
    source-nat auto
    service-group sg53
  port 80 http
    source-nat auto
    service-group sg80
  port 443 https
    source-nat auto
    service-group sg443
!
!
end
!Current config commit point for partition 0 is 0 & config mode is
classical-mode
vThunder-Active(config)(NOLICENSE)#
```

2. Run the following command on vThunder instance 1 using CLI:

```
vThunder-Active(config)(NOLICENSE)#show pki cert
```

If the deployment is successful, the following SSL configuration is displayed:

```
Name    Type                        Expiration  Status
-----------------------------------------------------------------
server certificate Jan 28 12:00:00 2028 GMT [Unexpired, Bound]
```

At this point, the vThunder instance 2 has the following prompt:

```
vThunder-Standby(config)(NOLICENSE)#
```

Figure 29 : vThunder instance 1 - Active

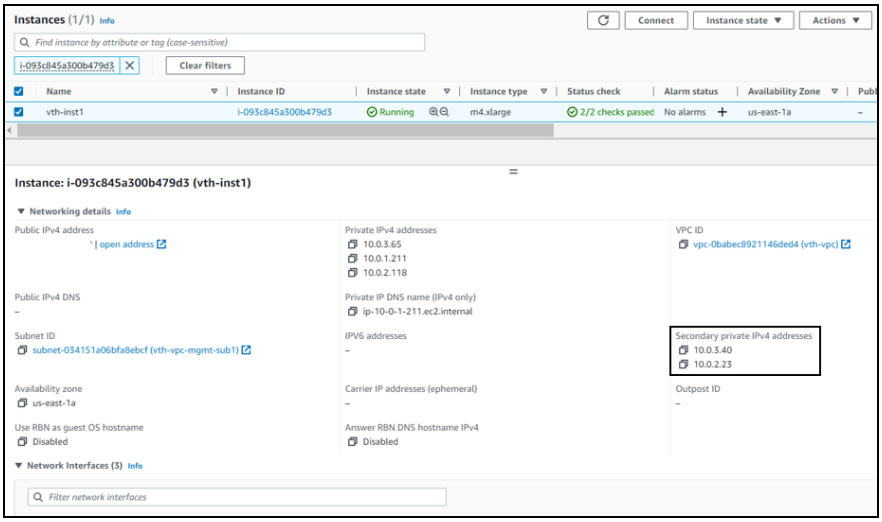
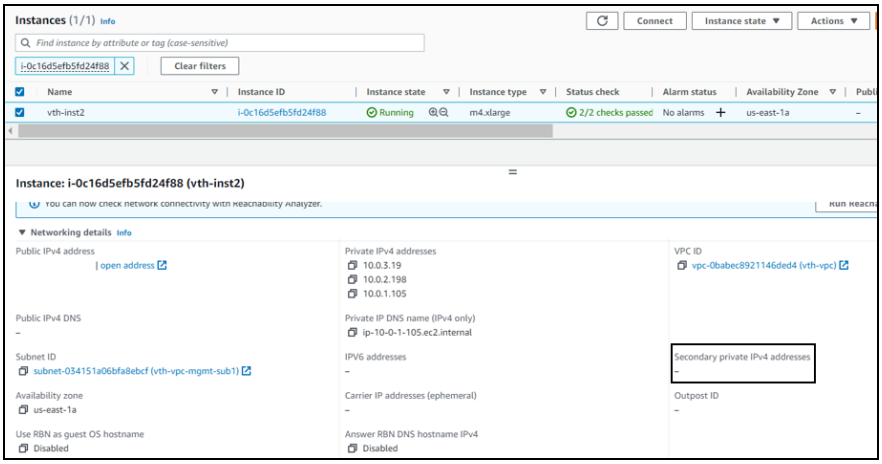


Figure 30 : vThunder instance 2 - Standby



3. Run the following command on vThunder instance 1 using CLI:

```
vThunder-Active(config)(NOLICENSE)#vrrp-a force-self-standby enable
vThunder-Active(config)(NOLICENSE)#
vThunder-ForcedStandby(config)(NOLICENSE)#
```

At this point, IP switching occurs and the vThunder instance 2 prompt becomes:

```
vThunder-Active(config)(NOLICENSE)#
```

4. Run the following command on vThunder instance 2 using CLI:

```
vThunder-Active(config)(NOLICENSE)#show running-config
```

If the deployment is successful, the following SLB configuration is displayed:

```
!Current configuration: 282 bytes
!Configuration last updated at 10:53:35 GMT Fri Jan 6 2023
!Configuration last saved at 10:53:37 GMT Fri Jan 6 2023
!64-bit Advanced Core OS (ACOS) version 5.2.1, build 153 (Dec-11-
2020,16:36)
!
vrrp-a common
  device-id 2
  set-id 1
  enable
!
terminal idle-timeout 0
!
ip dns primary 8.8.8.8
!
!
interface ethernet 1
  enable
  ip address dhcp
!
interface ethernet 2
  enable
  ip address dhcp
!
vrrp-a vrid 0
  floating-ip 10.0.3.12
  blade-parameters
    priority 98
!
vrrp-a peer-group
  peer 10.0.2.186
  peer 10.0.2.6
!
ip route 0.0.0.0 /0 10.0.1.1
```

```
ip route 0.0.0.0 /0 10.0.2.1
!
slb server vth-server 10.0.3.23
  port 53 udp
  port 80 tcp
  port 443 tcp
!
slb service-group sg443 tcp
  member vth-server 443
!
slb service-group sg53 udp
  member vth-server 53
!
slb service-group sg80 tcp
  member vth-server 80
!
slb virtual-server vip 10.0.2.23
  port 53 udp
    source-nat auto
    service-group sg53
  port 80 http
    source-nat auto
    service-group sg80
  port 443 https
    source-nat auto
    service-group sg443
!
sflow setting local-collection
!
sflow collector ip 127.0.0.1 6343
!
!
end
!Current config commit point for partition 0 is 0 & config mode is
classical-mode
vThunder-Active(config)(NOLICENSE)#
```

5. Run the following command on vThunder instance 2 using CLI:

```
vThunder-Active(config)(NOLICENSE)#show pki cert
```

If the deployment is successful, the following SSL configuration is displayed:

```
Name     Type                          Expiration  Status
-----------------------------------------------------------------
server certificate Jan 28 12:00:00 2028 GMT [Unexpired, Bound]
```

Figure 31 : vThunder instance 1 - Standby

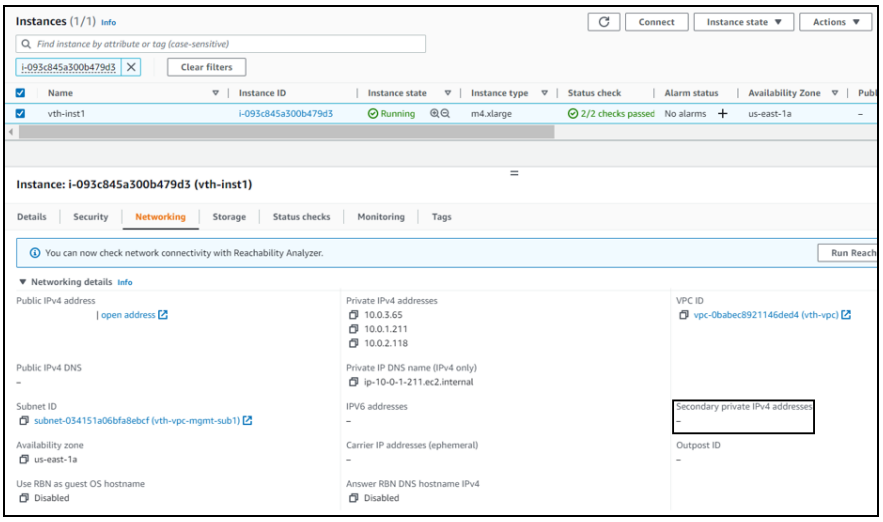
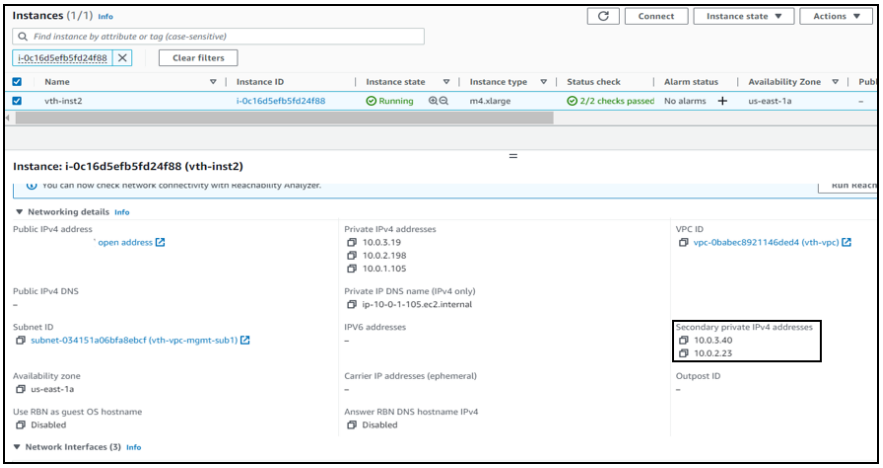


Figure 32 : vThunder instance 2 - Active



6. If you want to make vThunder instance 1 active, run the following command on vThunder instance 2 using CLI:

```
vThunder-Active(config)(NOLICENSE)#vrrp-a force-self-standby disable
vThunder-Active(config)(NOLICENSE)#
vThunder-ForcedStandby(config)(NOLICENSE)#
```

At this point, IP switching occurs and the vThunder instance 1 prompt becomes:

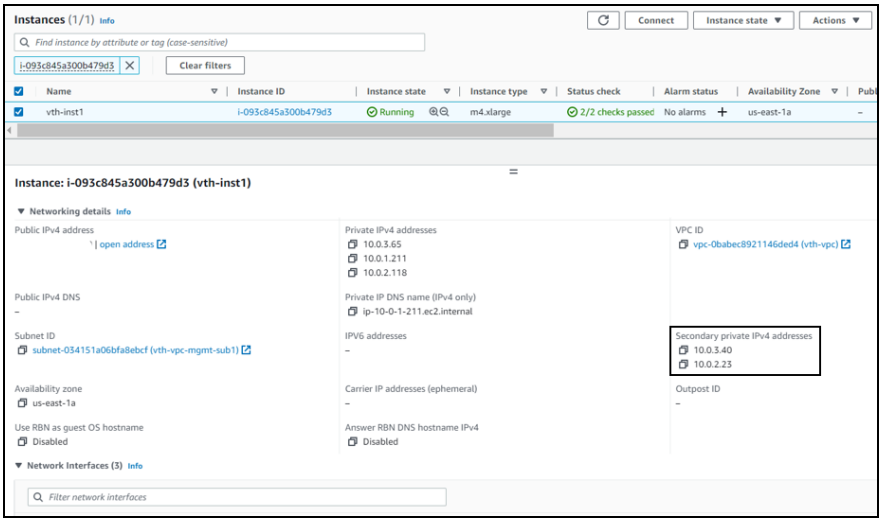
```
vThunder-Active(config)(NOLICENSE)#
```

Verify Traffic Flow

To verify the traffic flow from client machine to server machine via vThunder, perform the following:

1. From **AWS Management Console**, navigate to **EC2** > **Instances**.
2. Select the active vThunder instance name and then click the **Networking** tab.
3. Copy the IP address of the vThunder Secondary IPv4 Address under the **Private IPv4 address**.

   Figure 33 : vThunder instance 1



4. Select your client instance from the **Instances** list.
   Here, **`vth-client`** is the client instance name.
5. Click **Connect**.
   A **Connect to instance** window with **EC2 Instance Connect** tab is displayed.
6. Click **Connect**.
   A **Terminal** window is displayed.

7. Run the following command in the Terminal window to send the traffic from the client machine:

```
curl <vThunder_instance_secondary_private_IPv4_Address>
```

Example

```
curl 10.0.2.23
```

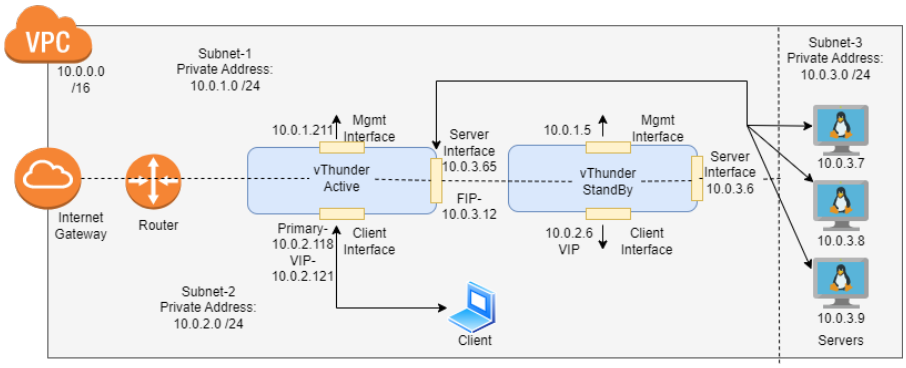
8. Verify if a response is received.

Deploy CFT A10-vThunder_ADC-3NIC-2VM-HA-GLM-PVTVIP

Using this template, two vThunder instances can be deployed containing:

- One management interface and two data interfaces each
- HA support
- GLM integration

Figure 34 : 3NIC-2VM-HA-GLM-PVTVIP Deployment Topology



The following topics are covered:

System Requirements

The CFT displays the default values when you download and save the files on your local machine. You can modify the default values based on your deployment.

You need to configure the following resources to deploy vThunder on the AWS cloud:

Table 9 : System Requirements

Resource Name	Description	Default Value
Stack	A new stack is created with the specified name and location.	Here, the stack name used is **vth**.
Network Interface Card [NIC]	Two types of interfaces are created for each vThunder instance: • One management interface • Two data interfaces	**vth-inst1-mgmt-nic1** **vth-inst1-data-nic2** **vth-inst1-data-nic3** **vth-inst2-mgmt-nic1** **vth-inst2-data-nic2** **vth-inst2-data-nic3**
Subnet	Three subnets are created with an address prefix each.	**vth-vpc-mgmt-sub1** **vth-vpc-data-sub1** **vth-vpc-data-sub2**
Virtual Private Network [VCN]	A virtual private network is assigned to the virtual machine instance.	**vth-vpc** Address prefix for virtual network: 10.0.0.0/16
Elastic Public IP	Two Elastic Public IP are created and attached to the management interface of the vThunder instances.	**vth-inst1-mgmt-nic1-ip** **vth-inst2-mgmt-nic1-ip**
Security Group	One security group is created for the management interface.	Here, the tag names are: **vth-sg-mgmt**

Resource Name	Description	Default Value
	One security group is created for the data interface. **Logical name**: • vThunderSecurityGroupMgmt • vThunderSecurityGroupData	`vth-sg-data`
vThunder Instance	Two vThunder EC2 instances are created: • One active • One standby **Default type**: m4.xlarge (40 Gb memory) Table 10 lists the supported instance types.	`vth-inst1` `vth-inst2`
Server Instance	One Ubuntu Server instance is created. **Default Size**: t2.micro	`vth-server`

Supported Instance Types

Table 10 : List of Supported Instance Types

Instance	vCPU	Memory	Number of Network Interfaces
c4.xlarge	4	7680	4
c4.4xlarge	16	30720	8
c4.8xlarge	36	61440	8
d2.xlarge	4	31232	4
d2.2xlarge	8	62464	4
d2.4xlarge	16	124928	8
d2.8xlarge	36	249856	8
m4.xlarge	4	16384	4

Instance	vCPU	Memory	Number of Network Interfaces
m4.2xlarge	8	32768	4
m4.4xlarge	16	65536	8
m4.10xlarge	40	163840	8
i2.xlarge	4	31232	4
i2.2xlarge	8	62464	4
i2.4xlarge	16	124928	8
i2.8xlarge	32	249856	8
c5d.large	2	4096	3
c5d.9xlarge	36	73728	8
c5d.2xlarge	8	32768	4
c5d.4xlarge	16	73728	8
c5.xlarge	4	8192	4
c5.2xlarge	8	16384	4
c5.4xlarge	16	32768	8
c5.9xlarge	36	73728	8
g3.4xlarge	16	124928	8
g3.8xlarge	32	249856	8
i3.large	2	15616	3
i3.xlarge	4	31232	4
i3.2xlarge	8	62464	4
i3.4xlarge	16	124928	8
i3.8xlarge	32	249856	8
m5d.large	2	8192	3
m5d.xlarge	4	16384	4
m5d.2xlarge	8	32768	4
m5d.4xlarge	16	65536	8
m5.large	2	8192	3
m5.xlarge	4	16384	4

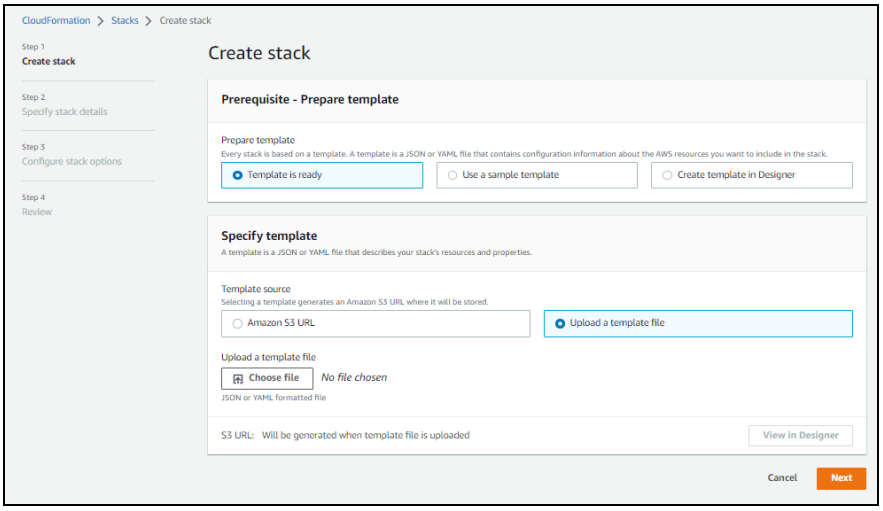
Instance	vCPU	Memory	Number of Network Interfaces
m5.2xlarge	8	32768	4
m5.4xlarge	16	65536	8
r5d.large	2	16384	3
r5d.xlarge	4	32768	4
r5d.2xlarge	8	65536	4
r5d.4xlarge	16	131072	8
r5.large	2	16384	3
r5.xlarge	4	32768	4
r5.2xlarge	8	65536	4
r5.4xlarge	16	131072	8
r4.large	2	15616	3
r4.xlarge	4	31232	4
r4.2xlarge	8	62464	4
r4.4xlarge	16	124928	8
r4.8xlarge	32	249856	8
t3.medium	2	4096	3
t3.large	2	8192	3
t3.xlarge	4	16384	4
t3.2xlarge	8	32768	4
z1d.large	2	16384	3
z1d.xlarge	4	32768	4
z1d.2xlarge	8	65536	4
z1d.3xlarge	12	98304	8
z1d.6xlarge	24	196608	8

Create vThunder Instances

To create vThunder instances, perform the following steps:

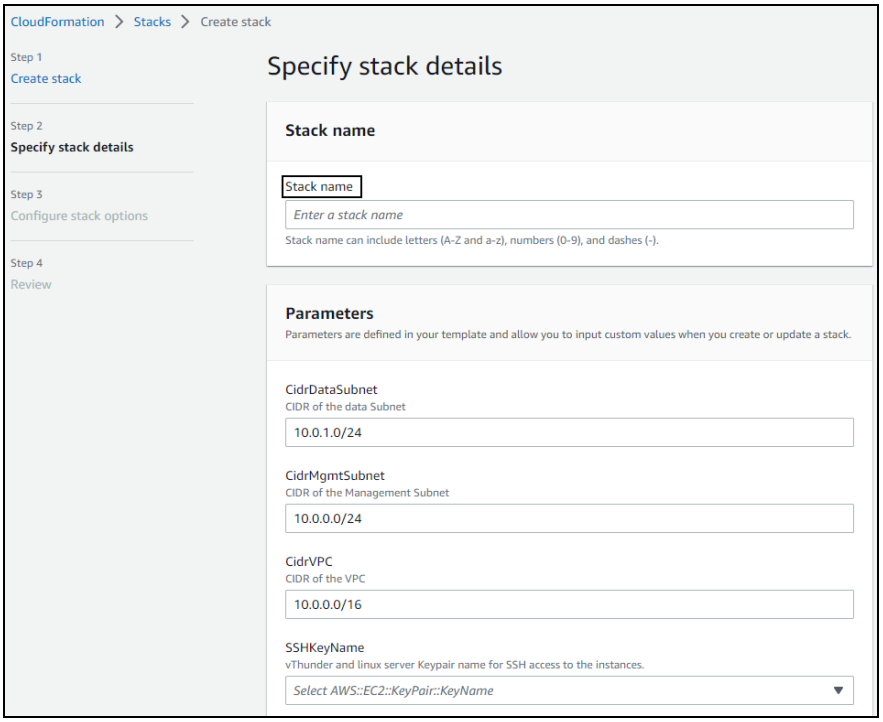
1. From **AWS Management Console**, navigate to **CloudFormation** > **Stacks** > **Create Stack** > **With new resources (standard)**.
   The **Create stack** window is displayed.

   Figure 35 : Create stack window



2. In the **Prerequisite - Prepare template** section, select **Template is ready** option.
   After the selecting this option, the **Specify template** section is displayed.
3. In the **Specify template** section, select **Upload a template file** and click **Choose file** to browse and upload the following template file from the downloaded CFT folder:
   CFT_TMPL_3NIC_2VM_HA_GLM_PVTVIP_1.json
   The selected template file name is displayed as the chosen file.
4. Click **Next**.
   The **Specify stack details** window is displayed.

Figure 36 : Specify stack details window



5. In the **Specify stack details** window, enter or select the following:
   a. In the **Stack name** section, enter a **Stack name**.

      Here, `vth` is the stack name.

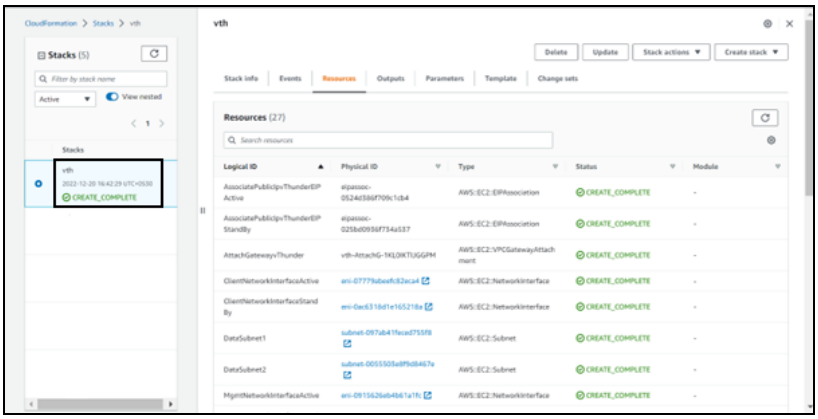
   b. In the **Parameter**s section, enter or select the following:
      - **KeyPairName**: *your SSH key*
      - **TagValue**: `a10-vthunder-adc`
      - **Zone:** *your availability zone*
   c. Verify the other fields and change the values as required. (Optional)
6. Click **Next**.
   The **Configure stack options** window is displayed.
7. Verify the fields and change the values as required. (Optional)
8. Click **Next**.
   The **Review** window is displayed.

9. Verify if all the stack configurations are correct and then click **Submit**.

NOTE: The system may take a few minutes to create the resources. So, wait until the stack status is **CREATE_COMPLETE**.

10. Verify if all the above resources are created in the **AWS Management Console** > **CloudFormation** > **Stacks** > <stack_name> > **Resources** tab.

Figure 37 : Resource listing in the resource group



The two vThunder instances are listed under **Resources** tab.

Access vThunder using GUI or CLI

vThunder instances can be accessed using any of the following ways:

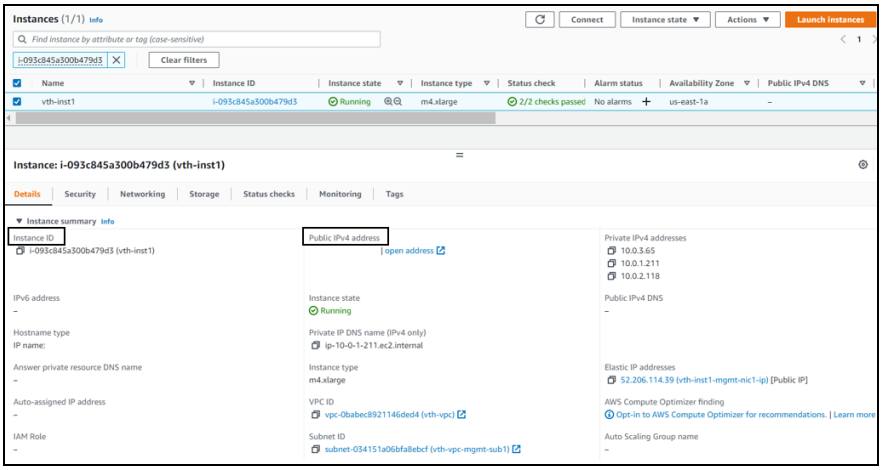
- Access vThunder using GUI
- Access vThunder using CLI

Access vThunder using GUI

To access vThunder instances using GUI, perform the following steps:

1. From **AWS Management Console**, navigate to **EC2** > **Instances**.
2. Select your instance name.
   Here, **`vth-inst1`**, **`vth-inst2`** are the vThunder instances.

Figure 38 : vThunder instance

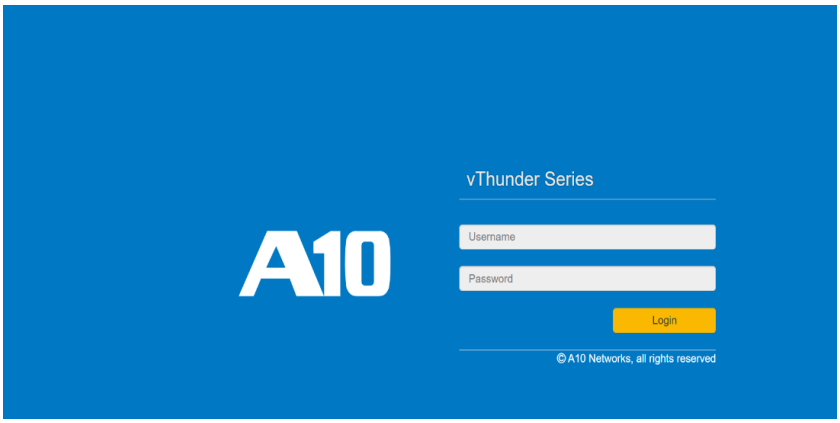


3. For each vThunder instance, perform the following steps:

   a. Copy the **Public IPv4 address** from the **Details** tab and replace the IP address in the below link:
   http://<vThunder_public_IPv4_address>

   b. Open the updated link in any browser.
   The vThunder login window is displayed.

   Figure 39 : vThunder GUI



   c. Enter the following credentials:

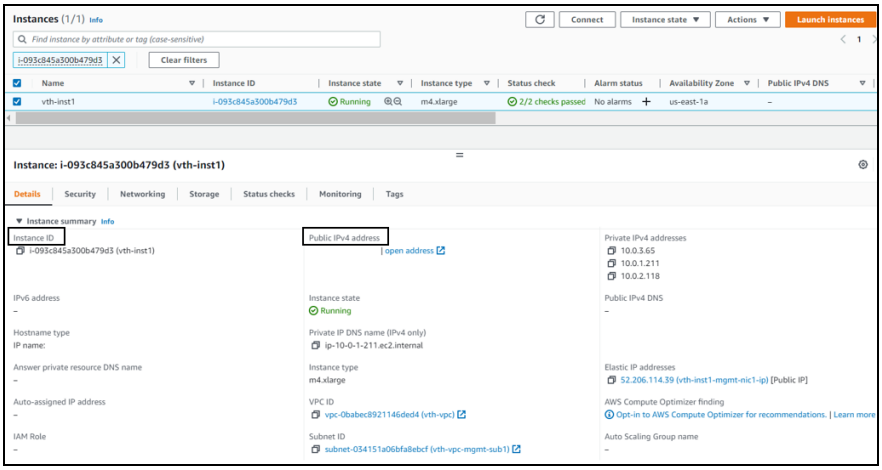
   - Username – admin
   - Password – *EC2 Instance ID*
     The home page is displayed if the entered credentials are correct.

Access vThunder using CLI

To access vThunder instances using CLI, perform the following steps:

1. From **AWS Management Console**, navigate to **EC2** > **Instances**.
2. Select your instance name.
   Here, **`vth-inst1, vth-inst2`** are the vThunder instances.

   Figure 40 : vThunder instance



3. For each vThunder instance, perform the following steps:
   a. Copy the **Public IPv4 address** from the **Details** tab.
   b. Open any SSH client and provide the following to establish a connection:
      - Hostname: Public IPv4 address
      - Username: admin
      - Key: SSH Key
   c. Connect to the session.
   d. In the SSH client session, run the following commands:

```
vThunder(NOLICENSE)>enable <---Execute command--->
Password:                  <---just press Enter key--->
vThunder(NOLICENSE)#config <---Configuration mode--->
vThunder(config)(NOLICENSE)#
```

   The vThunder instances are ready to use.

Configure Server and Client Machine

To test the traffic flow via vThunder, create and configure a server machine and a client machine:

Configure Server Machine

To configure a server machine, perform the following steps:

1. From **AWS Management Console**, navigate to **EC2** > **Instances**.
2. Select your server instance name.
   Here, **`vth-server`** is the server instance name.
3. Click **Connect**.
   A **Connect to instance** window with **EC2 Instance Connect** tab is displayed.
4. Click **Connect**.
   A **Terminal** window is displayed.
5. Run the following command in the Terminal window to create an Apache Server virtual machine:
   ```
   sudo apt install apache2
   ```
   While the Apache server is getting installed, you get a prompt to continue further. Enter 'Y' to continue. After the installation is complete, a newline prompt is displayed.

Configure a Client Machine

To configure a client machine, perform the following steps:

1. From **AWS Management Console**, navigate to **EC2** > **Instances**.
2. Click **Launch Instances**.
   A **Launch an instance** window is displayed.
3. In the **Name and tags** section, enter an instance name.
   Here, enter **`vth-client`** as the client instance name.

4. In the **Application and OS Images** section, select **Ubuntu**.
5. In the **Instance type** section, select the required instance type.
6. In the **Key pair (login)** field, select your SSH key.
7. In the **Network settings** section, click **Edit** to edit the following:
   - VPC: *your VPC*
     Here, `vth-vpc` is the VPC.
   - Subnet: Data subnet 1
     Here, `10.0.2.0/24` is the data subnet value.
   - Auto-assign public IP: Enable
   - Firewall (security groups): Select existing security group
   - Common security groups: *your data security group*
     Here, `vth-vThunderSecurityGroupData` is the security group.
8. Click **Launch instance**.

NOTE: The system may take a few minutes to launch the instance.

The client instance is displayed in the **Instances** list with the status as **Running**.

Import AWS Access Keys

In high availability configuration, IP switching occurs between the two vThunder instances. To enable the IP switching, the AWS keys should be imported on these vThunder instances.

For importing AWS keys on a vThunder instance, perform the following steps:

- Add 'All TCP' rule in Security Group
- Create AWS access keys file
- Create a new FTP server
- Upload access keys on FTP server
- Upload access keys on vThunder instances

Add 'All TCP' rule in Security Group

To add 'All TCP' rule in the security group, perform the following steps:

1. From **AWS Management Console**, navigate to **EC2** > **Security Groups**.
2. Select your security group that was created for the data interface.
   Here, **`vth-sg-data`** (**`vThunderSecurityGroupData`**) is the security group created for the data interface.
   The **Security Groups** window for data interface is displayed.
3. Select the **Inbound rules** tab.
4. Click **Edit inbound rules**.
   The Edit inbound rules window is displayed.
5. Click **Add rule**.
   A rule row is added.
6. Select **All TCP** in the **Type** field.
7. Specify **0.0.0.0/0** in the CIDR notation field.
8. Click **Save rules**.
   The rule is added in the security group.

NOTE: As the 'All TCP' rule is only used to transfer the AWS access keys to a vThunder instance, you can remove it from the security group after the AWS access keys are uploaded on both vThunder instances.

Create AWS access keys file

To create AWS access keys file, perform the following steps:

1. Open a text document on your local machine and copy the following information to this document:

```
[default]
aws_access_key_id = your_aws_access_key_id
aws_secret_access_key = your_aws_secret_access_key
```

2. Update the access key ID and secret access key as per your AWS account.
3. Save the text file with a name.
   Here, **aws_access_key.txt** is the access keys file name.

Create a new FTP server

If you don't have an existing FTP server, create a new FTP server.

To create FTP server, perform the following steps:

1. From **AWS Management Console**, navigate to **EC2** > **Instances**.
2. Click **Launch Instances**.
   A **Launch an instance** window is displayed.
3. In the **Name and tags** section, enter an instance name.
   Here, you can enter **`vth-stack-keys`** as the FTP server name.
4. In the **Application and OS Images** section, select **Ubuntu**.
5. In the **Instance type** section, select the required instance type.
6. In the **Key pair (login)** field, select your SSH key.
7. In the **Network settings** section, click **Edit** to edit the following:
   - VPC: *your VPC*
     Here, **`vth-vpc`** is the VPC.
   - Subnet: Data subnet 1
     Here, **`10.0.2.0/24`** is the data subnet 1 value.
   - Auto-assign public IP: Enable
   - Firewall (security groups): Select existing security group
   - Common security groups: *your data security group*
     Here, **`vth-vThunderSecurityGroupData`** is the security group.
8. Click **Launch instance**.

   NOTE: It may take the system a few minutes to launch the instance.

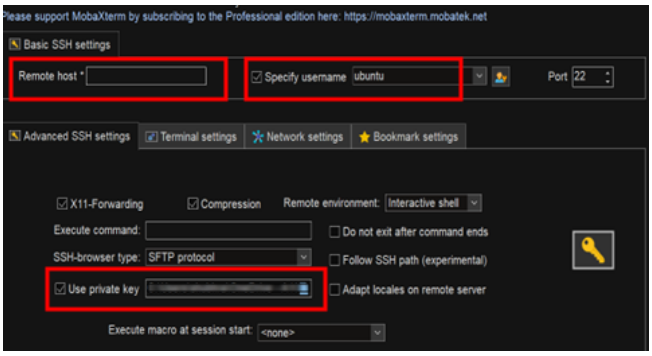
   The FTP server instance is displayed in the **Instances** list with status as **Running**.
9. Select your FTP server instance name.
   Here, **`vth-stack-keys`** is the FTP server instance.
10. Copy the **Public IPv4 address** from the **Details** tab.
11. Open any SSH client and enter the login details.
    For example: If you are using MobaXterm, perform the following steps:

a. Enter the public IPv4 address in the **Remote host** field.

b. Enter **ubuntu** in the **Specify username** field.

c. Select SSH key in the **Use private key** field.

Figure 41 : MobaXterm Basic SSH settings



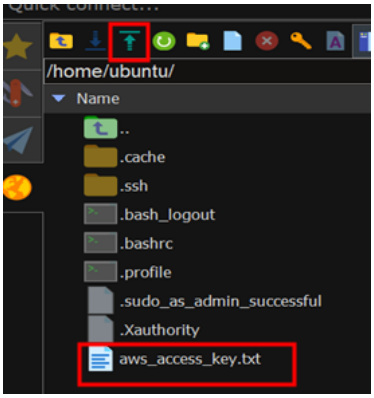
12. Connect to the FTP server instance.
If the connection is established, the FTP server instance is created.

Upload access keys on FTP server

To upload access keys on the FTP server, perform the following steps:

1. Upload the access keys file to the '/home/ubuntu' location of the FTP server. Here, **aws_access_key.txt** is the access key file.

Figure 42 : MobaXterm Upload



2. Verify if the file had been uploaded successfully.
3. Run the following command in the Terminal window to provide the uploaded file

with read and write permission:

```
sudo chmod 777 <access_key_filename>
```

Example:

```
sudo chmod 777 aws_access_key.txt
```

4. Run the following commands to update all the package information:

```
sudo apt update && sudo apt upgrade
```

NOTE: If a message "Daemons using outdated libraries Which services should be restarted?" is prompted, click **Ok** and press **Enter**.

5. Run the following command to install FTP server:

```
sudo apt-get install vsftpd
```

6. Run the following commands to start and enable the FTP server:

```
sudo systemctl start vsftpd
sudo systemctl enable vsftpd
```

7. Run the following commands to add a new user and a password:

```
sudo useradd -m <username>
sudo passwd <username>
New password: <----Enter password---->
Retype new password: <----Re-enter password---->
passwd: password updated successfully
```

Example:

```
sudo useradd -m vth-user
sudo passwd vth-user
New password: <----Enter password---->
Retype new password: <----Re-enter password---->
passwd: password updated successfully
```

8. Navigate to the ' /etc/' folder and run the following command to open the **`vsftpd.conf`** file:

```
sudo vi vsftpd.conf
```

9. Press **Esc** and enter **i** to enable edit/insert mode for the **`vsftpd.conf`** file.
10. Provide write permission to the uploaded file:

```
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
```

11. After the changes, press **Esc** then type **:wq** to save the changes and exit.
12. Run the following commands to restart the FTP server instance:

```
sudo systemctl restart vsftpd
cd /home/ubuntu
ftp localhost
Connected to localhost.
220 (vsFTP 3.0.5)
Name (localhost:ubuntu): <username>
331 Please specify the password.
Password: <----Enter password---->
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Example:

```
sudo systemctl restart vsftpd
cd /home/ubuntu
ftp localhost vth-user vth-user
Connected to localhost.
220 (vsFTP 3.0.5)
Name (localhost:ubuntu): vth-user
331 Please specify the password.
Password: <----Enter password---->
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

13. Put the access keys in the FTP server instance:

```
ftp> put <access_key_filename>
```

Example:

```
ftp> put aws_access_key.txt
```

The AWS access keys are uploaded on the FTP server instance.

Upload access keys on vThunder instances

To upload access keys on both vThunder instances, perform the following steps:

1. Run the following commands on each vThunder instance using CLI to import the access keys:

```
vThunder(config)#admin admin
vThunder(config-admin:admin)#aws-accesskey import use-mgmt-port
ftp://username@publicipoftheinstancewithkeys
               <---Update with the username and the Public IP of
FTP server instance---->
Password []?    <---Provide the password set for FTP server instance
user---->
File name [/]?aws_access_key.txt <---Provide the access keys file name-
--->
vThunder(config-admin:admin)#
```

2. Run the following commands on each vThunder instance to verify the access keys upload:

```
vThunder(config-admin:admin)#aws-accesskey show
[default]
aws_access_key_id = your_aws_access_key_id
aws_secret_access_key = your_aws_secret_access_key
vThunder(config-admin:admin)#exit
vThunder(config)#
vThunder(config)#
vThunder-Active(config)#
```

NOTE: Once the AWS access keys are transferred to vThunder, make sure to delete the FTP server.

Configure vThunder as an SLB with HA

The following topics are covered:

Initial Setup

Before deploying vThunder on AWS cloud as an SLB with HA, you need to configure the corresponding parameters in the CFT.

To configure the parameters, perform the following steps:

1. Navigate to the folder where you have downloaded the CFT, and open the CFT_TMPL_3NIC_2VM_HA_GLM_PVTVIP_CONFIG_SLB_SSL_HA_GLM_PARAM.json with a text editor.

   NOTE: Each parameter has a default value mentioned in the parameter file.

2. Configure the stack name created using CFT.

```
"stackDetails": {
  "value": [
    {
      "stackName": "vth"
    }
  ]},
```

3. Configure SLB server ports.

```
"server-list": {
  "value": [
    {
      "metadata": {
        "description": "SLB server host/fqdn-name."
      },
      "port-list": [
        {
          "port-number": 53,
```

```
            "protocol": "udp"
          },
          {
            "port-number": 80,
            "protocol": "tcp"
          },
          {
            "port-number": 443,
            "protocol": "tcp"
          }
        ]
      }
    ]
  },
```

4. Configure Service Group ports.
   The Service group name by default is “sg+port_number”. If you may want to change the service group name then after changing the name, change the names in the Virtual servers as well.

```
  "serviceGroupList": {
    "value": [
      {
        "name": "sg443",
        "protocol": "tcp",
         "member-list": [
          {
            "port": 443
          }
        ]
      },
      {
        "name": "sg53",
        "protocol": "udp",
        "member-list": [
          {
            "port": 53
          }
        ]
```

```
        },
        {
          "name": "sg80",
          "protocol": "tcp",
          "member-list": [
            {
              "port": 80
            }
          ]
        }
      ]
    },
```

5. Configure a Virtual Server and its ports.
   The virtual server default name is “vip”. It is the Private IP address of Ethernet1.

```
     "virtualServerList": {
        "virtual-server-name": "vip",
        "metadata": {
          "description": "virtual server is using ethernet 1 ip
address"
        },
        "value": [
          {
            "port-number": 53,
            "protocol": "udp",
            "auto": 1,
            "service-group": "sg53"
          },
          {
            "port-number": 80,
            "protocol": "http",
            "auto": 1,
            "service-group": "sg80"
          },
          {
            "port-number": 443,
            "protocol": "https",
            "auto": 1,
```

```
            "service-group": "sg443"
          }
        ]
      },
```

6. Configure DNS.

```
        "dns": {
            "value": "8.8.8.8"
        },
```

7. Configure a Network Gateway IP.
   The default value of network gateway IP address is 10.0.1.1 as this is the first IP address of the data subnet 1 configuration.

```
        "rib-list": [
            {
            "ip-dest-addr":"0.0.0.0",
            "ip-mask":"/0",
            "ip-nexthop-ipv4": [
                {
              "ip-next-hop": "10.0.1.1"
            },
            {
              "ip-next-hop": "10.0.2.1"
            }
            ]
            }
        ],
```

8. Set VRRP-A.

```
        "vrrp-a": {
            "set-id":1
        },
```

9. Set a Terminal Idle Timeout.

```
        "terminal": {
            "idle-timeout":0
        },
```

10. Set VRID.
    For vThunder instance 1, the default priority is 100, and for vThunder instance 2,

it is 99 (100-1).

```
        "vrid-list": [
       {
         "vrid-val": 0,
         "blade-parameters": {
           "priority": 100
         }
       }
     ],
```

11. Configure SSL.

```
     "sslConfig": {
         "requestTimeOut": 40,
         "Path": "server.pem",
         "File": "server",
         "CertificationType": "pem"
       }
```

NOTE: By default, SSL configuration is disabled i.e., no SSL configuration is applied.

The server.pem file is available in the downloaded CFT folder. You can edit this file as required or use a different certificate file, but the path should be changed accordingly.

12. Configure GLM account details.
You can get the **Entitlement Token** from GLM > **Licenses** > select your license > **Overview** > **Info** tab.

```
   "user_name": {
       "value": "xxxxxxxx@a10networks.com"
     },
     "user_password": {
       "value": "xxxxxxxx"
     },
     "entitlement_token": {
       "value": "xxxxxxxx"
     }
 }
```

13. Verify if all the configurations in the CFT_TMPL_3NIC_2VM_HA_GLM_PVTVIP_CONFIG_SLB_SSL_HA_GLM_PARAM.json file are correct and then save the changes.

Deploy vThunder as an SLB

To deploy vThunder on AWS cloud as an SLB, perform the following steps:

1. From your command prompt, navigate to the folder where you have downloaded the CFT.
2. Run the following command to create vThunder SLB instances:

```
python ./CFT_TMPL_3NIC_2VM_HA_GLM_PVTVIP_CONFIG_SLB_SSL_HA_GLM_2.py
```

A message is prompted to upload the SSL certificate and configure HA.

```
Do you want to upload ssl certificate(yes/no)?yes
Do you want to configure HA (yes/no)?yes
Configuring vThunder with instance id i-000d8c13e073a381c
configured ethernet ip
configured ethernet ip
Configured server vth-server
Configure service group
Configured virtual servers
SSL Configured.
Configured primary dns
Configured IP Route
Configured Vrrp A common
Configured idle timeout
Configured vrrp rid
Configured peer group
Configurations are saved on partition: shared
Configuring vThunder with instance id i-0baf10be64819fdc4
configured ethernet ip
configured ethernet ip
Configured server vth-server
Configure service group
Configured virtual servers
SSL Configured.
```

```
Configured primary dns
Configured IP Route
Configured Vrrp A common
Configured idle timeout
Configured vrrp rid
Configured peer group
Configurations are saved on partition: shared
```

If the SSL Certificate upload is successful, a message 'SSL Configured' is displayed.

Verify Deployment

To verify vThunder SLB deployment using CFT, perform the following steps:

1. Run the following command on vThunder instance 1 using CLI:

```
vThunder-Active(config)#show running-config
```

If the deployment is successful, the following SLB configuration is displayed:

```
!Current configuration: 349 bytes
!Configuration last updated at 10:56:58 GMT Fri Jan 6 2023
!Configuration last saved at 10:53:34 GMT Fri Jan 6 2023
!64-bit Advanced Core OS (ACOS) version 5.2.1, build 153 (Dec-11-
2020,16:36)
!
vrrp-a common
  device-id 1
  set-id 1
  enable
!
terminal idle-timeout 0
!
ip dns primary 8.8.8.8
!
!
glm use-mgmt-port
glm enable-requests
glm token vTh205fe920b
```

```
!
interface ethernet 1
  enable
  ip address dhcp
!
interface ethernet 2
  enable
  ip address dhcp
!
vrrp-a vrid 0
  floating-ip 10.0.3.12
  blade-parameters
    priority 99
!
vrrp-a peer-group
  peer 10.0.2.117
  peer 10.0.2.173
!
ip route 0.0.0.0 /0 10.0.1.1
ip route 0.0.0.0 /0 10.0.2.1
!
slb server vth-server 10.0.3.23
  port 53 udp
  port 80 tcp
  port 443 tcp
!
slb service-group sg443 tcp
  member vth-server 443
!
slb service-group sg53 udp
  member vth-server 53
!
slb service-group sg80 tcp
  member vth-server 80
!
slb virtual-server vip 10.0.2.23
  port 53 udp
```

```
    source-nat auto
    service-group sg53
  port 80 http
    source-nat auto
    service-group sg80
  port 443 https
    source-nat auto
    service-group sg443
!
!
end
!Current config commit point for partition 0 is 0 & config mode is
classical-mode
vThunder-Active(config)#
```

2. Run the following command on vThunder instance 1 using CLI:

```
vThunder-Active(config)#show pki cert
```

If the deployment is successful, the following SSL configuration is displayed:

```
Name    Type                        Expiration  Status
-----------------------------------------------------------------
server certificate Jan 28 12:00:00 2028 GMT [Unexpired, Bound]
```

At this point, the vThunder instance 2 has the following prompt:

```
vThunder-Standby(config)#
```

Figure 43 : vThunder instance 1 - Active

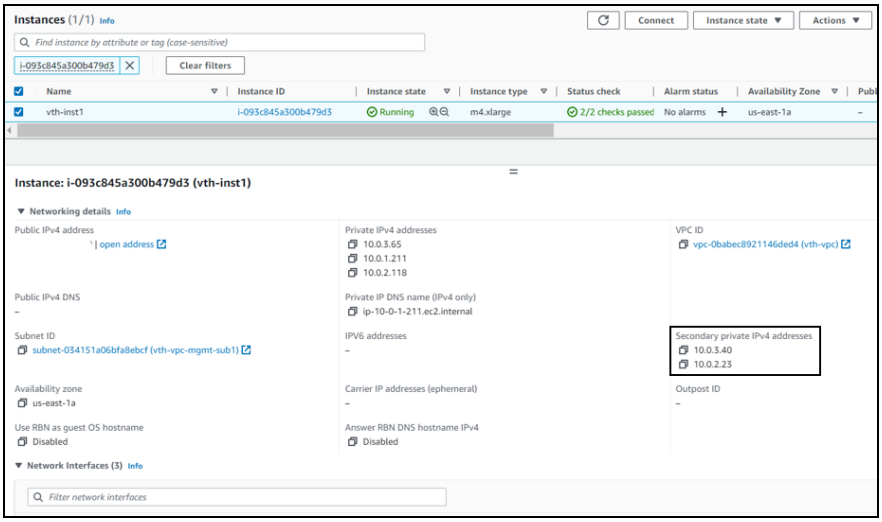
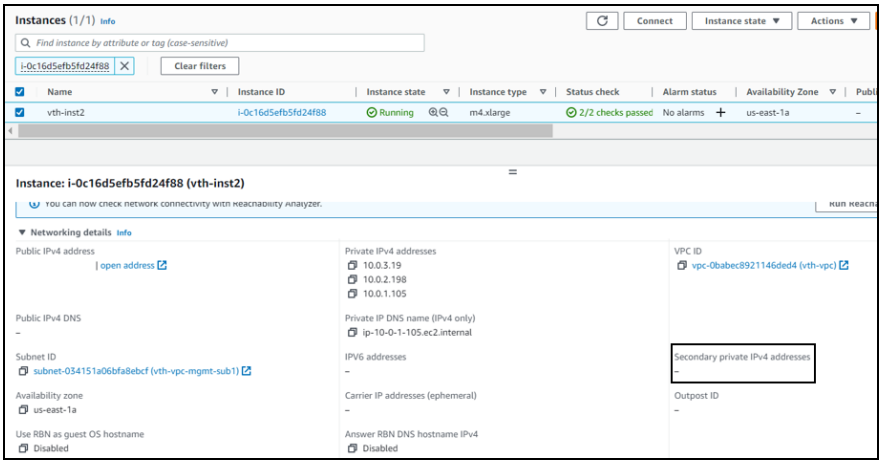


Figure 44 : vThunder instance 2 - Standby



3. Run the following command on vThunder instance 1 using CLI:

```
vThunder-Active(config)#vrrp-a force-self-standby enable
vThunder-Active(config)#
vThunder-ForcedStandby(config)#
```

At this point, IP switching occurs and the vThunder instance 2 prompt becomes:

```
vThunder-Active(config)#
```

4. Run the following command on vThunder instance 2 using CLI:

```
vThunder-Active(config)#show running-config
```

If the deployment is successful, the following SLB configuration is displayed:

```
!Current configuration: 282 bytes
!Configuration last updated at 10:53:35 GMT Fri Jan 6 2023
!Configuration last saved at 10:53:37 GMT Fri Jan 6 2023
!64-bit Advanced Core OS (ACOS) version 5.2.1, build 153 (Dec-11-
2020,16:36)
!
vrrp-a common
  device-id 2
  set-id 1
  enable
!
terminal idle-timeout 0
!
ip dns primary 8.8.8.8
!
!
glm use-mgmt-port
glm enable-requests
glm token vTh205fe920b
!
interface ethernet 1
  enable
  ip address dhcp
!
interface ethernet 2
  enable
  ip address dhcp
!
vrrp-a vrid 0
  floating-ip 10.0.3.12
  blade-parameters
    priority 98
!
vrrp-a peer-group
```

```
  peer 10.0.2.186
  peer 10.0.2.6
!
ip route 0.0.0.0 /0 10.0.1.1
ip route 0.0.0.0 /0 10.0.2.1
!
slb server vth-server 10.0.3.23
  port 53 udp
  port 80 tcp
  port 443 tcp
!
slb service-group sg443 tcp
  member vth-server 443
!
slb service-group sg53 udp
  member vth-server 53
!
slb service-group sg80 tcp
  member vth-server 80
!
slb virtual-server vip 10.0.2.121
  port 53 udp
    source-nat auto
    service-group sg53
  port 80 http
    source-nat auto
    service-group sg80
  port 443 https
    source-nat auto
    service-group sg443
!
sflow setting local-collection
!
sflow collector ip 127.0.0.1 6343
!
!
```

```
end
!Current config commit point for partition 0 is 0 & config mode is
classical-mode
vThunder-Active(config)#
```

5. Run the following command on vThunder instance 2 using CLI:

```
vThunder-Active(config)#show pki cert
```

If the deployment is successful, the following SSL configuration is displayed:

```
Name     Type                              Expiration   Status
-----------------------------------------------------------------
server certificate Jan 28 12:00:00 2028 GMT [Unexpired, Bound]
```

Figure 45 : vThunder instance 1 - Standby

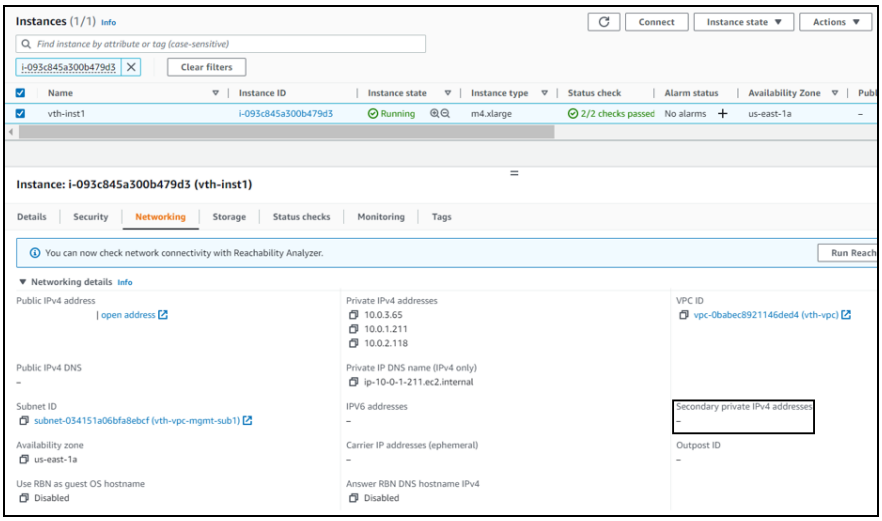
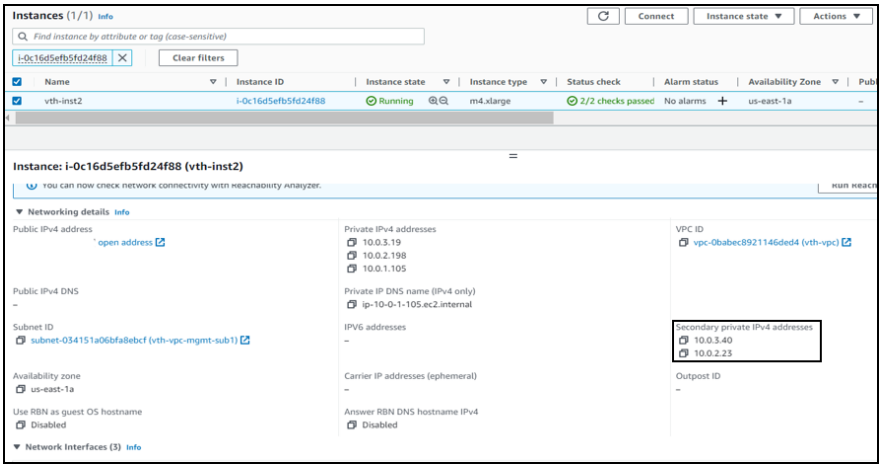


Figure 46 : vThunder instance 2 - Active



6. If you want to make vThunder instance 1 active, run the following command on vThunder instance 2 using CLI:

```
vThunder-Active(config)#vrrp-a force-self-standby disable
vThunder-Active(config)#
vThunder-ForcedStandby(config)#
```

At this point, IP switching occurs and the vThunder instance 1 prompt becomes:

```
vThunder-Active(config)#
```

Verify GLM

The application of license can be verified using any of the following ways:

- Verify License using GUI
- Verify License using CLI

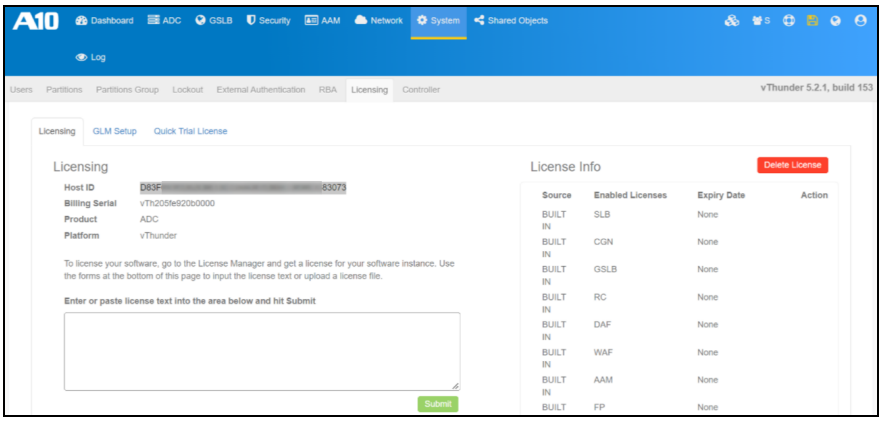
Verify License using GUI

To verify license using GUI, perform the following steps:

1. Log in to your vThunder instance using Public IPv4 address.
   Here, **`vth-inst1`** is the vThunder instance.
2. Navigate to **Profile** > **Setting** > **Licensing**.

3. Click the **Licensing** tab.
   If the license is successfully applied on vThunder, the Host ID is displayed.

   Figure 47 : GLM > Licensing window



Verify License using CLI

To verify license using CLI, perform the following steps:

1. Verify if NOLICENSE is removed from the vThunder prompt.
2. Run the following command on vThunder:

```
vThunder(config)#show license
```

   If the license is successfully applied on vThunder, the Host ID is displayed:

```
Host ID         : D83F****82EB633D****067DB84136****83073
```

3. Run the following command on vThunder instance:

```
vThunder(config)#show license-info
```

   If the license is successfully applied on vThunder, the following GLM configuration is displayed:

```
Host ID         : D83F****82EB633D****067DB84136****83073
USB ID          : Not Available
Billing Serials: vTh205fe920b0000
Token           : Not Available
Product         : ADC
Platform        : vThunder
Burst           : Disabled
```

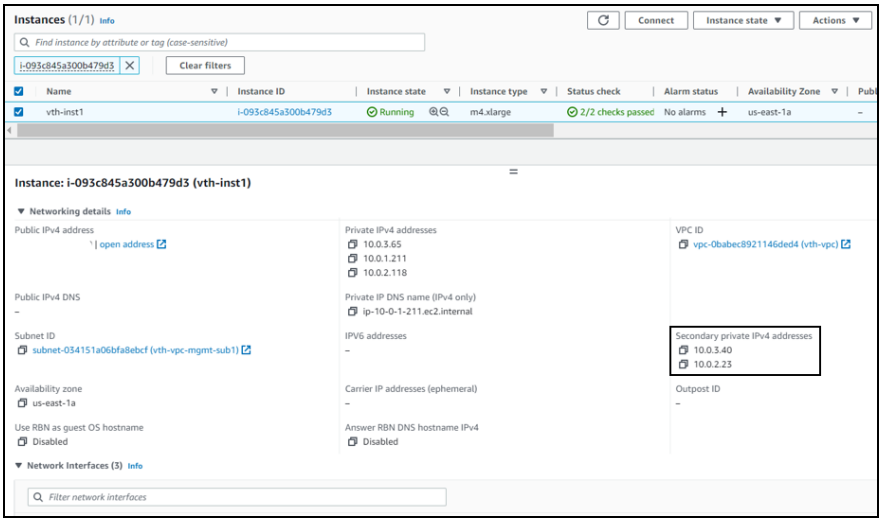
```
GLM Ping Interval In Hours : 24
------------------------------------------------------------------------
-------
Enabled Licenses   Expiry Date (UTC)         Notes
------------------------------------------------------------------------
-------
SLB                None
CGN                None
GSLB               None
RC                 None
DAF                None
WAF                None
AAM                None
FP                 None
WEBROOT            N/A              Requires an additional Webroot
license.
THREATSTOP         N/A              Requires an additional ThreatSTOP
license.
QOSMOS             N/A              Requires an additional QOSMOS
license.
WEBROOT_TI         N/A              Requires an additional Webroot
Threat Intel license.
IPSEC_VPN          N/A              Requires an additional IPsec VPN
license.
500 Mbps Bandwidth 20-January-2023
```

Verify Traffic Flow

To verify the traffic flow from client machine to server machine via vThunder, perform the following:

1. From **AWS Management Console**, navigate to **EC2** > **Instances**.
2. Select the active vThunder instance name and then click the **Networking** tab.
3. Copy the IP address of the vThunder Secondary IPv4 Address under the **Private IPv4 address**.

Figure 48 : vThunder instance 1



4. Select your client instance from the **Instances** list.
   Here, **`vth-client`** is the client instance name.
5. Click **Connect**.
   A **Connect to instance** window with **EC2 Instance Connect** tab is displayed.
6. Click **Connect**.
   A **Terminal** window is displayed.
7. Run the following command in the Terminal window to send the traffic from the client machine:

   ```
   curl <vThunder_instance_secondary_private_IPv4_Address>
   ```

   Example

   ```
   curl 10.0.2.23
   ```

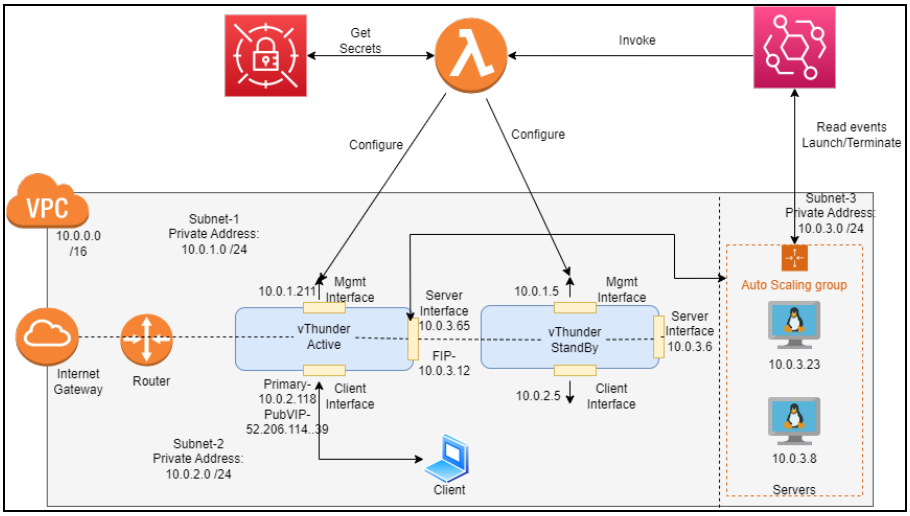
8. Verify if a response is received.

Deploy CFT A10-vThunder_ADC-3NIC-2VM-HA-GLM-PUBVIP-BACKAUTO

Using this template, two vThunder instances can be deployed containing:

- One management interface and two data interfaces each
- HA support
- GLM integration
- Backend server autoscaling support.

Figure 49 : 3NIC-2VM-HA-GLM-PUBVIP-BACKAUTO Deployment Topology



The following topics are covered:

System Requirements

The CFT displays the default values when you download and save the files on your local machine. You can modify the default values based on your deployment.

You need to configure the following resources to deploy vThunder on the AWS cloud:

Table 11 : System Requirements

Resource Name	Description	Default Value
Stack	A new stack is created with the specified name and location.	Here, the stack name used is **vth**.
Network Interface Card [NIC]	Two types of interfaces are created for each vThunder instance: • One management interface • Two data interfaces	**vth-inst1-mgmt-nic1** **vth-inst1-data-nic2** **vth-inst1-data-nic3** **vth-inst2-mgmt-nic1** **vth-inst2-data-nic2** **vth-inst2-data-nic3**
Subnet	Three subnets are created with an address prefix each.	**vth-vpc-mgmt-sub1** **vth-vpc-data-sub1** **vth-vpc-data-sub2**
Virtual Private Network [VCN]	A virtual private network is assigned to the virtual machine instance.	**vth-vpc** Address prefix for virtual network: 10.0.0.0/16

Resource Name	Description	Default Value
Elastic Public IP	Three Elastic Public IP are created and attached to management and data interface of vThunder instances.	**vth-inst1-mgmt-nic1-ip** **vth-inst1-data-nic1-ip** **vth-inst2-mgmt-nic1-ip**
Security Group	Two security groups are created: • One security group for the management interface. • One security group for the data interface. **Logical name**: • vThunderSecurityGroupMgmt • vThunderSecurityGroupData	Here, the tag names are: **vth-sg-mgmt** **vth-sg-data**
vThunder Instance	Two vThunder EC2 instances are created: • One active • One standby **Default type**: m4.xlarge (40 Gb memory) Table 12 lists the supported instance types.	**vth-inst1** **vth-inst2**
Lambda Function	A Lambda function is created, and the code is uploaded from S3 bucket.	**vth-lambda-function**
EventBus	An EventBus is created and event rules are added.	**vth-eventbus**
AutoScaling	An autoscaling group is created to add	**vth-auto-scale-group**

Resource Name	Description	Default Value
Group	or delete instances as needed.	

Supported Instance Types

Table 12 : List of Supported Instance Types

Instance	vCPU	Memory	Number of Network Interfaces
c4.xlarge	4	7680	4
c4.4xlarge	16	30720	8
c4.8xlarge	36	61440	8
d2.xlarge	4	31232	4
d2.2xlarge	8	62464	4
d2.4xlarge	16	124928	8
d2.8xlarge	36	249856	8
m4.xlarge	4	16384	4
m4.2xlarge	8	32768	4
m4.4xlarge	16	65536	8
m4.10xlarge	40	163840	8
i2.xlarge	4	31232	4
i2.2xlarge	8	62464	4
i2.4xlarge	16	124928	8
i2.8xlarge	32	249856	8
c5d.large	2	4096	3
c5d.9xlarge	36	73728	8
c5d.2xlarge	8	32768	4
c5d.4xlarge	16	73728	8
c5.xlarge	4	8192	4

Instance	vCPU	Memory	Number of Network Interfaces
c5.2xlarge	8	16384	4
c5.4xlarge	16	32768	8
c5.9xlarge	36	73728	8
g3.4xlarge	16	124928	8
g3.8xlarge	32	249856	8
i3.large	2	15616	3
i3.xlarge	4	31232	4
i3.2xlarge	8	62464	4
i3.4xlarge	16	124928	8
i3.8xlarge	32	249856	8
m5d.large	2	8192	3
m5d.xlarge	4	16384	4
m5d.2xlarge	8	32768	4
m5d.4xlarge	16	65536	8
m5.large	2	8192	3
m5.xlarge	4	16384	4
m5.2xlarge	8	32768	4
m5.4xlarge	16	65536	8
r5d.large	2	16384	3
r5d.xlarge	4	32768	4
r5d.2xlarge	8	65536	4
r5d.4xlarge	16	131072	8
r5.large	2	16384	3
r5.xlarge	4	32768	4
r5.2xlarge	8	65536	4
r5.4xlarge	16	131072	8
r4.large	2	15616	3
r4.xlarge	4	31232	4

Instance	vCPU	Memory	Number of Network Interfaces
r4.2xlarge	8	62464	4
r4.4xlarge	16	124928	8
r4.8xlarge	32	249856	8
t3.medium	2	4096	3
t3.large	2	8192	3
t3.xlarge	4	16384	4
t3.2xlarge	8	32768	4
z1d.large	2	16384	3
z1d.xlarge	4	32768	4
z1d.2xlarge	8	65536	4
z1d.3xlarge	12	98304	8
z1d.6xlarge	24	196608	8

Create S3 Bucket

The SLB configuration file is uploaded on AWS using S3 bucket.

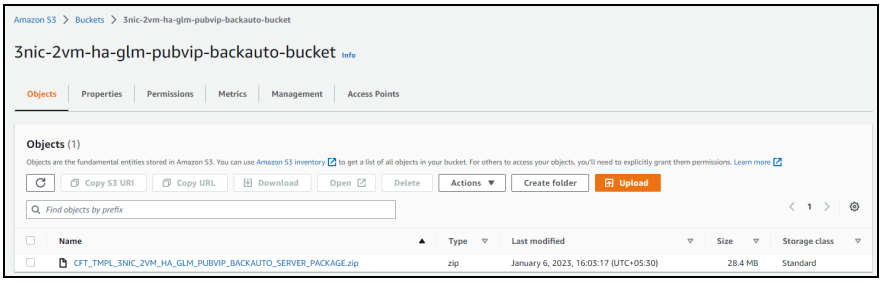
To create an S3 bucket, perform the following steps:

1. From your command prompt, navigate to the folder where you have downloaded the CFT.
2. Run the following command to create an S3 bucket and store the Lambda function python script:

   ```
   python ./CFT_TMPL_3NIC_2VM_HA_GLM_PUBVIP_BACKAUTO_SERVER_PACKAGE_S3_1.py
   ```

   A message 'File uploaded in S3 bucket successfully' is displayed.
3. Navigate to **AWS Management Console** > **Buckets** > <bucket_name> to verify if the 3nic-2vm-ha-glm-pubvip-backauto-bucket is created and CFT_TMPL_3NIC_2VM_HA_GLM_PUBVIP_BACKAUTO_SERVER_PACKAGE.zip is uploaded.

Figure 50 : S3 Bucket

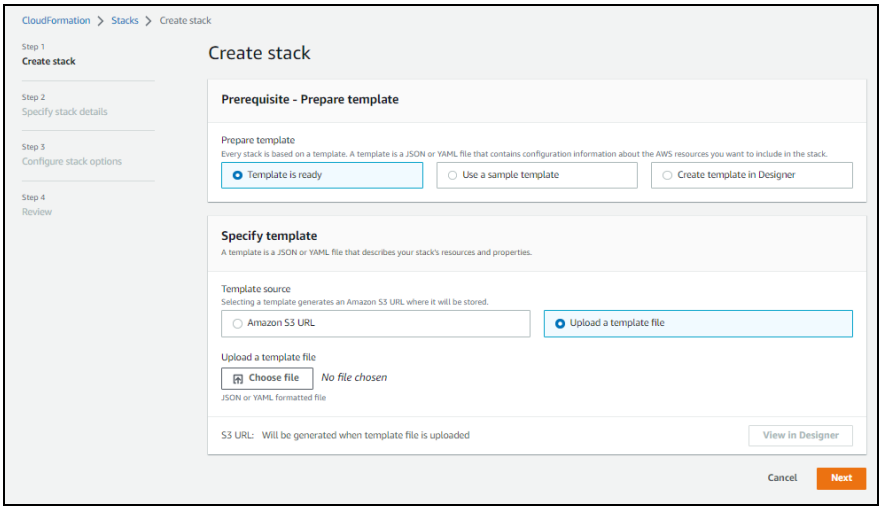


Create vThunder Instances

To create vThunder instances, perform the following steps:

1. From **AWS Management Console**, navigate to **CloudFormation** > **Stacks** > **Create Stack** > **With new resources (standard)**.
   The **Create stack** window is displayed.

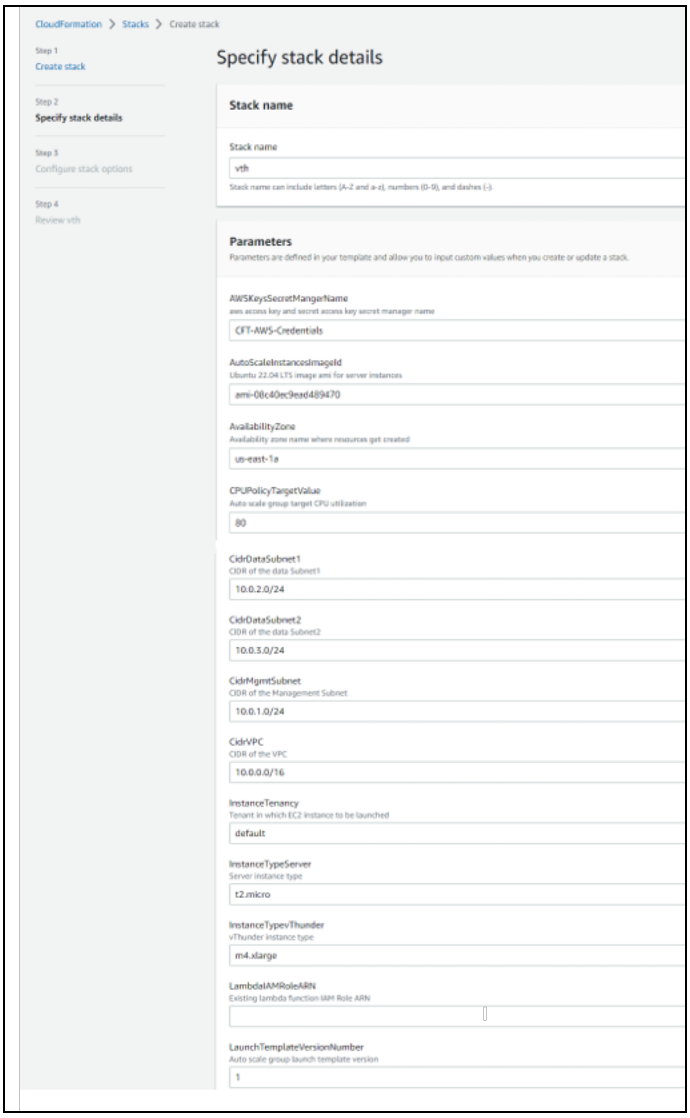
   Figure 51 : Create stack window



2. In the **Prerequisite - Prepare template** section, select **Template is ready** option. After the selecting this option, the **Specify template** section is displayed.
3. In the **Specify template** section, select **Upload a template file** and click **Choose file** to browse and upload the following template file from the downloaded CFT folder:
   CFT_TMPL_3NIC_2VM_HA_GLM_PUBVIP_BACKAUTO_2.json
   The selected template file name is displayed as the chosen file.

4. Click **Next**.
   The **Specify stack details** window is displayed.

   Figure 52 : Specify stack details window



5. In the **Specify stack details** window, enter or select the following:
   a. In the **Stack name** section, enter a **Stack name**.

      Here, **`vth`** is the stack name.

   b. In the **Parameters** section, enter or select the required value in the following fields:

- **AWSKeysSecretManagerName**: *your AWS keys secret manager name*
- **KeyPairName**: *your SSH key*
- **LambdaIAMRoleARN**: *your Lambda function execution role ARN*
- **TagValue**: `a10-vthunder-adc`
- **Zone**: *your availability zone*

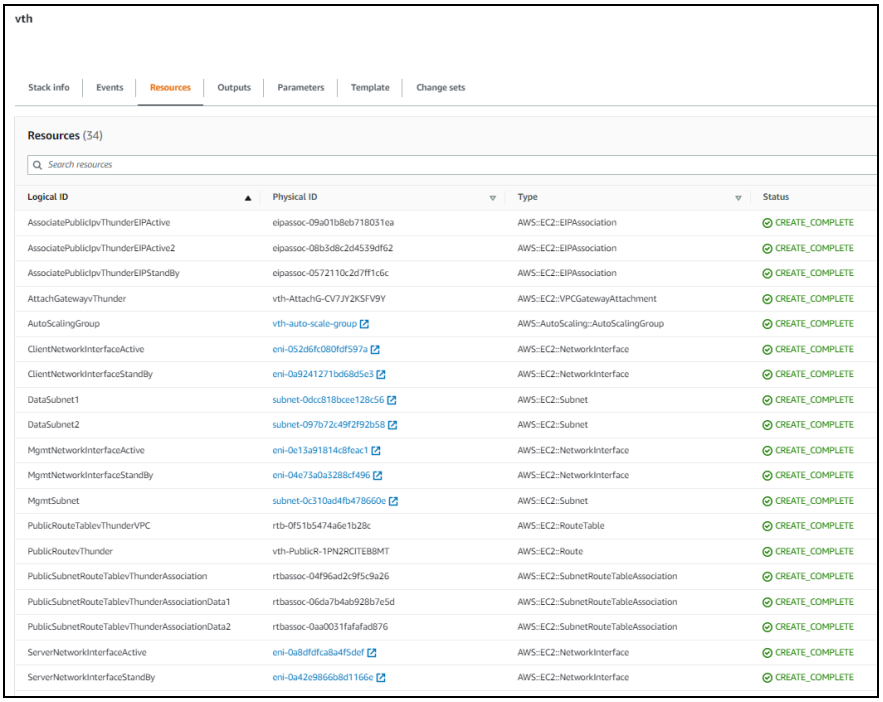
c. Verify the other fields and change the values as required. (Optional)

6. Click **Next**.
   The **Configure stack options** window is displayed.
7. Verify the fields and change the values as required. (Optional)
8. Click **Next**.
   The **Review** window is displayed.
9. Verify if all the stack configurations are correct and then click **Submit**.

NOTE: The system may take a few minutes to create the resources. So, wait until the stack status is **CREATE_COMPLETE**.

10. Verify if all the above resources are created in the **AWS Management Console** > **CloudFormation** > **Stacks** > <stack_name> > **Resources** tab.

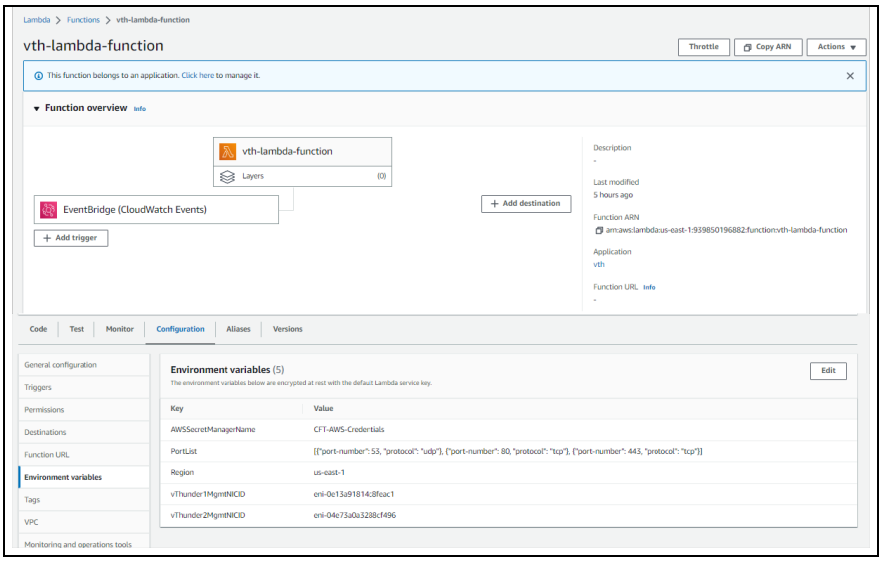
Figure 53 : Resource listing in the resource group



The two vThunder instances are listed under **Resources** tab.

11. Verify if the environment variables of the Lambda function are created in **AWS Management Console** > **Lambda** > **Functions** > <function_name> > **Configuration** > **Environment variables** tab.

Figure 54 : Lambda Function



NOTE: If you delete a stack having lambda function, you need to delete the Lambda function manually as it does not get deleted with the stack.

Access vThunder using GUI or CLI

vThunder instances can be accessed using any of the following ways:

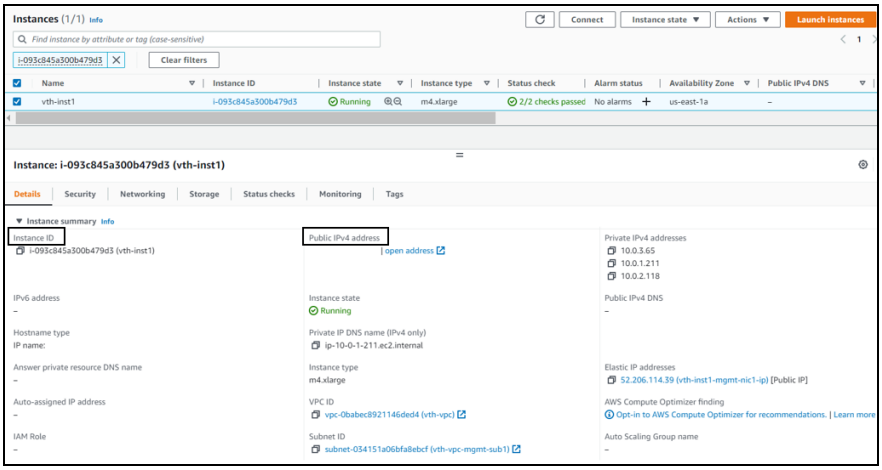
- Access vThunder using GUI
- Access vThunder using CLI

Access vThunder using GUI

To access vThunder instances using GUI, perform the following steps:

1. From **AWS Management Console**, navigate to **EC2** > **Instances**.
2. Select your instance name.
   Here, **`vth-inst1`**, **`vth-inst2`** are the vThunder instances.

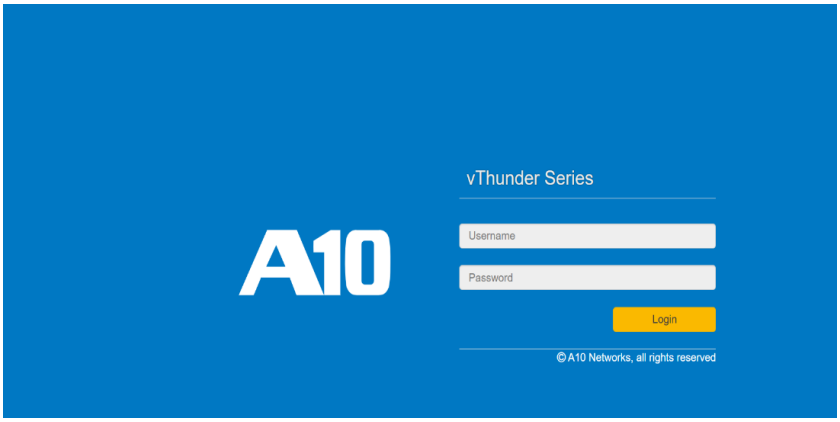
   Figure 55 : vThunder instance



3. For each vThunder instance, perform the following steps:
   a. Copy the **Public IPv4 address** from the **Details** tab and replace the IP address in the below link:
   http://<vThunder_public_IPv4_address>

b. Open the updated link in any browser.
The vThunder login window is displayed.

Figure 56 : vThunder GUI



c. Enter the following credentials:

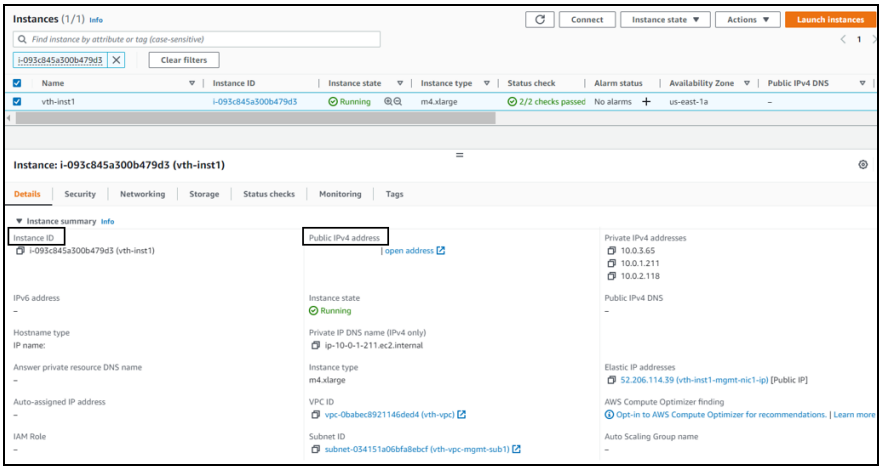
- Username – admin
- Password – *EC2 Instance ID*
  The home page is displayed if the entered credentials are correct.

Access vThunder using CLI

To access vThunder instances using CLI, perform the following steps:

1. From **AWS Management Console**, navigate to **EC2** > **Instances**.
2. Select your instance name.
   Here, **`vth-inst1, vth-inst2`** are the vThunder instances.

Figure 57 : vThunder instance



3. For each vThunder instance, perform the following steps:
   a. Copy the **Public IPv4 address** from the **Details** tab.
   b. Open any SSH client and provide the following to establish a connection:
      - Hostname: Public IPv4 address
      - Username: admin
      - Key: SSH Key
   c. Connect to the session.
   d. In the SSH client session, run the following commands:

```
vThunder(NOLICENSE)>enable <---Execute command--->
Password:                  <---just press Enter key--->
vThunder(NOLICENSE)#config <---Configuration mode--->
vThunder(config)(NOLICENSE)#
```

   The vThunder instances are ready to use.

Configure Server and Client Machine

To test the traffic flow via vThunder, create and configure a server machine and a client machine:

Configure a Server Machine

To configure a server machine, perform the following steps:

1. From **AWS Management Console**, navigate to **EC2** > **Instances**.
2. Select your autoscale server instance name.
   Here, **`autoscaling-vm`** is the server instance name.
3. Click **Connect**.
   A **Connect to instance** window with **EC2 Instance Connect** tab is displayed.
4. Click **Connect**.
   A **Terminal** window is displayed.
5. Run the following command in the Terminal window to create an Apache Server virtual machine:
   ```
   sudo apt install apache2
   ```
   While the Apache server is getting installed, you get a prompt to continue further. Enter 'Y' to continue. After the installation is complete, a newline prompt is displayed.

Configure a Client Machine

To configure a client machine, perform the following steps:

1. From **AWS Management Console**, navigate to **EC2** > **Instances**.
2. Click **Launch Instances**.
   A **Launch an instance** window is displayed.
3. In the **Name and tags** section, enter an instance name.
   Here, enter **`vth-client`** as the client instance name.
4. In the **Application and OS Images** section, select **Ubuntu**.
5. In the **Instance type** section, select the required instance type.
6. In the **Key pair (login)** field, select your SSH key.

7. In the **Network settings** section, click **Edit** to edit the following:
   - VPC: *your VPC*
     Here, `vth-vpc` is the VPC.
   - Subnet: Data subnet 1
     Here, `10.0.2.0/24` is the data subnet value.
   - Auto-assign public IP: Enable
   - Firewall (security groups): Select existing security group
   - Common security groups: *your data security group*
     Here, `vth-vThunderSecurityGroupData` is the security group.
8. Click **Launch instance**.

NOTE: The system may take a few minutes to launch the instance.

The client instance is displayed in the **Instances** list with the status as **Running**.

Import AWS Access Keys

In high availability configuration, IP switching occurs between the two vThunder instances. To enable the IP switching, the AWS keys should be imported on these vThunder instances.

For importing AWS keys on a vThunder instance, perform the following steps:

1. Add 'All TCP' rule in Security Group
2. Create AWS access keys file
3. Create a new FTP server
4. Upload access keys on FTP server
5. Upload access keys on vThunder instances

Add 'All TCP' rule in Security Group

To add 'All TCP' rule in the security group, perform the following steps:

1. From **AWS Management Console**, navigate to **EC2** > **Security Groups**.
2. Select your security group that was created for the data interface.
   Here, **`vth-sg-data`** (**`vThunderSecurityGroupData`**) is the security group created for data interface.
   The **Security Groups** window for data interface is displayed.
3. Select the **Inbound rules** tab.
4. Click **Edit inbound rules**.
   The Edit inbound rules window is displayed.
5. Click **Add rule**.
   A rule row is added.
6. Select **All TCP** in the **Type** field.
7. Specify **0.0.0.0/0** in the CIDR notation field.
8. Click **Save rules**.
   The rule is added in the security group.

NOTE: As the 'All TCP' rule is only used to transfer the AWS access keys to a vThunder instance, you can remove it from the security group after the AWS access keys are uploaded on both vThunder instances.

Create AWS access keys file

To create AWS access keys file, perform the following steps:

1. Open a text document on your local machine and copy the following information to this document:

```
[default]
aws_access_key_id = your_aws_access_key_id
aws_secret_access_key = your_aws_secret_access_key
```

2. Update the access key ID and secret access key as per your AWS account.
3. Save the text file with a name.
   Here, **aws_access_key.txt** is the access keys file name.

Create a new FTP server

If you don't have an existing FTP server, create a new FTP server.

To create FTP server, perform the following steps:

1. From **AWS Management Console**, navigate to **EC2** > **Instances**.
2. Click **Launch Instances**.
   A **Launch an instance** window is displayed.
3. In the **Name and tags** section, enter an instance name.
   Here, you can enter **`vth-stack-keys`** as the FTP server name.
4. In the **Application and OS Images** section, select **Ubuntu**.
5. In the **Instance type** section, select the required instance type.
6. In the **Key pair (login)** field, select your SSH key.
7. In the **Network settings** section, click **Edit** to edit the following:
   - VPC: *your VPC*
     Here, **`vth-vpc`** is the VPC.
   - Subnet: Data subnet 1
     Here, **`10.0.2.0/24`** is the data subnet 1 value.
   - Auto-assign public IP: Enable
   - Firewall (security groups): Select existing security group
   - Common security groups: *your data security group*
     Here, **`vth-vThunderSecurityGroupData`** is the security group.
8. Click **Launch instance**.

   NOTE: It may take the system a few minutes to launch the instance.

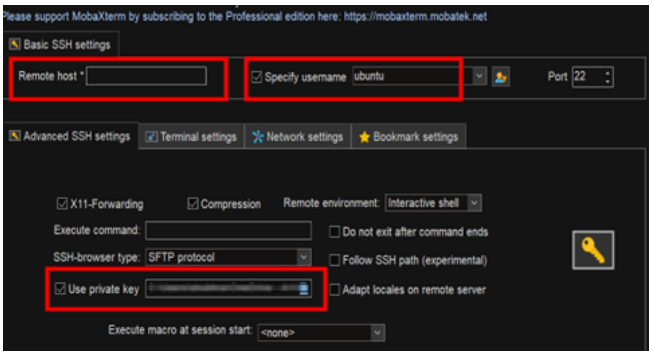
   The FTP server instance is displayed in the **Instances** list with status as **Running**.
9. Select your FTP server instance name.
   Here, **`vth-stack-keys`** is the FTP server instance.
10. Copy the **Public IPv4 address** from the **Details** tab.
11. Open any SSH client and enter the login details.
    For example: If you are using MobaXterm, perform the following steps:

a. Enter the public IPv4 address in the **Remote host** field.

b. Enter **ubuntu** in the **Specify username** field.

c. Select SSH key in the **Use private key** field.

Figure 58 : MobaXterm Basic SSH settings



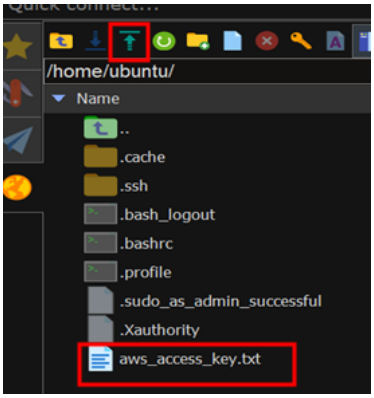
12. Connect to the FTP server instance.
If the connection is established, the FTP server instance is created.

Upload access keys on FTP server

To upload access keys on the FTP server, perform the following steps:

1. Upload the access keys file to the '/home/ubuntu' location of the FTP server. Here, **aws_access_key.txt** is the access key file.

Figure 59 : MobaXterm Upload



2. Verify if the file had been uploaded successfully.
3. Run the following command in the Terminal window to provide the uploaded file

with read and write permission:

```
sudo chmod 777 <access_key_filename>
```

Example:

```
sudo chmod 777 aws_access_key.txt
```

4. Run the following commands to update all the package information:

```
sudo apt update && sudo apt upgrade
```

NOTE: If a message "Daemons using outdated libraries Which services should be restarted?" is prompted, click **Ok** and press **Enter**.

5. Run the following command to install FTP server:

```
sudo apt-get install vsftpd
```

6. Run the following commands to start and enable the FTP server:

```
sudo systemctl start vsftpd
sudo systemctl enable vsftpd
```

7. Run the following commands to add a new user and a password:

```
sudo useradd -m <username>
sudo passwd <username>
New password: <----Enter password---->
Retype new password: <----Re-enter password---->
passwd: password updated successfully
```

Example:

```
sudo useradd -m vth-user
sudo passwd vth-user
New password: <----Enter password---->
Retype new password: <----Re-enter password---->
passwd: password updated successfully
```

8. Navigate to the ' /etc/' folder and run the following command to open the **`vsftpd.conf`** file:

```
sudo vi vsftpd.conf
```

9. Press **Esc** and enter **i** to enable edit/insert mode for the **`vsftpd.conf`** file.
10. Provide write permission to the uploaded file:

```
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
```

11. After the changes, press **Esc** then type **:wq** to save the changes and exit.
12. Run the following commands to restart the FTP server instance:

```
sudo systemctl restart vsftpd
cd /home/ubuntu
ftp localhost
Connected to localhost.
220 (vsFTP 3.0.5)
Name (localhost:ubuntu): <username>
331 Please specify the password.
Password: <----Enter password---->
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Example:

```
sudo systemctl restart vsftpd
cd /home/ubuntu
ftp localhost vth-user vth-user
Connected to localhost.
220 (vsFTP 3.0.5)
Name (localhost:ubuntu): vth-user
331 Please specify the password.
Password: <----Enter password---->
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

13. Put the access keys in the FTP server instance:

```
ftp> put <access_key_filename>
```

Example:

```
ftp> put aws_access_key.txt
```

The AWS access keys are uploaded on the FTP server instance.

Upload access keys on vThunder instances

To upload access keys on both vThunder instances, perform the following steps:

1. Run the following commands on each vThunder instance using CLI to import the access keys:

```
vThunder(config)#admin admin
vThunder(config-admin:admin)#aws-accesskey import use-mgmt-port
ftp://username@publicipoftheinstancewithkeys
               <---Update with the username and the Public IP of
FTP server instance---->
Password []?    <---Provide the password set for FTP server instance
user---->
File name [/]?aws_access_key.txt <---Provide the access keys file name-
--->
vThunder(config-admin:admin)#
```

2. Run the following commands on each vThunder instance to verify the access keys upload:

```
vThunder(config-admin:admin)#aws-accesskey show
[default]
aws_access_key_id = your_aws_access_key_id
aws_secret_access_key = your_aws_secret_access_key
vThunder(config-admin:admin)#exit
vThunder(config)#
vThunder(config)#
vThunder-Active(config)#
```

NOTE: Once the AWS access keys are transferred to vThunder, make sure to delete the FTP server.

Configure vThunder as an SLB with HA

The following topics are covered:

Initial Setup

Before deploying vThunder on AWS cloud as an SLB with HA, you need to configure the corresponding parameters in the CFT.

To configure the parameters, perform the following steps:

1. Navigate to the folder where you have downloaded the CFT, and open the CFT_TMPL_3NIC_2VM_HA_GLM_PUBVIP_BACKAUTO_CONFIG_SSL_SLB_HA_GLM_PARAM.json with a text editor.

   NOTE: Each parameter has a default value mentioned in the parameter file.

2. Configure the stack name created using CFT.

```
    "stackDetails": {
      "value": [
        {
          "stackName": "vth"
        }
      ]},
```

3. Set the capacity of the autoscale server.

```
    "desiredCapacity": 1,
```

4. Configure Service Group.
   The Service group name by default is “sg+port_number”. If you may want to change the service group name then after changing the name, change the names in the Virtual servers as well.

```
      "serviceGroup": {
        "value": [
          {
```

```
            "name": "sg443",
            "protocol": "tcp"
          },
          {
            "name": "sg53",
            "protocol": "udp"
          },
          {
            "name": "sg80",
            "protocol": "tcp"
          }
        ]
      },
```

5. Configure a Virtual Server and its ports.
   The virtual server default name is “vip”. It is the Private IP address of Ethernet1.

```
      "virtualServerList": {
        "virtual-server-name": "vip",
        "metadata": {
          "description": "virtual server is using ethernet 1 ip
address"
        },
        "value": [
          {
            "port-number": 53,
            "protocol": "udp",
            "auto": 1,
            "service-group": "sg53"
          },
          {
            "port-number": 80,
            "protocol": "http",
            "auto": 1,
            "service-group": "sg80"
          },
          {
            "port-number": 443,
            "protocol": "https",
```

```
            "auto": 1,
            "service-group": "sg443"
          }
        ]
      },
```

6. Configure DNS.

```
      "dns": {
          "value": "8.8.8.8"
      },
```

7. Configure a Network Gateway IP.
   The default value of network gateway IP address for data subnet is 10.0.2.1 and for management subnet, it is 10.0.1.1.

```
    "rib-list": [
      {
        "ip-dest-addr": "0.0.0.0",
        "ip-mask": "/0",
        "ip-nexthop-ipv4": [
          {
            "ip-next-hop": "10.0.2.1"
          },
          {
            "ip-next-hop": "10.0.1.1"
          }
        ]
      }
    ],
```

8. Set VRRP-A.

```
    "vrrp-a": {
      "set-id": 1
    },
```

9. Set a Terminal Idle Timeout.

```
    "terminal": {
      "idle-timeout": 0
    },
```

10. Set VRID.
    The default value of VRID is 0. For vThunder instance 1, the default priority is 100,

and for vThunder instance 2, it is 99 (100-1).

```
"vrid-list": [
  {
    "vrid-val": 0,
    "blade-parameters": {
      "priority": 100
    }
  }
],
```

11. Configure SSL.

```
"sslConfig": {
  "requestTimeOut": 40,
  "Path": "server.pem",
  "File": "server",
  "certificationType": "pem"
},
```

NOTE: By default, SSL configuration is disabled i.e., no SSL configuration is applied.

The server.pem file is available in the downloaded CFT folder. You can edit this file as required or use a different certificate file, but the path should be changed accordingly.

12. Configure GLM account details.
You can get the **Entitlement Token** from GLM > **Licenses** > select your license > **Overview** > **Info** tab.

```
"user_name": {
  "value": "xxxxxxx@a10networks.com"
},
"user_password": {
  "value": "user_password"
},
"entitlement_token": {
  "value": "XXXXXXXXXX"
},
```

13. Configure SLB server ports.

```
    "port-list": [
    {
      "port-number": 53,
      "protocol": "udp"
    },
    {
      "port-number": 80,
      "protocol": "tcp"
    },
    {
      "port-number": 443,
      "protocol": "tcp"
    }
  ],
```

14. Verify if all the configurations in the CFT_TMPL_3NIC_2VM_HA_GLM_PUBVIP_BACKAUTO_CONFIG_SSL_SLB_HA_GLM_PARAM.json file are correct and then save the changes.

Deploy vThunder as an SLB

To deploy vThunder on AWS cloud as an SLB, perform the following steps:

1. From your command prompt, navigate to the folder where you have downloaded the CFT.
2. Run the following command to create vThunder SLB instances:

```
python ./CFT_TMPL_3NIC_2VM_HA_GLM_PUBVIP_BACKAUTO_CONFIG_SSL_SLB_HA_GLM_3.py
```

A message is prompted to upload the SSL certificate and configure HA.

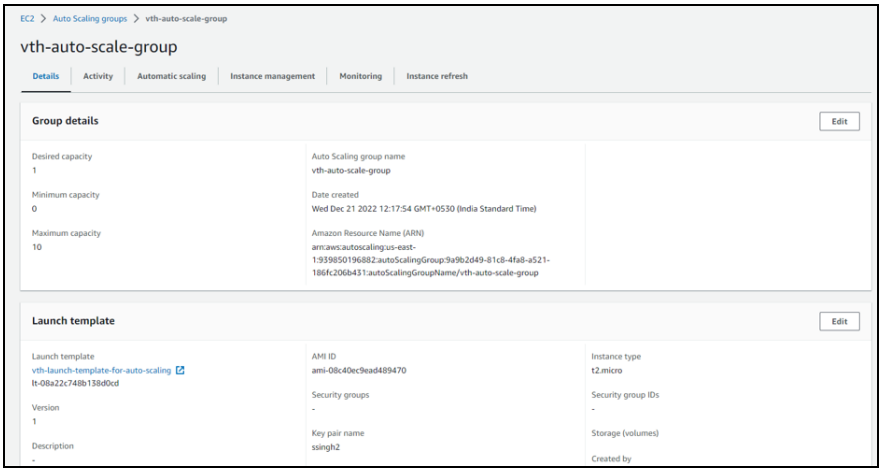
```
Do you want to upload ssl certificate(yes/no)?yes
Do you want to configure GLM (yes/no)?yes
Do you want to configure HA (yes/no)?yes
Configuring vThunder with instance id i-0daaab8a0d8a4b461
configured ethernet ip
configured ethernet ip
Configure service group
```

```
Configured virtual servers
SSL Configured.
License activation completed.
Configured primary dns
configure glm.
Glm license request sent successfully.
Successfully Configured IP route
Configured Vrrp A common configuration
Configured idle timeout
Configured vrrp rid
Configured peer group
Configurations are saved on partition: shared
Updated desired capacity of autoscale group to 1
Configuring vThunder with instance id i-056836e60d88b9eec
configured ethernet ip
configured ethernet ip
Configure service group
Configured virtual servers
SSL Configured.
License activation completed.
Configured primary dns
configure glm.
Glm license request sent successfully.
Successfully Configured IP route
Configured Vrrp A common configuration
Configured idle timeout
Configured vrrp rid
Configured peer group
Configurations are saved on partition: shared
Updated desired capacity of autoscale group to 1
```

If the SSL Certificate upload is successful, a message 'SSL Configured' is displayed.

3. Verify the desired capacity of the autoscale server from **AWS Management Console** > **EC2** > **AutoScaling Group** > <autoscale_group_name>.

Figure 60 : AutoScaling Group Details



Verify Deployment

To verify vThunder SLB deployment using CFT, perform the following steps:

1. Run the following command on Master controller using CLI:

```
vThunder-Active(config)#show running-config
```

If the deployment is successful, the following SLB configuration is displayed:

```
!Current configuration: 349 bytes
!Configuration last updated at 10:56:58 GMT Fri Jan 6 2023
!Configuration last saved at 10:53:34 GMT Fri Jan 6 2023
!64-bit Advanced Core OS (ACOS) version 5.2.1, build 153 (Dec-11-
2020,16:36)
!
vrrp-a common
  device-id 1
  set-id 1
  enable
!
terminal idle-timeout 0
!
ip dns primary 8.8.8.8
!
```

```
!
glm use-mgmt-port
glm enable-requests
glm token vTh205fe920b
!
interface ethernet 1
  enable
  ip address dhcp
!
interface ethernet 2
  enable
  ip address dhcp
!
vrrp-a vrid 0
  floating-ip 10.0.3.12
  blade-parameters
    priority 99
!
vrrp-a peer-group
  peer 10.0.2.117
  peer 10.0.2.173
!
ip route 0.0.0.0 /0 10.0.2.1
ip route 0.0.0.0 /0 10.0.1.1
!
slb server i-0b3aad427ca3c2f23 10.0.3.23
  port 53 udp
  port 80 tcp
  port 443 tcp
!
slb service-group sg443 tcp
  member i-0b3aad427ca3c2f23 443
!
slb service-group sg53 udp
  member i-0b3aad427ca3c2f23 53
!
slb service-group sg80 tcp
```

```
  member i-0b3aad427ca3c2f23 80
!
slb virtual-server vip 10.0.2.121
  port 53 udp
    source-nat auto
    service-group sg53
  port 80 http
    source-nat auto
    service-group sg80
  port 443 https
    source-nat auto
    service-group sg443
!
!
end
!Current config commit point for partition 0 is 0 & config mode is
classical-mode
vThunder-Active(config)#
```

2. Run the following command on vThunder instance 1 using CLI:

```
vThunder-Active(config)#show pki cert
```

If the deployment is successful, the following SSL configuration is displayed:

```
Name    Type                        Expiration  Status
-----------------------------------------------------------------
server certificate Jan 28 12:00:00 2028 GMT [Unexpired, Bound]
```

At this point, the vThunder instance 2 has the following prompt:

```
vThunder-Standby(config)#
```

Figure 61 : vThunder instance 1 - Active

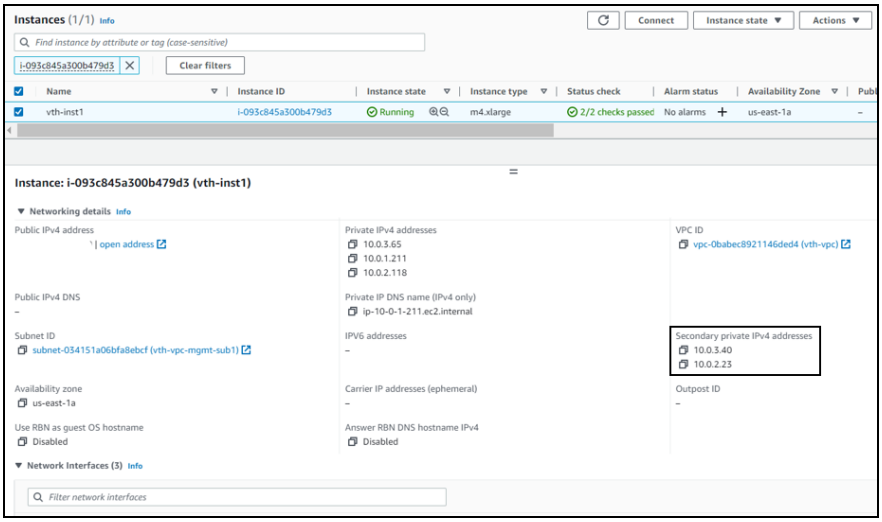
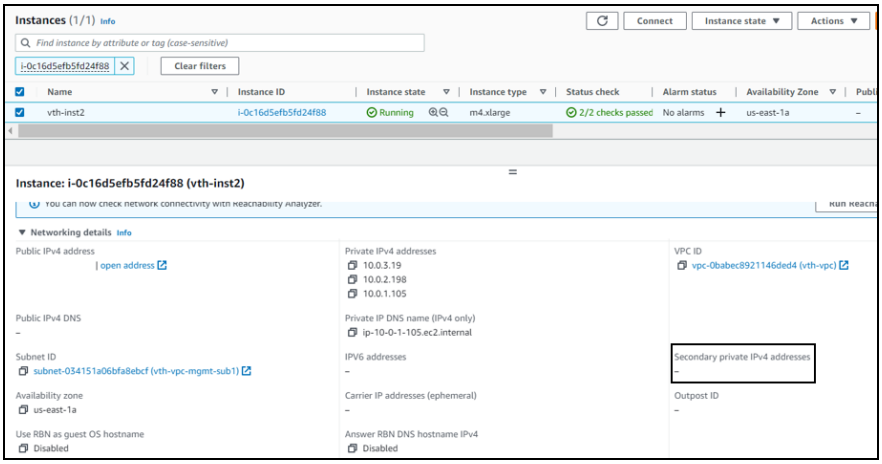


Figure 62 : vThunder instance 2 - Standby



3. Run the following command on vThunder instance 1 using CLI:

```
vThunder-Active(config)#vrrp-a force-self-standby enable
vThunder-Active(config)#
vThunder-ForcedStandby(config)#
```

At this point, IP switching occurs and the vThunder instance 2 prompt becomes:

```
vThunder-Active(config)#
```

4. Run the following command on vThunder instance 2 using CLI:

```
vThunder-Active(config)#show running-config
```

If the deployment is successful, the following SLB configuration is displayed:

```
!Current configuration: 282 bytes
!Configuration last updated at 10:53:35 GMT Fri Jan 6 2023
!Configuration last saved at 10:53:37 GMT Fri Jan 6 2023
!64-bit Advanced Core OS (ACOS) version 5.2.1, build 153 (Dec-11-
2020,16:36)
!
vrrp-a common
  device-id 2
  set-id 1
  enable
!
terminal idle-timeout 0
!
ip dns primary 8.8.8.8
!
!
glm use-mgmt-port
glm enable-requests
glm token vTh205fe920b
!
interface ethernet 1
  enable
  ip address dhcp
!
interface ethernet 2
  enable
  ip address dhcp
!
vrrp-a vrid 0
  floating-ip 10.0.3.12
  blade-parameters
    priority 98
!
vrrp-a peer-group
```

```
  peer 10.0.2.117
  peer 10.0.2.173
!
ip route 0.0.0.0 /0 10.0.2.1
ip route 0.0.0.0 /0 10.0.1.1
!
slb server i-0b3aad427ca3c2f23 10.0.3.23
  port 53 udp
  port 80 tcp
  port 443 tcp
!
slb service-group sg443 tcp
  member i-0b3aad427ca3c2f23 443
!
slb service-group sg53 udp
  member i-0b3aad427ca3c2f23 53
!
slb service-group sg80 tcp
  member i-0b3aad427ca3c2f23 80
!
slb virtual-server vip 10.0.2.121
  port 53 udp
    source-nat auto
    service-group sg53
  port 80 http
    source-nat auto
    service-group sg80
  port 443 https
    source-nat auto
    service-group sg443
!
!
end
!Current config commit point for partition 0 is 0 & config mode is
classical-mode
vThunder-Active(config)#
```

5. Run the following command on vThunder instance 2 using CLI:

```
vThunder-Active(config)#show pki cert
```

If the deployment is successful, the following SSL configuration is displayed:

```
Name     Type                          Expiration  Status
-----------------------------------------------------------------
server certificate Jan 28 12:00:00 2028 GMT [Unexpired, Bound]
```

Figure 63 : vThunder instance 1 - Standby

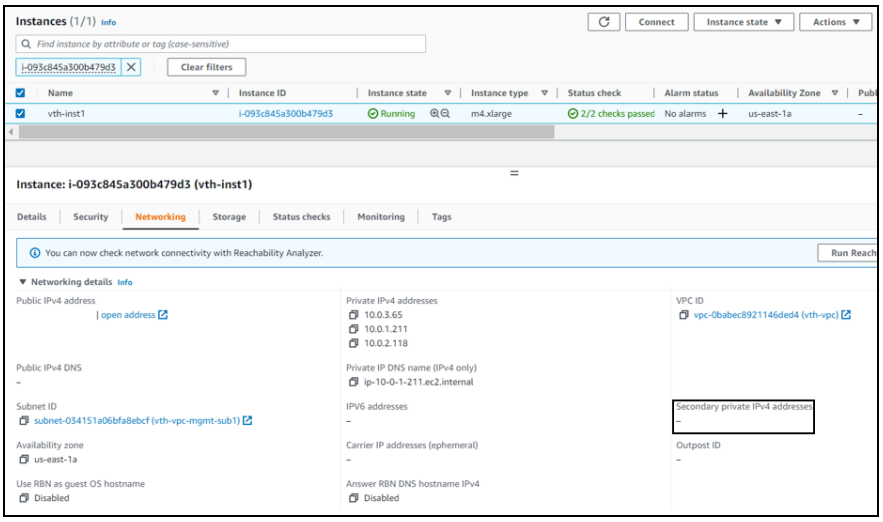
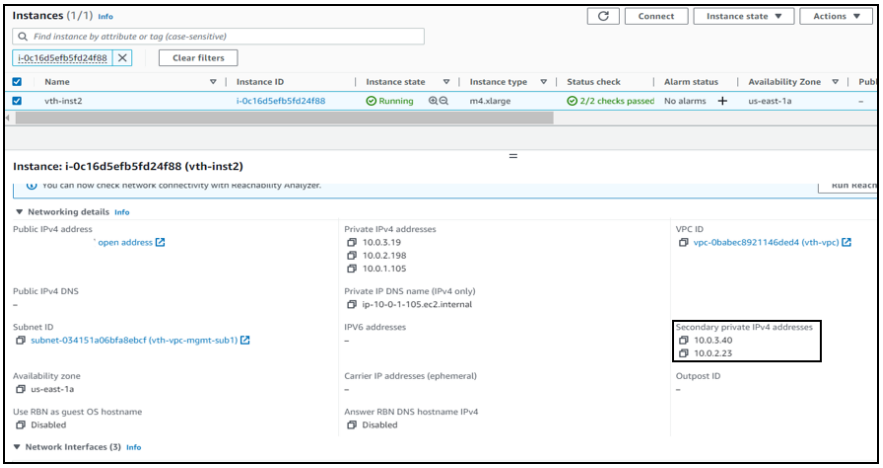


Figure 64 : vThunder instance 2 - Active



6. If you want to make vThunder instance 1 active, run the following command on vThunder instance 2 using CLI:

```
vThunder-Active(config)#vrrp-a force-self-standby disable
vThunder-Active(config)#
vThunder-ForcedStandby(config)#
```

At this point, IP switching occurs and the vThunder instance 1 prompt becomes:

```
vThunder-Active(config)#
```

Verify GLM

The application of license can be verified using any of the following ways:

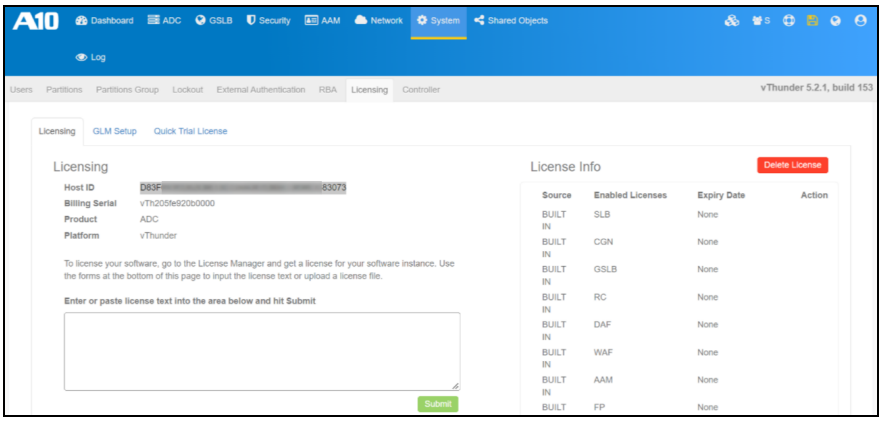
- Verify License using GUI
- Verify License using CLI

Verify License using GUI

To verify license using GUI, perform the following steps:

1. Log in to your vThunder instance using Public IPv4 address.
   Here, **`vth-inst1`** is the vThunder instance.
2. Navigate to **Profile** > **Setting** > **Licensing**.
3. Click the **Licensing** tab.
   If the license is successfully applied on vThunder, the Host ID is displayed.

   Figure 65 : GLM > Licensing window



Verify License using CLI

To verify license using CLI, perform the following steps:

1. Verify if NOLICENSE is removed from the vThunder prompt.
2. Run the following command on vThunder:

```
vThunder(config)#show license
```

If the license is successfully applied on vThunder, the Host ID is displayed:

```
Host ID         : D83F****82EB633D****067DB84136****83073
```

3. Run the following command on vThunder instance:

```
vThunder(config)#show license-info
```

If the license is successfully applied on vThunder, the following GLM configuration is displayed:

```
Host ID         : D83F****82EB633D****067DB84136****83073
USB ID          : Not Available
Billing Serials: vTh205fe920b0000
Token           : Not Available
Product         : ADC
Platform        : vThunder
Burst           : Disabled
GLM Ping Interval In Hours : 24
-------------------------------------------------------------------------
-------
Enabled Licenses   Expiry Date (UTC)         Notes
-------------------------------------------------------------------------
-------
SLB                None
CGN                None
GSLB               None
RC                 None
DAF                None
WAF                None
AAM                None
FP                 None
WEBROOT            N/A              Requires an additional Webroot
```

```
license.
THREATSTOP          N/A              Requires an additional ThreatSTOP
license.
QOSMOS              N/A              Requires an additional QOSMOS
license.
WEBROOT_TI          N/A              Requires an additional Webroot
Threat Intel license.
IPSEC_VPN           N/A              Requires an additional IPsec VPN
license.
500 Mbps Bandwidth 20-January-2023
```

Verify Traffic Flow

For this template, the traffic flow is verified using the following:

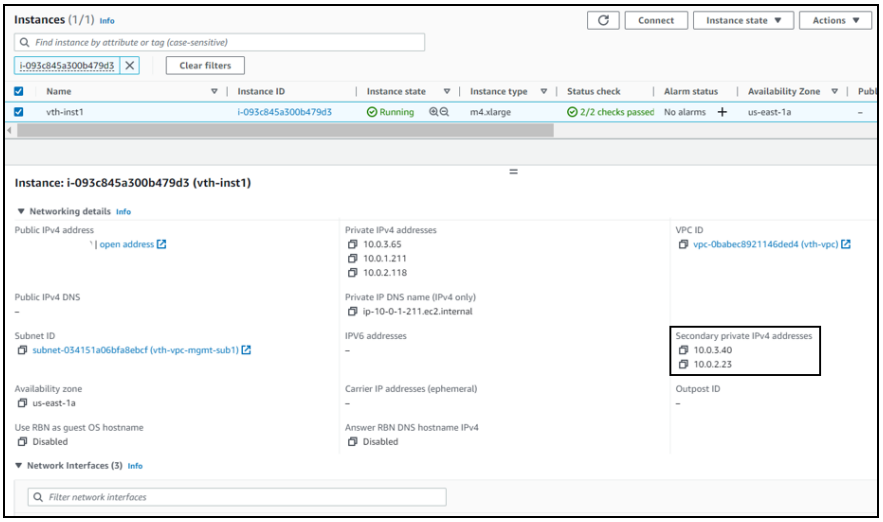
- Secondary Private IPv4 Address
- Allocated Public IPv4 address

Secondary Private IPv4 Address

To verify the traffic flow from client machine to server machine via vThunder instance secondary private IPv4 address, perform the following:

1. From **AWS Management Console**, navigate to **EC2** > **Instances**.
2. Select the active vThunder instance name and then click the **Networking** tab.
3. Copy the IP address of the vThunder Secondary IPv4 Address under the **Private IPv4 address**.

Figure 66 : vThunder instance 1 - Networking tab



4. Select your client instance from the **Instances** list.
   Here, **`vth-client`** is the client instance name.
5. Click **Connect**.
   A **Connect to instance** window with **EC2 Instance Connect** tab is displayed.
6. Click **Connect**.
   A **Terminal** window is displayed.
7. Run the following command in the Terminal window to send the traffic from the client machine:

   ```
   curl <vThunder_instance_secondary_private_IPv4_Address>
   ```

   Example

   ```
   curl 10.0.2.23
   ```

8. Verify if a response is received.

Allocated Public IPv4 address

To verify the traffic flow from client machine to server machine via vThunder instance allocated public IPv4 address, perform the following:

1. Select the active vThunder instance name and then click its **Networking** tab.
2. Copy the Allocated IPv4 address of data interface under the **Elastic IP address**.
   Here, **`vth-inst1-data-nic1-ip`** is the Elastic Public IP of data interface.

Figure 67 : Elastic IP addresses

3. Select your client instance from the **Instances** list.
   Here, **`vth-client`** is the client instance name.
4. Click **Connect**.
   A **Connect to instance** window with **EC2 Instance Connect** tab is displayed.
5. Click **Connect**.
   A **Terminal** window is displayed.
6. Run the following command in the Terminal window to send the traffic from client machine:

   ```
   curl <Allocated_IPv4_address_of_data_interface>
   ```

   Example

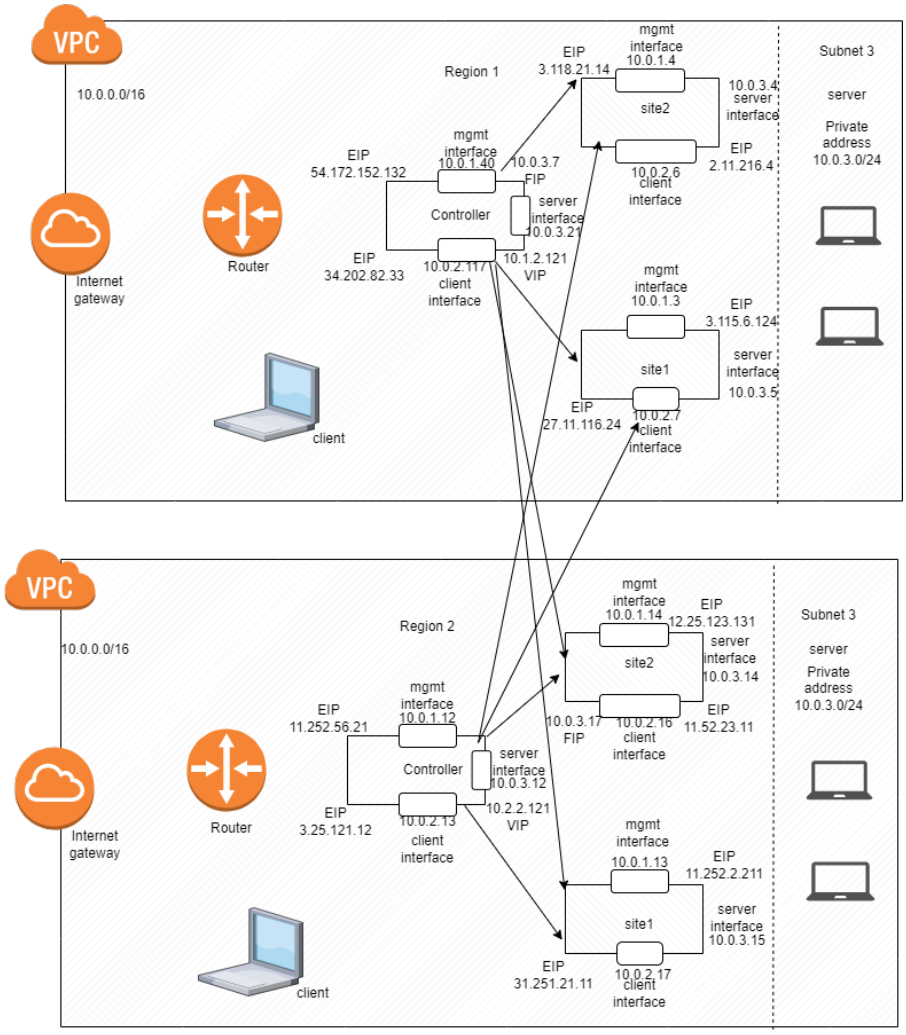
   ```
   curl 18.210.210.10
   ```

7. Verify if a response is received.

Deploy CFT A10-vThunder_ADC-3NIC-6VM-2RG-GSLB

Using this template, two regions each containing one GSLB controller and two site devices can be deployed. A server is assigned to each site devices.

Figure 68 : 3NIC-6VM-2RG-GSLB Deployment Topology



The following topics are covered:

System Requirements

The CFT displays the default values when you download and save the files on your local machine. You can modify the default values as required for your deployment.

You need to configure the following resources to deploy vThunder on the AWS cloud:

Table 13 : System Requirements

Resource Name	Description	Default Value
Stack	Two stacks are created with the specified name and location.	Here, the stack name used are: **vth-stack1** **vth-stack2**
Network Interface Card [NIC]	Total nine interfaces are created for each region. Two types of interfaces are created for each vThunder instance: • One management interface • Two data interfaces	**vth-stack1-inst1-mgmt-nic1** **vth-stack1-inst1-data-nic1** **vth-stack1-inst1-data-nic2** **vth-stack1-inst2-mgmt-nic1** **vth-stack1-inst2-data-nic1** **vth-stack1-inst2-data-nic2**

Resource Name	Description	Default Value
		vth-stack1-inst3-mgmt-nic1 vth-stack1-inst3-data-nic1 vth-stack1-inst3-data-nic2 vth-stack2-inst1-mgmt-nic1 vth-stack2-inst1-data-nic1 vth-stack2-inst1-data-nic2 vth-stack2-inst2-mgmt-nic1 vth-stack2-inst2-data-nic1 vth-stack2-inst2-data-nic2 vth-stack2-inst3-mgmt-nic1 vth-stack2-inst3-data-nic1 vth-stack2-inst3-data-nic2
Subnet	Three subnets are created with an address prefix each, for a region.	vth-stack1-vpc-mgmt-sub1 vth-stack1-vpc-data-sub1

Resource Name	Description	Default Value
		vth-stack1-vpc-data-sub2 **vth-stack2-vpc-mgmt-sub1** **vth-stack2-vpc-data-sub1** **vth-stack2-vpc-data-sub2**
Virtual Private Network [VCN]	A virtual private network is assigned to the virtual machine instance.	**vth-stack1-vpc** **vth-stack2-vpc** Address prefix for virtual network: 10.0.0.0/16
Elastic Public IP	Six Elastic Public IPs are created and attached to management and data interface of vThunder instances for each region.	**vth-stack1-inst1-mgmt-nic1-ip** **vth-stack1-inst1-data1-nic1-ip** **vth-stack1-inst2-mgmt-nic1-ip** **vth-stack1-inst2-data1-nic1-ip** **vth-stack1-inst3-mgmt-nic1-ip** **vth-stack1-inst3-data1-nic1-ip** **vth-stack2-inst1-mgmt-nic1-ip** **vth-stack2-inst1-**

Resource Name	Description	Default Value
		data1-nic1-ip **vth-stack2-inst2-mgmt-nic1-ip** **vth-stack2-inst2-data1-nic1-ip** **vth-stack2-inst3-mgmt-nic1-ip** **vth-stack2-inst3-data1-nic1-ip**
Security Group	Two security groups are created for each region: • One security group for management interface. • One security group for client-side data interface and server-side data interface. **Logical name**: • vThunderSecurityGroupMgmt • vThunderSecurityGroupData	Here, the tag names are: **vth-stack1-sg-mgmt** **vth-stack1-sg-data** **vth-stack2-sg-mgmt** **vth-stack2-sg-data**
vThunder Instance	Three vThunder EC2 instances are created for each region. **Default type**: m4.xlarge (40 Gb memory) Table 14 lists the supported instance types.	**vth-stack1-controller-region1** **vth-stack1-site1-region1** **vth-stack1-site2-region1** **vth-stack2-controller-region2**

Resource Name	Description	Default Value
		`vth-stack2-site1-region2` `vth-stack2-site2-region2`

Server Instance	Two Ubuntu Server instances are created for each region.
	Default Size: t2.micro
		`vth-stack1-server1`
		`vth-stack1-server2`
		`vth-stack2-server1`
		`vth-stack2-server2`

Supported Instance Types

Table 14 : List of Supported Instance Types

Instance	vCPU	Memory	Number of Network Interfaces
c4.xlarge	4	7680	4
c4.4xlarge	16	30720	8
c4.8xlarge	36	61440	8
d2.xlarge	4	31232	4
d2.2xlarge	8	62464	4
d2.4xlarge	16	124928	8
d2.8xlarge	36	249856	8
m4.xlarge	4	16384	4
m4.2xlarge	8	32768	4
m4.4xlarge	16	65536	8
m4.10xlarge	40	163840	8
i2.xlarge	4	31232	4

Instance	vCPU	Memory	Number of Network Interfaces
i2.2xlarge	8	62464	4
i2.4xlarge	16	124928	8
i2.8xlarge	32	249856	8
c5d.large	2	4096	3
c5d.9xlarge	36	73728	8
c5d.2xlarge	8	32768	4
c5d.4xlarge	16	73728	8
c5.xlarge	4	8192	4
c5.2xlarge	8	16384	4
c5.4xlarge	16	32768	8
c5.9xlarge	36	73728	8
g3.4xlarge	16	124928	8
g3.8xlarge	32	249856	8
i3.large	2	15616	3
i3.xlarge	4	31232	4
i3.2xlarge	8	62464	4
i3.4xlarge	16	124928	8
i3.8xlarge	32	249856	8
m5d.large	2	8192	3
m5d.xlarge	4	16384	4
m5d.2xlarge	8	32768	4
m5d.4xlarge	16	65536	8
m5.large	2	8192	3
m5.xlarge	4	16384	4
m5.2xlarge	8	32768	4
m5.4xlarge	16	65536	8
r5d.large	2	16384	3
r5d.xlarge	4	32768	4

Instance	vCPU	Memory	Number of Network Interfaces
r5d.2xlarge	8	65536	4
r5d.4xlarge	16	131072	8
r5.large	2	16384	3
r5.xlarge	4	32768	4
r5.2xlarge	8	65536	4
r5.4xlarge	16	131072	8
r4.large	2	15616	3
r4.xlarge	4	31232	4
r4.2xlarge	8	62464	4
r4.4xlarge	16	124928	8
r4.8xlarge	32	249856	8
t3.medium	2	4096	3
t3.large	2	8192	3
t3.xlarge	4	16384	4
t3.2xlarge	8	32768	4
z1d.large	2	16384	3
z1d.xlarge	4	32768	4
z1d.2xlarge	8	65536	4
z1d.3xlarge	12	98304	8
z1d.6xlarge	24	196608	8

Create vThunder Instances

The vThunder instances are created in two regions:

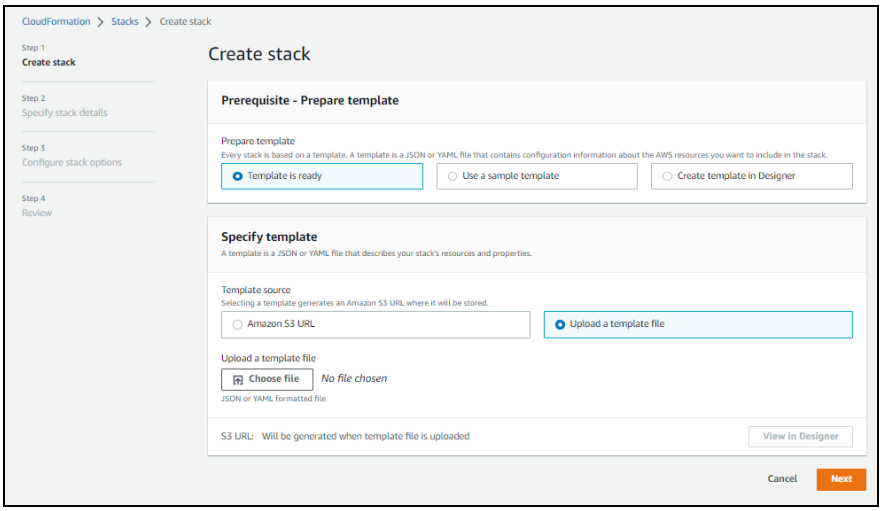
- vThunder instances for Region1
- vThunder instances for Region2

vThunder instances for Region1

To create vThunder instances for region1, perform the following steps:

1. From **AWS Management Console**, navigate to **CloudFormation** > **Stacks** > **Create Stack** > **With new resources (standard)**.
   The **Create stack** window is displayed.

   Figure 69 : Create stack window

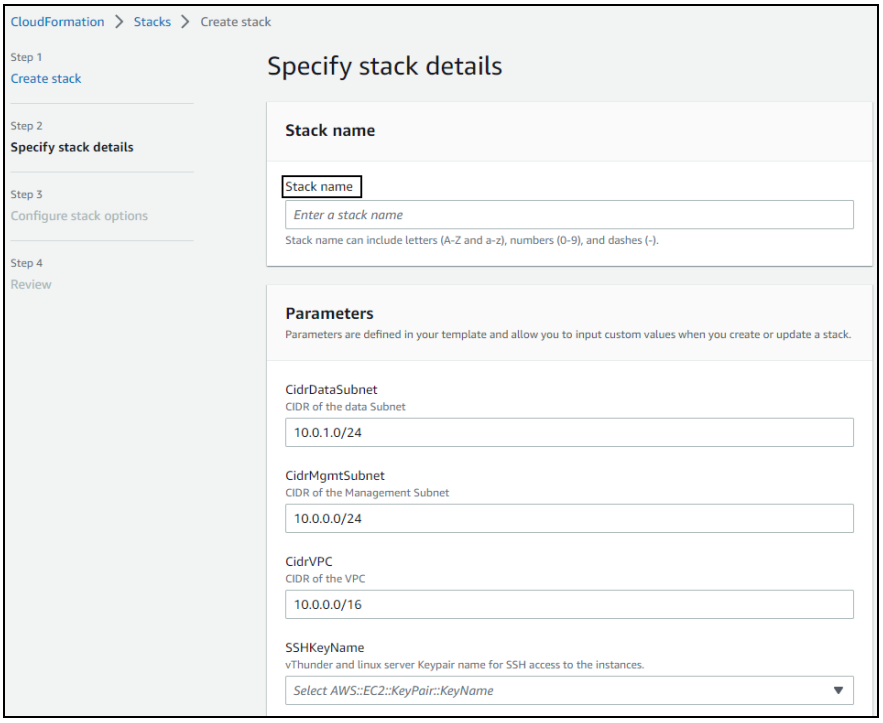


2. Select region1 from the dropdown on the **AWS Management Console** top ribbon.

   Figure 70 : Region dropdown

3. In the **Prerequisite - Prepare template** section, select **Template is ready** option. After the selecting this option, the **Specify template** section is displayed.
4. In the **Specify template** section, select **Upload a template file** and click **Choose file** to browse and upload each of the following template files from the downloaded CFT folder:
   CFT_TMPL_3NIC_6VM_2RG_GSLB_REGION_1
   The selected template file name is displayed as the chosen file.
5. Click **Next**.
   The **Specify stack details** window is displayed.

Figure 71 : Specify stack details window



6. In the **Specify stack details** window, enter or select the following:
   a. In the **Stack name** section, enter a **Stack name**.

      Here, **`vth-stack1`** is the stack name.

   b. In the **Parameters** section, enter or select the required value in the following fields:
      - **KeyPairName**: *your SSH key*
      - **TagValue**: **`a10-vthunder-adc`**
      - **Zone:** *your availability zone*

   c. Verify the other fields and change the values as required. (Optional)
      If the default IP addresses of the following fields changed then the updated IP addresses must be within CIDR range of Data Subnet 1:
      - PrimaryIpGSLBMaster
      - PrimaryIpSite1
      - PrimaryIpSite2

- VIPGSLBMaster
- VIPSite1
- VIPSite2

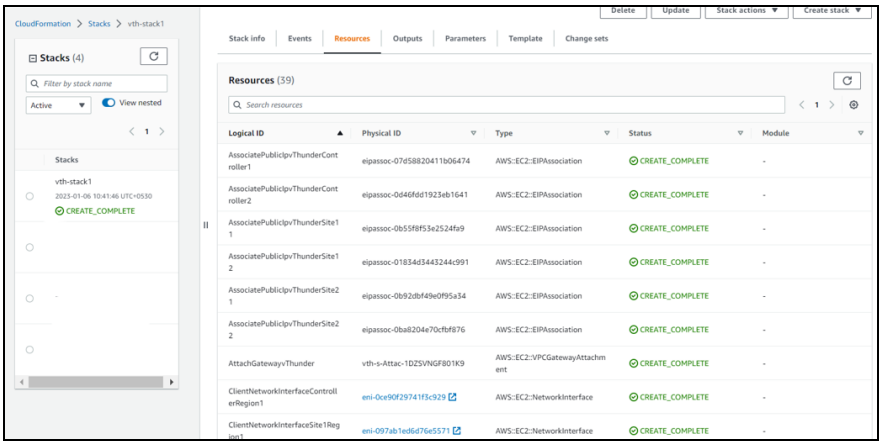
NOTE: 10.1.2.0/24 is default CIDR of region1.

7. Click **Next**.
   The **Configure stack options** window is displayed.
8. Verify the fields and change the values as required. (Optional)
9. Click **Next**.
   The **Review** window is displayed.
10. Verify if all the stack configurations are correct and then click **Submit**.

NOTE: The system may take a few minutes to create the resources. So, wait until the stack status is **CREATE_COMPLETE**.

11. Verify if all the above resources are created in the **AWS Management Console** > **CloudFormation** > **Stacks** > <stack_name> > **Resources** tab.

Figure 72 : Resource listing in the resource group



The three vThunder instances with the following Logical IDs for region1 are listed under **Resources** tab:

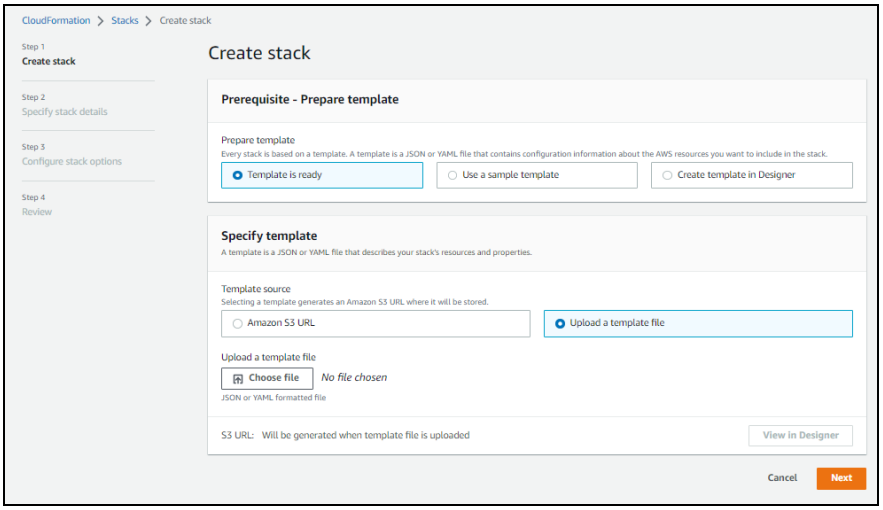
- vThunderControllerRegion1 (`vth-stack1-controller-region1`)
- vThunderSite1Region1 (`vth-stack1-site1-region1)`

- vThunderSite2Region1 (`vth-stack1-site2-region1`)

vThunder instances for Region2

1. From **AWS Management Console**, navigate to **CloudFormation** > **Stacks** > **Create Stack** > **With new resources (standard)**.
   The **Create stack** window is displayed.

   Figure 73 : Create stack window

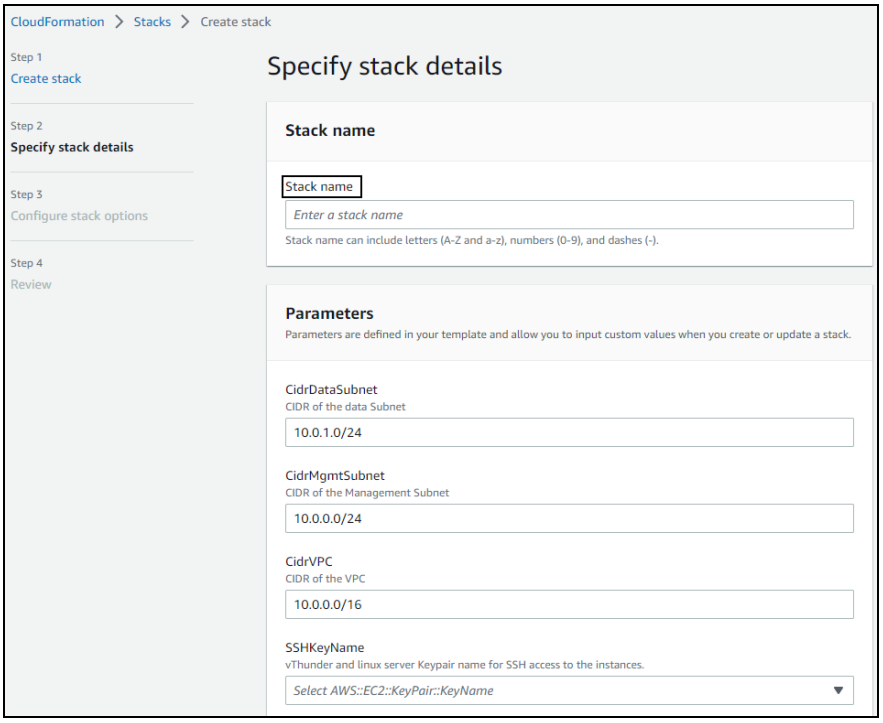


2. Select region2 from the dropdown on the **AWS Management Console** top ribbon.

   Figure 74 : Region dropdown

3. In the **Prerequisite - Prepare template** section, select **Template is ready** option. After the selection of this option, the **Specify template** section is displayed.
4. In the **Specify template** section, select **Upload a template file** and click **Choose file** to browse and upload each of the following template files from the downloaded CFT folder:
   CFT_TMPL_3NIC_6VM_2RG_GSLB_REGION_2
   The selected template file name is displayed as the chosen file.
5. Click **Next**.
   The **Specify stack details** window is displayed.

Figure 75 : Specify stack details window



6. In the **Specify stack details** window, enter or select the following:
   a. In the **Stack name** section, enter a **Stack name**.

      Here, **`vth-stack2`** is the stack name.

   b. In the **Parameters** section, enter or select the required value in the following fields:
      - **KeyPairName**: *your SSH key*
      - **TagValue**: **`a10-vthunder-adc`**
      - **Zone:** *your availability zone*

   c. Verify the other fields and change the values as required. (Optional)
      If the default IP addresses of the following fields changed then the updated IP addresses must be within CIDR range of Data Subnet 1:
      - PrimaryIpGSLBMaster
      - PrimaryIpSite1
      - PrimaryIpSite2

- VIPGSLBMaster
- VIPSite1
- VIPSite2

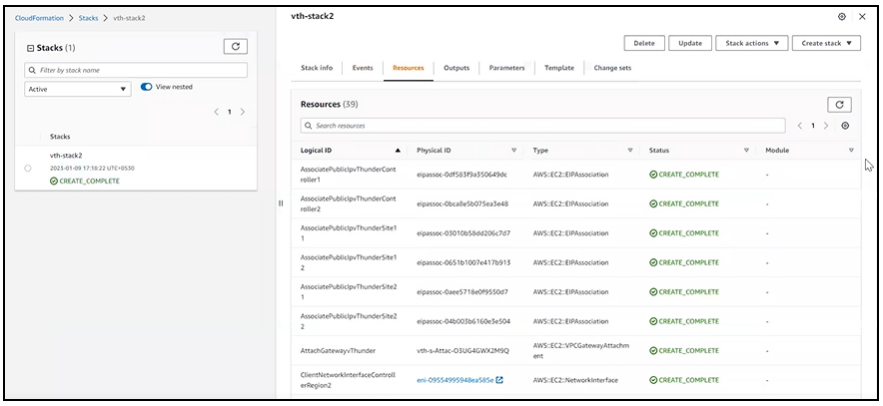
NOTE: 10.2.2.0/24 is default CIDR of region2.

7. Click **Next**.
   The **Configure stack options** window is displayed.
8. Verify the fields and change the values as required. (Optional)
9. Click **Next**.
   The **Review** window is displayed.
10. Verify if all the stack configurations are correct and then click **Submit**.

NOTE: It may take the system a few minutes to create the resources. So, wait until the stack status is **CREATE_COMPLETE**.

11. Verify if all the above resources are created in the **AWS Management Console** > **CloudFormation** > **Stacks** > <stack_name> > **Resources** tab.

Figure 76 : Resource listing in the resource group



The three vThunder instances with the following Logical IDs for region2 are listed under **Resources** tab:

- vThunderControllerRegion2 (`vth-stack2-controller-region2`)
- vThunderSite1Region2 (`vth-stack2-site1-region2`)
- vThunderSite2Region2 (`vth-stack2-site2-region2`)

Access vThunder using GUI or CLI

vThunder instances can be accessed using any of the following ways:

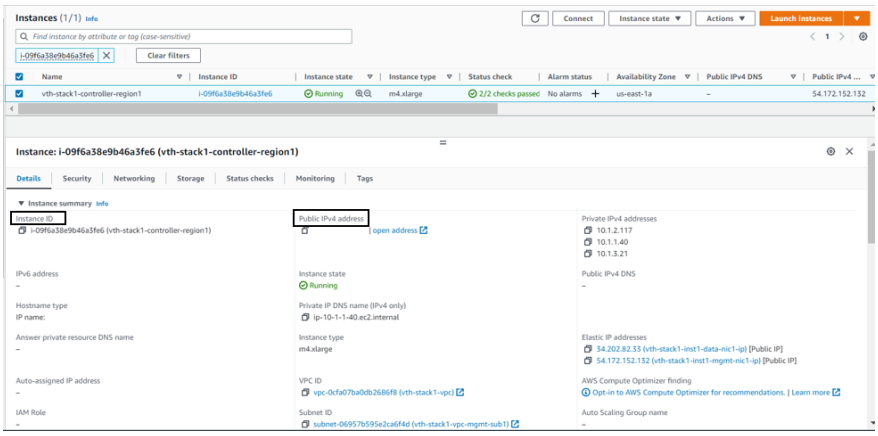
- Access vThunder using GUI
- Access vThunder using CLI

Access vThunder using GUI

To access vThunder instances using GUI, perform the following steps:

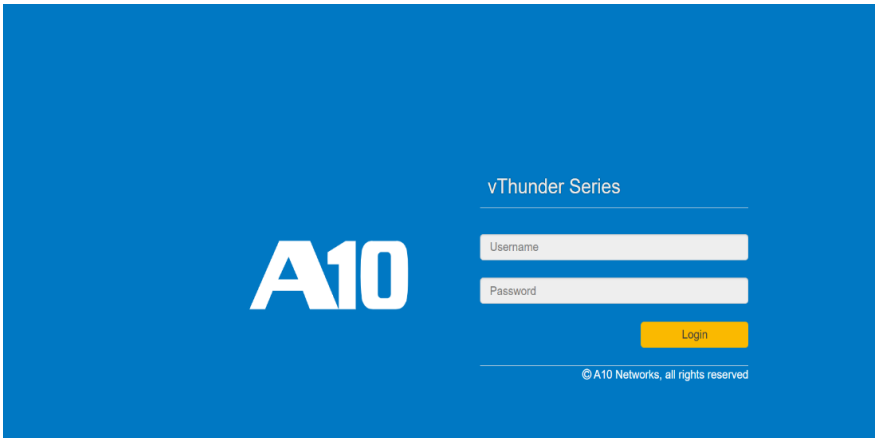
1. From **AWS Management Console**, navigate to **EC2** > **Instances**.
2. Select your master controller instance name.
   Here, **`vth-stack1-controller-region1`** is the master controller instance name.

   Figure 77 : Master Controller Instance Name



3. Copy the **Public IPv4 address** from the **Details** tab and replace the IP address in the below link:
   http://<vThunder_public_IPv4_address>
4. Open the updated link in any browser.
   The vThunder login window is displayed.

Figure 78 : vThunder GUI



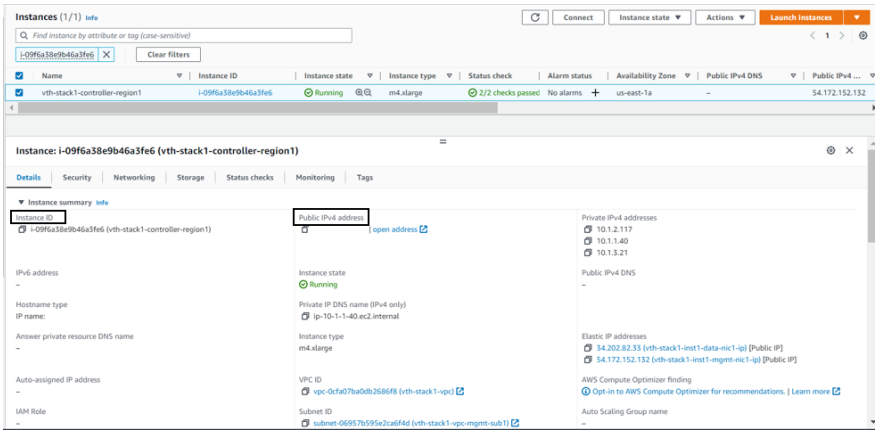
5. Enter the following credentials:
   - Username – admin
   - Password – *EC2 Instance ID*
     The home page is displayed if the entered credentials are correct.
6. Similarly, verify the following vThunder instances of region1 and region2 are up and running:
   - **`vth-stack1-site1-region1`**
   - **`vth-stack1-site2-region1`**
   - **`vth-stack2-controller-region2`**
   - **`vth-stack2-site1-region2`**
   - **`vth-stack2-site2-region2`**

Access vThunder using CLI

To access vThunder instances using CLI, perform the following steps:

1. From **AWS Management Console**, navigate to **EC2** > **Instances**.
2. Select your master controller instance name.
   Here, **`vth-stack1-controller-region1`** is the master controller instance name.

Figure 79 : Master Controller Instance Name



3. Copy the **Public IPv4 address** from the **Details** tab.
4. Open any SSH client and provide the following to establish a connection:
   - Hostname: Public IPv4 address
   - Username: admin
   - Key: SSH Key
5. Connect to the session.
6. In the SSH client session, run the following commands:

```
vThunder(NOLICENSE)>enable <---Execute command--->
Password:                  <---just press Enter key--->
vThunder(NOLICENSE)#config <---Configuration mode--->
vThunder(config)(NOLICENSE)#
```

7. Similarly, verify the following vThunder instances of region1 and region2 are up and running:
   - **vth-stack1-site1-region1**
   - **vth-stack1-site2-region1**
   - **vth-stack2-controller-region2**
   - **vth-stack2-site1-region2**
   - **vth-stack2-site2-region2**

Configure vThunder as an SLB

The following topics are covered:

Initial Setup

Before deploying vThunder on AWS cloud as an SLB with HA, you need to configure the corresponding parameters in the CFT.

To configure the parameters, perform the following steps:

1. Navigate to the folder where you have downloaded the CFT, and open the CFT_TMPL_3NIC_6VM_2RG_GSLB_CONFIG_GSLB_PARAM.json with a text editor.

   NOTE: Each parameter has a default value mentioned in the parameter file.

2. Configure the stack names for the two regions created using CFT.
   If the default value of the region fields are changed, then change the geo-location values accordingly.

```
   "stackRegionDetails":{
    "value" : [
{
       "region": "us-east-1",
       "stackName":"vth-stack1"
    },
        {
       "region": "us-east-2",
       "stackName":"vth-stack2"
    }
        ]
   },
```

3. Configure SLB server port for site devices.

```
    "parameters": {
        "slbServerPortList1": {
```

```
            "value": [
                {
                    "port-number": 80,
                    "protocol": "tcp",
                    "health-check-disable":0
                }
            ]
        },
        "slbServerPortList2": {
            "value": [
                {
                    "port-number": 80,
                    "protocol": "tcp",
                    "health-check-disable":0
                }
            ]
        },
        "slbServerPortList3": {
            "value": [
                {
                    "port-number": 80,
                    "protocol": "tcp",
                    "health-check-disable":0
                }
            ]
        },
        "slbServerPortList4": {
            "value": [
                {
                    "port-number": 80,
                    "protocol": "tcp",
                    "health-check-disable":0
                }
            ]
        },
```

4. Configure service group port for site devices.

```
        "serviceGroupList1": {
            "value": [
                {
                    "name":"sg",
                    "protocol":"tcp",
                    "health-check-disable":0,
                    "member-list": [
                        {
                            "port":80
                        }
                    ]
                }
            ]
        },
        "serviceGroupList2": {
            "value": [
                {
                    "name":"sg",
                    "protocol":"tcp",
                    "health-check-disable":0,
                    "member-list": [
                        {
                            "port":80
                        }
                    ]
                }
            ]
        },
        "serviceGroupList3": {
            "value": [
                {
                    "name":"sg",
                    "protocol":"tcp",
                    "health-check-disable":0,
                    "member-list": [
                        {
```

```
                            "port":80
                        }
                    ]
                }
            ]
        },
        "serviceGroupList4": {
            "value": [
                {
                    "name":"sg",
                    "protocol":"tcp",
                    "health-check-disable":0,
                    "member-list": [
                        {
                            "port":80
                        }
                    ]
                }
            ]
        },
```

5. Configure SLB virtual server for site devices.
   The virtual server default name is “vs1”.

```
        "virtualServerList1": {
            "virtual-server-name": "vs1",
            "metadata": {
                "description": "virtual server is using VIP from
ethernet 1 secondary subnet"
            },
            "value": [
                {
                    "port-number":80,
                    "protocol":"tcp",
                    "auto":1,
                    "service-group":"sg"
                }
            ]
        },
```

```
        "virtualServerList2": {
            "virtual-server-name": "vs1",
            "metadata": {
                "description": "virtual server is using VIP from
ethernet 1 secondary subnet"
            },
            "value": [
                {
                    "port-number":80,
                    "protocol":"tcp",
                    "auto":1,
                    "service-group":"sg"
                }
            ]
        },
        "virtualServerList3": {
            "virtual-server-name": "vs1",
            "metadata": {
                "description": "virtual server is using VIP from
ethernet 1 secondary subnet"
            },
            "value": [
                {
                    "port-number":80,
                    "protocol":"tcp",
                    "auto":1,
                    "service-group":"sg"
                }
            ]
        },
        "virtualServerList4": {
            "virtual-server-name": "vs1",
            "metadata": {
                "description": "virtual server is using VIP from
ethernet 1 secondary subnet"
            },
            "value": [
```

```
                {
                    "port-number":80,
                    "protocol":"tcp",
                    "auto":1,
                    "service-group":"sg"
                }
            ]
        },
```

6. Configure GSLB service IP address for controller.

```
        "serviceipList1": {
            "node-name": "vs1",
            "value": [
                {
                    "port-num": 80,
                    "port-proto": "tcp"
                }
            ]
        },
        "serviceipList2": {
            "node-name": "vs2",
            "value": [
                {
                    "port-num": 80,
                    "port-proto": "tcp"
                }
            ]
        },
        "serviceipList3": {
            "node-name": "vs3",
            "value": [
                {
                    "port-num": 80,
                    "port-proto": "tcp"
                }
            ]
        },
        "serviceipList4": {
```

```
            "node-name": "vs4",
            "value": [
                {
                    "port-num": 80,
                    "port-proto": "tcp"
                }
            ]
        },
```

7. Configure GSLB site details for controller.

```
        "siteList1": {
            "site-name": "eastus_1",
            "vip-name": "vs1",
            "device-name": "slb1",
            "geo-location": "North America,United States"
        },
        "siteList2": {
            "site-name": "eastus_2",
            "vip-name": "vs2",
            "device-name": "slb2",
            "geo-location": "North America,United States"
        },
        "siteList3": {
            "site-name": "eastus2_1",
            "vip-name": "vs3",
            "device-name": "slb3",
            "geo-location": "North America.United States.California.San
Jose"
        },
        "siteList4": {
            "site-name": "eastus2_2",
            "vip-name": "vs4",
            "device-name": "slb4",
            "geo-location": "North America.United States.California.San
Jose"
        },
```

8. Configure system geo location details for controller.

```
        "geolocation": {
            "geo-location-iana": "0",
            "geo-location-geolite2-city": "1",
            "geolite2-city-include-ipv6": "0",
            "geo-location-geolite2-country": "0"
        },
```

9. Configure GSLB DNS policy for controller.

```
        "dnsPolicy": {
            "policy-name": "a10",
            "type": "health-check, geographic"
        },
```

10. Configure GSLB virtual server for controller.

```
        "gslbserverList1": {
            "virtual-server-name": "gslb-server",
            "use-if-ip": 1,
            "ethernet": 1,
            "metadata": {
                "description": "gslb virtual server is using VIP from
ethernet 1 secondary subnet"
            },
            "value": [
                {
                    "port-number":53,
                    "protocol":"udp",
                    "gslb-enable": 1
                }
            ]
        },
        "gslbserverList2": {
            "virtual-server-name": "gslb-server",
            "use-if-ip": 1,
            "ethernet": 1,

            "metadata": {
                "description": "gslb virtual server is using VIP from
ethernet 1 secondary subnet"
```

```
            },
            "value": [
                {
                    "port-number":53,
                    "protocol":"udp",
                    "gslb-enable": 1
                }
            ]
        },
```

11. Configure GSLB protocol status for controller.

```
        "gslbprotocolStatus": {
            "status-interval": 1
        },
```

12. Configure GSLB group for controller.

```
        "gslbcontrollerGroup1": {
            "name": "default",
            "priority": 255
        },
        "gslbcontrollerGroup2": {
            "name": "default",
            "priority": 100
        },
```

13. Configure GSLB zone for controller.

```
        "gslbzone": {
            "service-port": 80,
            "service-name": "www",
            "name" : "gslb.a10.com"
        },
```

14. Configure default route for vThunder instances.

```
        "defaultroute1":
        {
            "next-hop": "10.1.2.1"
        },
        "defaultroute2":
        {
```

```
            "next-hop": "10.2.2.1"
        }
```

15. Verify if all the configurations in the CFT_TMPL_3NIC_6VM_2RG_GSLB_CONFIG_GSLB_PARAM.json file are correct and then save the changes.

Deploy vThunder as an SLB

To deploy vThunder on AWS cloud as an SLB, perform the following steps:

1. From your command prompt, navigate to the folder where you have downloaded the CFT.
2. Run the following command to create vThunder SLB instances:

```
python ./CFT_TMPL_3NIC_6VM_2RG_GSLB_CONFIG_GSLB_3.py
```

If the deployment is successful, the following configuration is displayed:

```
Gathering public and private ip addresses for site devices
configured ethernet- 1 ip
configured ethernet- 2 ip
Configuring slb server for site: i-0dd6dc4a32d78c35b
Successfully Configured slb server for site: i-0dd6dc4a32d78c35b
Configuring service group for site: i-0dd6dc4a32d78c35b
Successfully Configured service group for site:i-0dd6dc4a32d78c35b
Successfully Configured virtual server for site: i-0dd6dc4a32d78c35b
Successfully Configured gslb site: i-0dd6dc4a32d78c35b
Successfully Configured default route:i-0dd6dc4a32d78c35b
Configurations are saved on partition: shared
configured ethernet- 1 ip
configured ethernet- 2 ip
Configuring slb server for site: i-0f59e5109420618b5
Successfully Configured slb server for site: i-0f59e5109420618b5
Configuring service group for site: i-0f59e5109420618b5
Successfully Configured service group for site:i-0f59e5109420618b5
Successfully Configured virtual server for site: i-0f59e5109420618b5
Successfully Configured gslb site: i-0f59e5109420618b5
Successfully Configured default route:i-0f59e5109420618b5
Configurations are saved on partition: shared
configured ethernet- 1 ip
```

```
configured ethernet- 2 ip
Configuring slb server for site: i-009c612b4d8b956c1
Successfully Configured slb server for site: i-009c612b4d8b956c1
Configuring service group for site: i-009c612b4d8b956c1
Successfully Configured service group for site:i-009c612b4d8b956c1
Successfully Configured virtual server for site: i-009c612b4d8b956c1
Successfully Configured gslb site: i-009c612b4d8b956c1
Successfully Configured default route:i-009c612b4d8b956c1
Configurations are saved on partition: shared
configured ethernet- 1 ip
configured ethernet- 2 ip
Configuring slb server for site: i-0037786b1d931ed5a
Successfully Configured slb server for site: i-0037786b1d931ed5a
Configuring service group for site: i-0037786b1d931ed5a
Successfully Configured service group for site:i-0037786b1d931ed5a
Successfully Configured virtual server for site: i-0037786b1d931ed5a
Successfully Configured gslb site: i-0037786b1d931ed5a
Successfully Configured default route:i-0037786b1d931ed5a
Configurations are saved on partition: shared
configuring controller devices
configured ethernet- 1 ip
configured ethernet- 2 ip
Successfully Configured gslb server for controller: i-071cdac9b18de2832
Successfully Configured ServiceIp for site: i-071cdac9b18de2832
Successfully Configured ServiceIp for site: i-071cdac9b18de2832
Successfully Configured ServiceIp for site: i-071cdac9b18de2832
Successfully Configured ServiceIp for site: i-071cdac9b18de2832
Successfully Configured site information for: i-071cdac9b18de2832
Successfully Configured site information for: i-071cdac9b18de2832
Successfully Configured site information for: i-071cdac9b18de2832
Successfully Configured site information for: i-071cdac9b18de2832
Successfully Configured gslb policy for :i-071cdac9b18de2832
Successfully Configured gslb zone for :i-071cdac9b18de2832
Successfully Configured gslb controller and status interval: i-
071cdac9b18de2832
Successfully Configured gslb controller group: i-071cdac9b18de2832
Successfully Configured geo location: i-071cdac9b18de2832
```

```
Successfully Configured default route:i-071cdac9b18de2832
Configurations are saved on partition: shared
configured ethernet- 1 ip
configured ethernet- 2 ip
Successfully Configured gslb server for controller: i-0d8a97008c3cddb9d
Successfully Configured gslb controller group: i-0d8a97008c3cddb9d
Successfully Configured default route:i-0d8a97008c3cddb9d
Configurations are saved on partition: shared
```

3. Verify the following for each site devices:
   - Interfaces are enabled
   - SLB is configured
   - Site device is enabled to be a GSLB device.
   - Default route is configured pointing to the client-side data interface for traffic to exit the vThunder.
4. Verify the following for each GSLB controller:
   - Interfaces are enabled
   - vThunder device is configured with the required GSLB configuration
   - Geo location is enabled.
   - Default route is configured pointing to the client-side data interface for traffic to exit the vThunder.

Verify Deployment

To verify deployment using CFT, perform the following steps:

1. Verify SLB configuration on the following vThunder instances:

   CONTROLLER - Master configuration

   Run the following command:

```
vThunder-gslb:Master(config)(NOLICENSE)#show running-config
```

If the deployment is successful, the following controller and site configuration is displayed on vThunder master controller:

```
!Current configuration: 246 bytes
!Configuration last updated at 11:58:47 GMT Mon Jan 9 2023
!Configuration last saved at 11:58:51 GMT Mon Jan 9 2023
!64-bit Advanced Core OS (ACOS) version 5.2.1, build 153 (Dec-11-
2020,16:36)
!
!
no system geo-location load iana
system geo-location load GeoLite2-City
!
!
interface ethernet 1
  enable
  ip address dhcp
!
interface ethernet 2
  enable
  ip address dhcp
!
!
ip route 0.0.0.0 /0 10.1.2.1
ip route 0.0.0.0 /0 10.1.1.1
!
slb virtual-server gslb-server 10.1.2.121
  port 53 udp
    gslb-enable
!
gslb service-ip vs1 10.1.2.123
  external-ip 35.153.250.242
  port 80 tcp
!
gslb service-ip vs2 10.1.2.124
  external-ip 44.208.102.77
```

```
  port 80 tcp
!
gslb service-ip vs3 10.2.2.123
  external-ip 18.188.27.110
  port 80 tcp
!
gslb service-ip vs4 10.2.2.124
  external-ip 3.14.157.128
  port 80 tcp
!
gslb group default
  enable
  priority 255
!
gslb site eastus_1
  geo-location "North America,United States"
  slb-dev slb1 107.21.185.247
    vip-server vs1
!
gslb site eastus_2
  geo-location "North America,United States"
  slb-dev slb2 3.220.107.48
    vip-server vs2
!
gslb site eastus2_1
  geo-location "North America.United States.California.San Jose"
  slb-dev slb3 18.116.22.194
    vip-server vs3
!
gslb site eastus2_2
  geo-location "North America.United States.California.San Jose"
  slb-dev slb4 3.134.234.243
    vip-server vs4
!
gslb policy a10
  metric-order health-check geographic
```

```
  dns server authoritative
!
gslb zone gslb.a10.com
  policy a10
  service 80 www
    dns-a-record vs1 static
    dns-a-record vs2 static
    dns-a-record vs3 static
    dns-a-record vs4 static
!
gslb protocol status-interval 1
!
gslb protocol enable controller
!
!
end
!Current config commit point for partition 0 is 0 & config mode is
classical-mode
```

CONTROLLER - Member configuration

Run the following command:

```
vThunder-gslb:Member(config)(NOLICENSE)#show running-config
```

If the deployment is successful, the following controller and site configuration is displayed on vThunder member controller:

```
!Current configuration: 182 bytes
!Configuration last updated at 11:59:07 GMT Mon Jan 9 2023
!Configuration last saved at 11:59:05 GMT Mon Jan 9 2023
!64-bit Advanced Core OS (ACOS) version 5.2.1, build 153 (Dec-11-
2020,16:36)
!
!
interface ethernet 1
  enable
  ip address dhcp
!
```

```
interface ethernet 2
  enable
  ip address dhcp
!
!
ip route 0.0.0.0 /0 10.2.2.1
ip route 0.0.0.0 /0 10.2.1.1
!
slb virtual-server gslb-server 10.2.2.121
  port 53 udp
    gslb-enable
!
gslb service-ip vs1 10.1.2.123
  external-ip 35.153.250.242
  port 80 tcp
!
gslb service-ip vs2 10.1.2.124
  external-ip 44.208.102.77
  port 80 tcp
!
gslb service-ip vs3 10.2.2.123
  external-ip 18.188.27.110
  port 80 tcp
!
gslb service-ip vs4 10.2.2.124
  external-ip 3.14.157.128
  port 80 tcp
!
gslb group default
  enable
  primary 18.209.143.246
!
gslb site eastus_1
  geo-location "North America,United States"
  slb-dev slb1 107.21.185.247
    vip-server vs1
```

```
!
gslb site eastus_2
  geo-location "North America,United States"
  slb-dev slb2 3.220.107.48
    vip-server vs2
!
gslb site eastus2_1
  geo-location "North America.United States.California.San Jose"
  slb-dev slb3 18.116.22.194
    vip-server vs3
!
gslb site eastus2_2
  geo-location "North America.United States.California.San Jose"
  slb-dev slb4 3.134.234.243
    vip-server vs4
!
gslb policy a10
  metric-order health-check geographic
  dns server authoritative
!
gslb zone gslb.a10.com
  policy a10
  service 80 www
    dns-a-record vs1 static
    dns-a-record vs2 static
    dns-a-record vs3 static
    dns-a-record vs4 static
!
gslb protocol status-interval 1
!
gslb protocol enable controller
!
!
end
!Current config commit point for partition 0 is 0 & config mode is
classical-mode
```

SITE1 REGION1 configuration

Run the following command:

```
vThunder(config)(NOLICENSE)#show running-config
```

If the deployment is successful, the following controller and site configuration is displayed on vThunder site1 region1:

```
!Current configuration: 89 bytes
!Configuration last updated at 11:57:22 GMT Mon Jan 9 2023
!Configuration last saved at 11:57:26 GMT Mon Jan 9 2023
!64-bit Advanced Core OS (ACOS) version 5.2.1, build 153 (Dec-11-
2020,16:36)
!
!
interface ethernet 1
  enable
  ip address dhcp
!
interface ethernet 2
  enable
  ip address dhcp
!
!
ip route 0.0.0.0 /0 10.1.2.1
ip route 0.0.0.0 /0 10.1.1.1
!
slb server vth-stack1-server1 10.1.3.52
  health-check-disable
  port 80 tcp
    health-check-disable
!
slb service-group sg tcp
  member vth-stack1-server1 80
!
slb virtual-server vs1 10.1.2.123
  port 80 tcp
```

```
    source-nat auto
    service-group sg
!
!
gslb protocol enable device
!
!
end
!Current config commit point for partition 0 is 0 & config mode is
classical-mode
```

SITE2 REGION1 configuration

Run the following command:

```
vThunder(config)(NOLICENSE)#show running-config
```

If the deployment is successful, the following controller and site configuration is displayed on vThunder site2 region1:

```
!Current configuration: 89 bytes
!Configuration last updated at 11:57:39 GMT Mon Jan 9 2023
!Configuration last saved at 11:57:43 GMT Mon Jan 9 2023
!64-bit Advanced Core OS (ACOS) version 5.2.1, build 153 (Dec-11-
2020,16:36)
!
!
interface ethernet 1
  enable
  ip address dhcp
!
interface ethernet 2
  enable
  ip address dhcp
!
!
ip route 0.0.0.0 /0 10.1.2.1
ip route 0.0.0.0 /0 10.1.1.1
!
```

```
slb server vth-stack1-server2 10.1.3.250
  health-check-disable
  port 80 tcp
    health-check-disable
!
slb service-group sg tcp
  member vth-stack1-server2 80
!
slb virtual-server vs1 10.1.2.124
  port 80 tcp
    source-nat auto
    service-group sg
!
!
gslb protocol enable device
!
!
end
!Current config commit point for partition 0 is 0 & config mode is
classical-mode
```

SITE1 REGION2 configuration

Run the following command:

```
vThunder(config)(NOLICENSE)#show running-config
```

If the deployment is successful, the following controller and site configuration is displayed on vThunder site1 region2 :

```
!Current configuration: 89 bytes
!Configuration last updated at 11:57:55 GMT Mon Jan 9 2023
!Configuration last saved at 11:57:59 GMT Mon Jan 9 2023
!64-bit Advanced Core OS (ACOS) version 5.2.1, build 153 (Dec-11-
2020,16:36)
!
!
interface ethernet 1
  enable
```

```
  ip address dhcp
!
interface ethernet 2
  enable
  ip address dhcp
!
!
ip route 0.0.0.0 /0 10.2.2.1
ip route 0.0.0.0 /0 10.2.1.1
!
slb server vth-stack2-server3 10.2.3.179
  health-check-disable
  port 80 tcp
    health-check-disable
!
slb service-group sg tcp
  member vth-stack2-server3 80
!
slb virtual-server vs1 10.2.2.123
  port 80 tcp
    source-nat auto
    service-group sg
!
!
gslb protocol enable device
!
!
end
!Current config commit point for partition 0 is 0 & config mode is
classical-mode
```

SITE2 REGION2 configuration

Run the following command:

```
vThunder(config)(NOLICENSE)#show running-config
```

If the deployment is successful, the following controller and site configuration is displayed on vThunder site2 region2 :

```
!Current configuration: 89 bytes
!Configuration last updated at 11:58:10 GMT Mon Jan 9 2023
!Configuration last saved at 11:58:14 GMT Mon Jan 9 2023
!64-bit Advanced Core OS (ACOS) version 5.2.1, build 153 (Dec-11-
2020,16:36)
!
!
interface ethernet 1
  enable
  ip address dhcp
!
interface ethernet 2
  enable
  ip address dhcp
!
!
ip route 0.0.0.0 /0 10.2.2.1
ip route 0.0.0.0 /0 10.2.1.1
!
slb server vth-stack2-server4 10.2.3.254
  health-check-disable
  port 80 tcp
    health-check-disable
!
slb service-group sg tcp
  member vth-stack2-server4 80
!
slb virtual-server vs1 10.2.2.124
  port 80 tcp
    source-nat auto
    service-group sg
!
!
```

```
gslb protocol enable device
!
!
end
!Current config commit point for partition 0 is 0 & config mode is
classical-mode
```

2. Verify GSLB group information on the following vThunder instances:

CONTROLLER - Master configuration

a. Run the following command:

```
vThunder-gslb:Master(NOLICENSE)#show gslb group
```

b. Verify if the public ip of member controller is displayed in the response:

```
                      Pri = Priority, Attrs = Attributes
                      S-Cfg = Secure Config
                      S-State = Secure Status
                          D = Disabled, L = Learn
                          P = Passive, * = Master
                          E = Enabled, EF = Enable-Fallback
                          Unsec = Unsecure, Unkwn = Unknown
                          Estng = Establishing, Estd = Established
Group: default, Master: local
Member                          Sys-ID    Pri Attrs  Status      S-
Cfg S-State Address
-------------------------------------------------------------------
--------------
local                           7b8a5001 255 L*     OK
vThunder                        58dd5c28 100 PL     Synced      D
   Unsec   13.58.227.170
```

CONTROLLER - Member configuration

a. Run the following command:

```
vThunder-gslb:Member(NOLICENSE)#show gslb group
```

b. Verify if the public ip of master controller is displayed in the response:

```
                      Pri = Priority, Attrs = Attributes
                      S-Cfg = Secure Config
```

```
                      S-State = Secure Status
                          D = Disabled, L = Learn
                          P = Passive, * = Master
                          E = Enabled, EF = Enable-Fallback
                          Unsec = Unsecure, Unkwn = Unknown
                          Estng = Establishing, Estd = Established
Group: default, Master: vThunder
Member                            Sys-ID   Pri Attrs  Status      S-
Cfg S-State Address
-------------------------------------------------------------------
--------------
local                             58dd5c28 100 L      OK
vThunder                          7b8a5001 255 L*     Synced      D
   Unsec   18.209.143.246
```

3. Verify the GSLB protocol information on the following vThunder instances:

CONTROLLER - Master configuration

Run the following command:

```
vThunder-gslb:Master(NOLICENSE)#show gslb protocol
```

The following configuration is displayed on vThunder master controller:

```
GSLB site: eastus_1
  SLB device: slb1 (10.1.1.170:10578) Established
 Session ID:    7629
 Secure Config:                      Disable |Current SSL State:
              Unsecure
 Connection succeeded:                     1 |Connection failed:
                     0
 Open packet sent:                         1 |Open packet received:
                     1
 Open session succeeded:                   1 |Open session failed:
                     0
 Sessions Dropped:                         0 |Update packet received:
                  1320
 Keepalive packet sent:                   23 |Keepalive packet
received:                   22
```

```
 Notify packet sent:                          0 |Notify packet received:
                      0
 Message Header Error:                        0 |Protocol RDT(ms):
                      0
 GSLB Protocol Version:                       2 |Peer ACOS Version:
        5.2.1 Build 153
 Secure negotiation Success:                  0 |Secure negotiation
Failures:                    0
 SSL handshake Success:                       0 |SSL handshake Failures:
                      0

GSLB site: eastus_2
  SLB device: slb2 (10.1.1.170:4776) Established
 Session ID:    20481
 Secure Config:                        Disable |Current SSL State:
               Unsecure
 Connection succeeded:                        1 |Connection failed:
                      0
 Open packet sent:                            1 |Open packet received:
                      1
 Open session succeeded:                      1 |Open session failed:
                      0
 Sessions Dropped:                            0 |Update packet received:
                   1320
 Keepalive packet sent:                      23 |Keepalive packet
received:                     22
 Notify packet sent:                          0 |Notify packet received:
                      0
 Message Header Error:                        0 |Protocol RDT(ms):
                      0
 GSLB Protocol Version:                       2 |Peer ACOS Version:
        5.2.1 Build 153
 Secure negotiation Success:                  0 |Secure negotiation
Failures:                    0
 SSL handshake Success:                       0 |SSL handshake Failures:
                      0
```

```
GSLB site: eastus2_1
  SLB device: slb3 (10.1.1.170:3352) Established
 Session ID:    25287
 Secure Config:                      Disable |Current SSL State:
               Unsecure
 Connection succeeded:                     1 |Connection failed:
                      0
 Open packet sent:                         1 |Open packet received:
                      1
 Open session succeeded:                   1 |Open session failed:
                      0
 Sessions Dropped:                         0 |Update packet received:
                   1320
 Keepalive packet sent:                   23 |Keepalive packet
received:                    22
 Notify packet sent:                       0 |Notify packet received:
                      0
 Message Header Error:                     0 |Protocol RDT(ms):
                     12
 GSLB Protocol Version:                    2 |Peer ACOS Version:
        5.2.1 Build 153
 Secure negotiation Success:               0 |Secure negotiation
Failures:                    0
 SSL handshake Success:                    0 |SSL handshake Failures:
                      0

GSLB site: eastus2_2
  SLB device: slb4 (10.1.1.170:19222) Established
 Session ID:    6077
 Secure Config:                      Disable |Current SSL State:
               Unsecure
 Connection succeeded:                     1 |Connection failed:
                      0
 Open packet sent:                         1 |Open packet received:
                      1
```

```
 Open session succeeded:                    1 |Open session failed:
                       0
 Sessions Dropped:                          0 |Update packet received:
                    1406
 Keepalive packet sent:                    24 |Keepalive packet
received:                     23
 Notify packet sent:                        0 |Notify packet received:
                       0
 Message Header Error:                      0 |Protocol RDT(ms):
                      12
 GSLB Protocol Version:                     2 |Peer ACOS Version:
        5.2.1 Build 153
 Secure negotiation Success:                0 |Secure negotiation
Failures:                     0
 SSL handshake Success:                     0 |SSL handshake Failures:
                       0

GSLB protocol is disabled for site devices.
```

CONTROLLER - Member configuration

Run the following command:

```
vThunder-gslb:Member(NOLICENSE)#show gslb protocol
```

The following configuration is displayed on vThunder member controller:

```
GSLB site: eastus_1
  SLB device: slb1 (0.0.0.0:0) GroupControl
 Session ID:    Not Available
 Secure Config:                          None |Current SSL State:
                    None
 Connection succeeded:                      0 |Connection failed:
                       0
 Open packet sent:                          0 |Open packet received:
                       0
 Open session succeeded:                    0 |Open session failed:
                       0
 Sessions Dropped:                          0 |Update packet received:
```

```
                      0
 Keepalive packet sent:                    0 |Keepalive packet
received:                      0
 Notify packet sent:                       0 |Notify packet received:
                      0
 Message Header Error:                     0 |Protocol RDT(ms):
                      0
 GSLB Protocol Version:                    2
 Secure negotiation Success:               0 |Secure negotiation
Failures:                      0
 SSL handshake Success:                    0 |SSL handshake Failures:
                      0

GSLB site: eastus_2
  SLB device: slb2 (0.0.0.0:0) GroupControl
 Session ID:    Not Available
 Secure Config:                          None |Current SSL State:
                   None
 Connection succeeded:                     0 |Connection failed:
                      0
 Open packet sent:                         0 |Open packet received:
                      0
 Open session succeeded:                   0 |Open session failed:
                      0
 Sessions Dropped:                         0 |Update packet received:
                      0
 Keepalive packet sent:                    0 |Keepalive packet
received:                      0
 Notify packet sent:                       0 |Notify packet received:
                      0
 Message Header Error:                     0 |Protocol RDT(ms):
                      0
 GSLB Protocol Version:                    2
 Secure negotiation Success:               0 |Secure negotiation
Failures:                      0
 SSL handshake Success:                    0 |SSL handshake Failures:
```

```
                      0

GSLB site: eastus2_1
  SLB device: slb3 (0.0.0.0:0) GroupControl
 Session ID:    Not Available
 Secure Config:                          None |Current SSL State:
                   None
 Connection succeeded:                      0 |Connection failed:
                      0
 Open packet sent:                          0 |Open packet received:
                      0
 Open session succeeded:                    0 |Open session failed:
                      0
 Sessions Dropped:                          0 |Update packet received:
                      0
 Keepalive packet sent:                     0 |Keepalive packet
received:                     0
 Notify packet sent:                        0 |Notify packet received:
                      0
 Message Header Error:                      0 |Protocol RDT(ms):
                      0
 GSLB Protocol Version:                     2
 Secure negotiation Success:                0 |Secure negotiation
Failures:                    0
 SSL handshake Success:                     0 |SSL handshake Failures:
                      0

GSLB site: eastus2_2
  SLB device: slb4 (0.0.0.0:0) GroupControl
 Session ID:    Not Available
 Secure Config:                          None |Current SSL State:
                   None
 Connection succeeded:                      0 |Connection failed:
                      0
 Open packet sent:                          0 |Open packet received:
                      0
```

```
 Open session succeeded:                    0 |Open session failed:
                        0
 Sessions Dropped:                          0 |Update packet received:
                        0
 Keepalive packet sent:                     0 |Keepalive packet
received:                      0
 Notify packet sent:                        0 |Notify packet received:
                        0
 Message Header Error:                      0 |Protocol RDT(ms):
                        0
 GSLB Protocol Version:                     2
 Secure negotiation Success:                0 |Secure negotiation
Failures:                      0
 SSL handshake Success:                     0 |SSL handshake Failures:
                        0

GSLB protocol is disabled for site devices.
```

Verify Traffic Flow

The traffic flow can be tested using the following:

- DNS Lookup
- WGET

DNS Lookup

To verify the traffic flow from via vThunder, perform the following:

1. Perform a DNS lookup on server1 of region1 using the master controller's client-side data interface public IP in the following command:

   ```
   $ dig @master_controller_data_public_IP www.gslb.a10.com
   ```

   The master controller's client-side data interface public IP is used as DNS server IP. You can get this data interface public IP from **AWS Management Console** >

EC2 > **Instances** > <stack_name_master_controller_region1> > **Networking** > **Elastic IP address**.

Figure 80 : Master Controller Data Interface Public IP

The following response is received:

```
$ dig @34.202.82.33  www.gslb.a10.com
; <<>> DiG 9.18.1-Ubuntu <<>> @34.202.82.33 www.gslb.a10.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 62463
;; flags: qr aa rd; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1400
;; QUESTION SECTION:
;www.gslb.a10.com.              IN      A

;; ANSWER SECTION:
www.gslb.a10.com.       10      IN      A       34.199.187.33
www.gslb.a10.com.       10      IN      A       3.232.227.57
www.gslb.a10.com.       10      IN      A       3.140.125.3
www.gslb.a10.com.       10      IN      A       3.16.234.54

;; Query time: 0 msec
;; SERVER: 34.202.82.33#53(34.202.82.33)(UDP)
;; WHEN: Mon Jan 09 17:43:25 IST 2023
;; MSG SIZE  rcvd: 125
```

2. Perform the DNS lookup again.

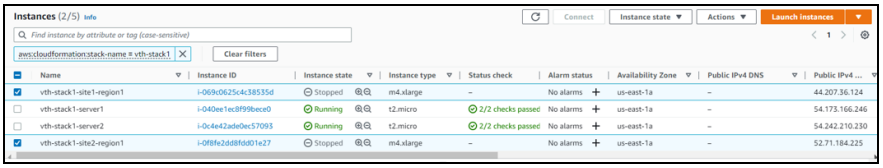
```
$ dig @34.202.82.33  www.gslb.a10.com
; <<>> DiG 9.18.1-Ubuntu <<>> @34.202.82.33 www.gslb.a10.com
```

```
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 62463
;; flags: qr aa rd; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1400
;; QUESTION SECTION:
;www.gslb.a10.com.              IN      A

;; ANSWER SECTION:
www.gslb.a10.com.       10      IN      A       3.232.227.57
www.gslb.a10.com.       10      IN      A       34.199.187.33
www.gslb.a10.com.       10      IN      A       3.140.125.3
www.gslb.a10.com.       10      IN      A       3.16.234.54

;; Query time: 0 msec
;; SERVER: 34.202.82.33#53(34.202.82.33) (UDP)
;; WHEN: Mon Jan 09 17:44:25 IST 2023
;; MSG SIZE  rcvd: 125
```

The response is received with shuffled server IP addresses.

3. Stop the site instances of region1.

Figure 81 : Stopped Site instances



4. Perform the DNS lookup to verify if you receive a response after stopping the site instances.

```
$ dig @34.202.82.33  www.gslb.a10.com
; <<>> DiG 9.18.1-Ubuntu <<>> @34.202.82.33 www.gslb.a10.com
; (1 server found)
;; global options: +cmd
```

```
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 62463
;; flags: qr aa rd; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1400
;; QUESTION SECTION:
;www.gslb.a10.com.              IN      A

;; ANSWER SECTION:
www.gslb.a10.com.       10      IN      A       3.140.125.3
www.gslb.a10.com.       10      IN      A       3.16.234.54
www.gslb.a10.com.       10      IN      A       3.232.227.57
www.gslb.a10.com.       10      IN      A       34.199.187.33

;; Query time: 0 msec
;; SERVER: 34.202.82.33#53(34.202.82.33) (UDP)
;; WHEN: Mon Jan 09 17:46:25 IST 2023
;; MSG SIZE  rcvd: 125
```

The response is received with site devices secondary data1 public IPs based on round robin.

WGET

To verify the traffic flow via vThunder, perform the following:

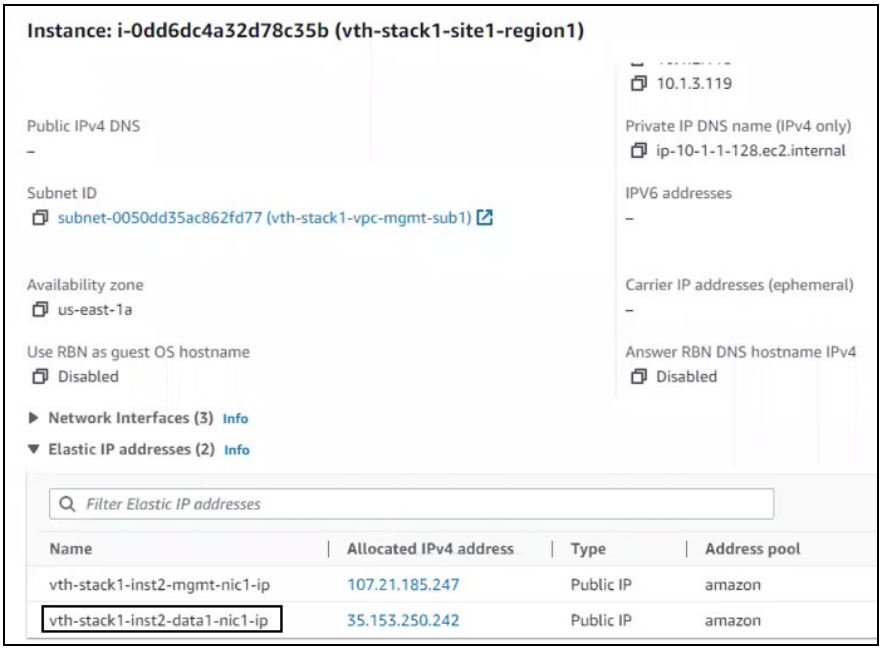
1. From **AWS Management Console**, navigate to **EC2** > **Instances**.
2. Select any one of the server instance assigned to the site devices.
   Here, **`vth-stack1-server1`** is the server instance name.
3. Click **Connect**.
   A **Connect to instance** window with **EC2 Instance Connect** tab is displayed.
4. Click **Connect**.
   A **Terminal** window is displayed.
5. Run the following command in the Terminal window to create an Apache Server virtual machine:

```
$ sudo apt install apache2
```

While the Apache server is getting installed, you get a prompt to continue further. Enter 'Y' to continue. After the installation is complete, a newline prompt is displayed.

6. From **AWS Management Console** > **EC2** > **Instances**, select the site instance of the corresponding server on which Apache was installed.
   Here, **`vth-stack1-site1-region1`** is the site instance.
7. Navigate to **Networking** tab > **Elastic IP addresses** and copy the **Allocated IPv4 address** of site instance data interface.

   Figure 82 : Site Instance Data Interface Public IP



8. Run the following command on the server1 of region1:

```
$ wget site_device_data-public-IP
```

The following response is received:

```
$ wget 35.153.250.242
--2023-01-09 17:49:47--  http://35.153.250.242/
Connecting to 35.153.250.242:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10671 (10K) [text/html]
```

```
Saving to: 'index.html.3'

index.html.3                                        100%
[=============================================================================
=========]  10.42K  --.-KB/s    in 0s

                      2023-01-09 17:49:47 (63.8 MB/s) - 'index.html.3' saved [1067
```

Appendix

The following topics are covered:

Security Policy for AWS User

To deploy the vThunder instance using a CFT template, an AWS user requires certain security policies. The following security policies are recommended:

Predefined

- AmazonEC2FullAccess
- AmazonS3FullAccess
- AmazonS3ObjectLambdaExecutionRolePolicy
- AmazonVPCFullAccess

Custom

- Create and Edit Secrets

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:CreateSecret",
                "secretsmanager:PutSecretValue"
            ],
            "Resource": "arn:aws:secretsmanager:us-east-
1:939850196882:secret:*"
        }
```

```
    ]
}
```

- Lambda Update

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ConfigureFunctions",
            "Effect": "Allow",
            "Action": [
                "lambda:UpdateFunctionConfiguration",
                "lambda:GetFunction"
            ],
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringLike": {
                    "lambda:Layer": [
                        "arn:aws:lambda:*:939850196882:layer:*:*"
                    ]
                }
            }
        }
    ]
}
```

- Manage Secrets

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetResourcePolicy",
                "secretsmanager:GetSecretValue",
                "secretsmanager:DescribeSecret",
                "secretsmanager:ListSecretVersionIds"
            ],
            "Resource": "arn:aws:secretsmanager:us-east-
```

```
1:939850196882:secret:*"
        },
        {
            "Effect": "Allow",
            "Action": "secretsmanager:ListSecrets",
            "Resource": "*"
        }
    ]
}
```

- Cloud Watch Logs and Streams

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetResourcePolicy",
                "secretsmanager:GetSecretValue",
                "secretsmanager:DescribeSecret",
                "secretsmanager:ListSecretVersionIds"
            ],
            "Resource": "arn:aws:secretsmanager:us-east-
1:939850196882:secret:*"
        },
        {
            "Effect": "Allow",
            "Action": "secretsmanager:ListSecrets",
            "Resource": "*"
        }
    ]
}
```

Contact Us