Installing vThunder ADC using AWS CFT Templates

v1.3.0

September, 2024

© 2024 A10 Networks, Inc. All rights reserved.

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

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 **6**

Terminology 7

Prerequisites 8

Image Repository 9

Deployment Templates **11**

Thunder-2NIC-1VM 22

Create Thunder Virtual Machine 23

Configure Thunder 28

Verify Deployment 29

Verify Traffic Flow 33

Thunder-3NIC-2VM 37

Create Thunder Virtual Machines for Same Availability Zone 41

Configure Thunder 48

Verify Deployment 49

Verify Traffic Flow 60

Create Thunder Virtual Machines for Different Availability Zones 63

Configure Thunder 74

Verify Deployment 74

Verify Traffic Flow 81

Thunder-3NIC-3VM 85

Create Thunder Virtual Machines 87

Configure Thunder 98

Verify Deployment 99

Verify Traffic Flow 121

ADC Configuration Templates **128**

Change Password 132

Basic Server Load Balancer 134

Backend Server Autoscaling 144

Create S3 Bucket 144
Create AWS Secrets Manager secret 146
Create vThunder credentials secret 146
Create Lambda Function and Autoscaling Group 147
Configure Lambda Function and ASG 153
SSL Certificate 155
A10 License 157
High Availability 158
High Availability Across Availability Zones 162
Hybrid Cloud GSLB 174
Architectural References 174
GSLB Deployment Topology 176
Configure Hybrid Cloud GSLB 177
Appendix 199
Access Thunder Virtual Machine 199
Create Server Machine 202
Create Client Machine 204
Create Virtual Private Cloud 206
Create Virtual Private Cloud for Different Availability Zones 210
Create Elastic Public IP 215
Default Password Policy 218
Configure the FTP server 219
Create FTP server 219
Configure FTP server 220
Delete the resources 221
Install Python3 222
Verify Installation 222
Install Python dependencies 223
List of ACOS AMI ID 224
Security Policy for AWS User 236

Predefined 236
Custom 237
Supported Instance Types 240
License Information 243
Support Information 244
What's New 245
1.3.0 245
1.2.0 245
1.1.0 246
1.0.0 246

Introduction

The A10 Thunder® Application Delivery Controller (ADC) is a high-performance solution designed to accelerate and optimize critical applications, ensuring their reliable and efficient delivery.

This document assists you in deploying Thunder® ADC instances on AWS Cloud using CloudFormation Templates (CFT).

The following steps provide a high-level overview of the deployment process:

1. Provision the AWS Cloud network infrastructure.

   There are custom templates available for creating new virtual private cloud for same or different availability zone, network security groups (NSGs), subnets, and elastic public IP to provision the new infrastructure. The creation of these resources is optional; if the resources already exist, they can be reused.

   For more information, see Create Elastic Public IP, Create Virtual Private Cloud, and Create Virtual Private Cloud for Different Availability Zones.

   For more information on other prerequisites, see Prerequisites.

2. Create Thunder virtual machine/s on the AWS Cloud.

   There are custom templates available for creating virtual machines (VMs) along with network interface card (NIC) settings on AWS Cloud with built-in Thunder.

   For more information on the various deployment templates, see Deployment Templates.

   You can deploy Thunder on the AWS Cloud using AWS Management Console.

3. Configure Thunder.

   There are custom Python scripts available to apply the new Thunder configurations. Different scripts are available for various configuration needs.

   For more information, see ADC Configuration Templates.

Terminology

The following is a glossary of AWS terms commonly used in this document:

- **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
- **AWS Management Console** — A web console to create and monitor AWS resources. For more information, see https://aws.amazon.com/console/
- **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.
- **Autoscaling Group (ASG) —** A feature that enables automatic scaling of the number of instances (virtual servers or machines) in response to varying demand for applications or services.
- **AWS Access Keys** — The credentials for an IAM user or the AWS account root user.
- **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.
- **Global Server Load Balancing (GSLB)** — A process to distribute incoming network traffic across multiple servers or data centers located in different geographical locations.
- **Health Probe** — A health probe is used to determine the health status of the virtual machine instances in the backend pool.
- **High Availability (HA)** — A capability to remain operational and accessible for a significantly high percentage of the time.

- **Hybrid Cloud** — A cloud computing model that combines private cloud and public cloud services within the same seamless infrastructure.
- **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
- **Network security group (NSG)** — A network security group (NSG) contains a list of security rules that allow or deny network traffic to resources connected to AWS virtual networks (VNet). The NSGs can be associated with subnets or individual NICs attached to the VMs. When an NSG is associated with a subnet, the rules apply to all the resources connected to the subnet.
- **Python3** — The latest major version of the Python programming language.
- **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 that you can 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
- **vThunder** — An A10 Thunder instance for virtual machine.

Prerequisites

To create and configure Thunder virtual machine on the AWS cloud using CFT templates, you must ensure that the following prerequisites are met:

1. Download A10 CFT Templates from GitHub.
2. Ensure that you have an AWS account with sufficient permissible role. For more information, see Security Policy for AWS User.
3. Access AWS Management Console to create Thunder virtual machine using CFT templates.

4. Sign up here to get Thunder Trial license.
5. Install Python on your machine to execute the ADC Thunder configuration scripts. For more information, see Install Python3.
6. 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
7. 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

For any queries, reach out to A10 Networks Support.

Image Repository

Table 1 provides the list of ACOS versions and modules that support the CFT templates:

Table 1 : Supported ACOS versions

ACOS Version	ADC	CGN	SSLi	TPS
64-bit Advanced Core OS (ACOS) version 6.0.4	√	X	X	X
64-bit Advanced Core OS (ACOS) version 6.0.3	√	X	X	X
64-bit Advanced Core OS (ACOS) version 6.0.2	√	X	X	X
64-bit Advanced Core OS (ACOS) version 6.0.1	√	X	X	X
64-bit Advanced Core OS	√	X	X	X

Table 1 : Supported ACOS versions

ACOS Version	ADC	CGN	SSLi	TPS
(ACOS) version 6.0.0-P2-SP1				
64-bit Advanced Core OS (ACOS) version 6.0.0-P1	√	X	X	X
64-bit Advanced Core OS (ACOS) version 5.2.1-P8	√	X	X	X
64-bit Advanced Core OS (ACOS) version 5.2.1-P7	√	X	X	X
64-bit Advanced Core OS (ACOS) version 5.2.1-P6	√	X	X	X

Deployment Templates

This section helps you in provisioning a new Thunder virtual machine on the AWS cloud.

Before proceeding, it is recommended to review the Prerequisites.

To provision a new virtual Thunder ADC instance on an AWS cloud, perform the following steps:

1. Create Virtual Private Cloud/Create Virtual Private Cloud for Different Availability Zones
   It is not mandatory to create new resources, the existing resources can be used in deployment and configuration.
2. Create Elastic Public IP.
   It is not mandatory to create new resource, the existing resource can be used in deployment and configuration. Some deployment templates require Elastic Public IP.
3. Select an appropriate template for deploying vThunder ADC on AWS cloud according to your use case.

   Table 2 provides a list of various use cases along with their respective supported CFT templates.

Table 2 : Supported CFT Templates

Use Case	Template Name	Number of Thunder/s	Number of NICs	Data-in NIC VIP	Description
Standalone Thunder ADC	Thunder-2NIC-1VM	1	2	Private	• Creates one vThunder instance with one management and one data NIC (data-in), see Figure 1.

Table 2 : Supported CFT Templates

Use Case	Template Name	Number of Thunder/s	Number of NICs	Data-in NIC VIP	Description
					• Applies additional configuration on vThunder as required: ◦ Change Password ◦ A10 License ◦ SSL Certificate ◦ Basic Server Load Balancer ◦ Backend Server Autoscaling
Thunder ADC in High Availability mode with Private/Public VIP	Thunder-3NIC-2VM	2	3	Private or Public	• Creates two vThunder instances with HA setup and each vThunder has one management and two data NICs (data-in and data-out), see Figure 6. • Configures data-in

Table 2 : Supported CFT Templates

Use Case	Template Name	Number of Thunder/s	Number of NICs	Data-in NIC VIP	Description
					network interface card (NIC) with Private/Public VIP. • When one instance becomes unavailable, another instance seamlessly handles the request without requiring manual intervention. • High availability can be configured only within the same availability zone in the same region. • Applies additional configuration on vThunder as required:

Table 2 : Supported CFT Templates

Use Case	Template Name	Number of Thunder/s	Number of NICs	Data-in NIC VIP	Description
					◦ Change Password ◦ A10 License ◦ SSL Certificate ◦ Basic Server Load Balancer ◦ High Availability
Thunder ADC in High Availability mode with Private/Public VIP and Backend Server Autoscale	Thunder-3NIC-2VM	2	3	Private or Public	• Creates two vThunder instances with HA setup and each vThunder has one management and two data NICs (data-in and data-out), see Figure 6. • Configures data-in network interface card (NIC) with Private/Public VIP. • Applies SLB

Table 2 : Supported CFT Templates

Use Case	Template Name	Number of Thunder/s	Number of NICs	Data-in NIC VIP	Description
					configuration using Lambda function for newly added/deleted web/app servers via autoscaling group. • When one instance becomes unavailable, another instance seamlessly handles the request without requiring manual intervention. • High availability can be configured only within the same availability zone in the same region. • Applies

Table 2 : Supported CFT Templates

Use Case	Template Name	Number of Thunder/s	Number of NICs	Data-in NIC VIP	Description
					additional configuration on vThunder as required: ◦ Change Password ◦ A10 License ◦ SSL Certificate ◦ Basic Server Load Balancer ◦ Backend Server Autoscaling ◦ High Availability
Thunder ADC in High Availability mode Across Availability Zones with Private/Public VIP	Thunder-3NIC-2VM	2	3	Private or Public	• Creates two vThunder instances with HA setup across different subnet or different availability zone and each vThunder has one management

Table 2 : Supported CFT Templates

Use Case	Template Name	Number of Thunder/s	Number of NICs	Data-in NIC VIP	Description
					and two data NICs (data-in and data-out), see Figure 6. • Configures data-in network interface card (NIC) with Private/Public VIP. • When one instance becomes unavailable, another instance seamlessly handles the request without requiring manual intervention. • High availability can be configured within the same availability

Table 2 : Supported CFT Templates

Use Case	Template Name	Number of Thunder/s	Number of NICs	Data-in NIC VIP	Description
					zone having six subnets for each interface or across different availability zones having 3 subnets each in the same region. • Applies additional configuration on vThunder as required: ◦ Change Password ◦ A10 License ◦ SSL Certificate ◦ Basic Server Load Balancer ◦ Backend Server Autoscaling ◦ High Availability Across Availability

Table 2 : Supported CFT Templates

Use Case	Template Name	Number of Thunder/s	Number of NICs	Data-in NIC VIP	Description
					Zones
Thunder ADC with GSLB (Disaster Recovery Site in a cross-region or hybrid cloud environment)	Thunder-3NIC-3VM	3	3	Public	• Creates three vThunder instances each vThunder has one management and two data NICs (data-in and data-out) in the same region1 and zone1, see Figure 27. These three vThunder instances are referred as Master Controller (Active), Site1 and Site2. • The identical set of vThunder resources should be deployed in region2 zone1 using the same template. The

Table 2 : Supported CFT Templates

Use Case	Template Name	Number of Thunder/s	Number of NICs	Data-in NIC VIP	Description
					three vThunder instances in region2 zone1 are referred as the Member Controller (Standby), Site1, and Site2. • When region1 experiences an outage, region2 seamlessly handles all requests through DNS switch over. • Applies additional configuration on vThunder as required: ◦ Change Password ◦ A10 License ◦ SSL Certificate ◦ Hybrid

Table 2 : Supported CFT Templates

Use Case	Template Name	Number of Thunder/s	Number of NICs	Data-in NIC VIP	Description
					Cloud GSLB

After completing the deployment process, proceed to configure your setup. For more information, see ADC Configuration Templates.

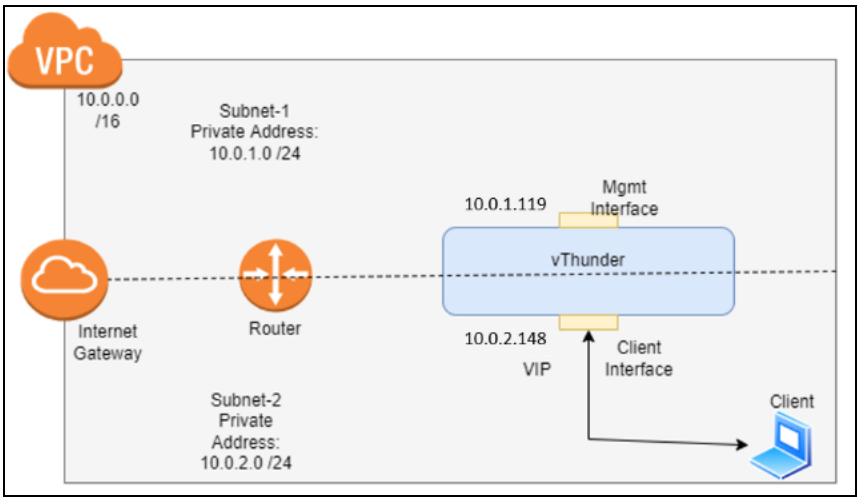
Thunder-2NIC-1VM

This template creates a new virtual machine with pre-loaded Thunder instance and creates two new network interface cards (NICs).

For more information, see Create Thunder Virtual Machine.

NOTE: Use a suitable VM size that supports at least two NICs. For VM sizes, see List of Supported Instance Types.

Figure 1 : Standalone Thunder ADC



Additional Thunder configurations are available that can be applied as needed:

- Change Password
- A10 License
- SSL Certificate
- Basic Server Load Balancer
- Backend Server Autoscaling

Various templates are available for different deployment needs.

For more information, see Deployment Templates.

The following topics are covered:

Create Thunder Virtual Machine 23
Configure Thunder 28
Verify Deployment 29
Verify Traffic Flow 33

Create Thunder Virtual Machine

The A10-vThunder-2NIC-1VM template is used to create a Thunder virtual machine with two network interface cards.

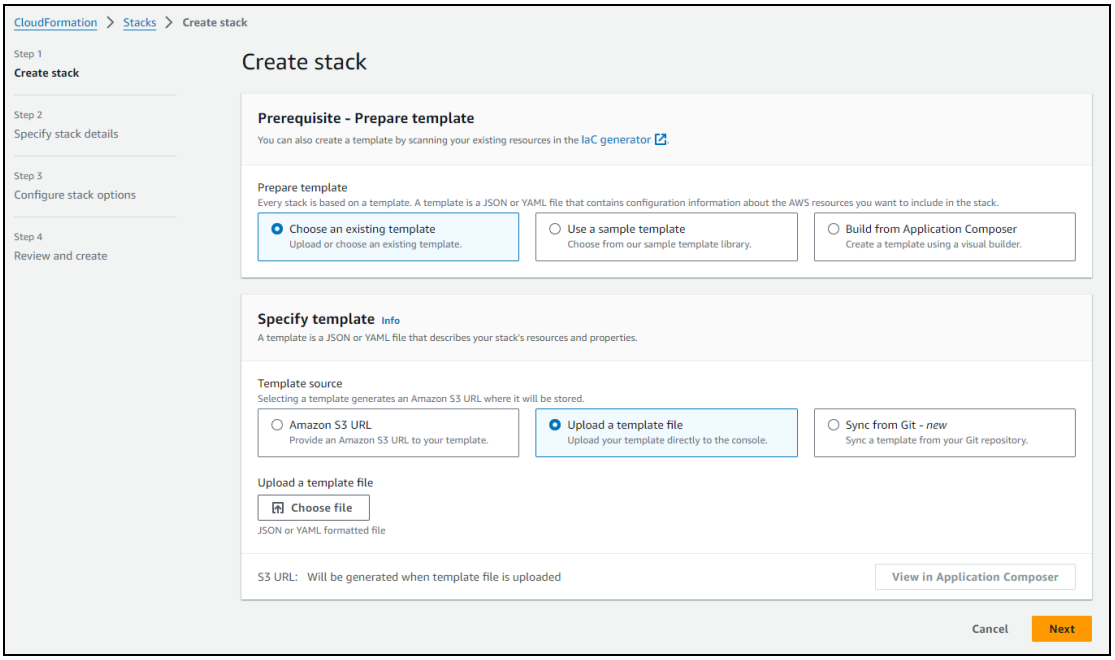
Before deploying this template, it is recommended to review the Prerequisites.

To create the Thunder virtual machines using A10-vThunder-2NIC-1VM template, perform the following steps:

1. Download **A10-vThunder_CFT-TEMPLATES** > **A10-vThunder-2NIC-1VM** template from GitHub.
2. Download A10-vThunder-2NIC-1VM template.
3. From AWS Management Console, navigate to **CloudFormation** > **Stacks** > **Create Stack** > **With new resources (standard)**.

   The Create stack window is displayed.

Figure 2 : Create stack window



4. In the **Prerequisite - Prepare template** section, select **Choose an existing template**.

   After selecting this option, the Specify template section is displayed.

5. 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.json

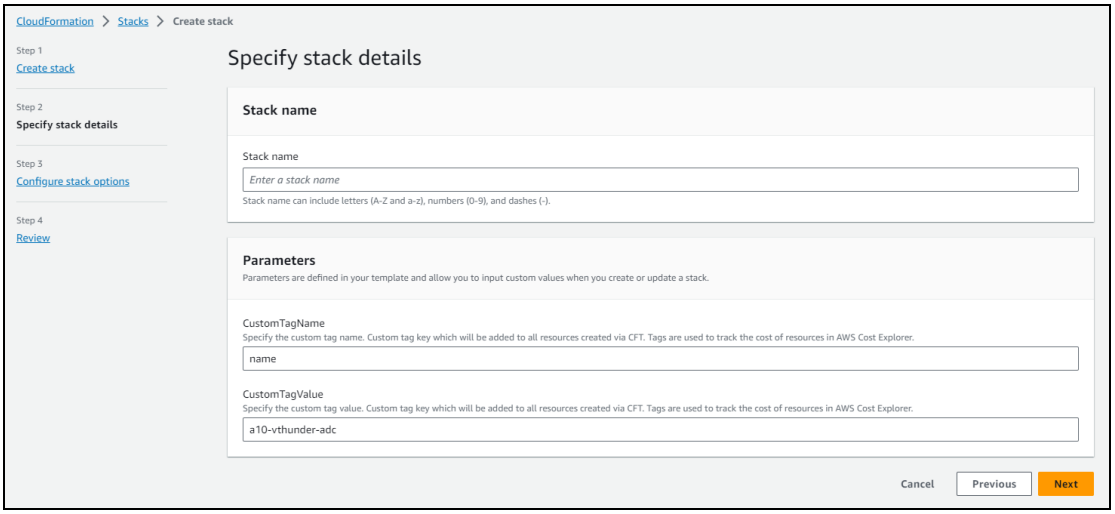
   The selected template file name is displayed as the chosen file.

NOTE: This template contains pre-populated default values that can be modified as required. It does not create a new Elastic IP, security groups, subnets, and Virtual Private Network.

6. Click **Next**.

   The **Specify stack details** window is displayed.

Figure 3 : Specify stack details window



7. In the **Specify stack details** window, enter or select the following:

Table 3 : JSON Parameters

Resource Name	Description
Stack name	Specify a stack name containing letters (A-Z and a-z), numbers (0-9), and dashes (-). Here, the stack name is provided as **`vth-2nic-1vm`**.
AMIID	Specify the AMI ID of the required A10 vThunder image. By default, A10 Thunder ADC for Advanced Load Balancing - BYOL AMI ID is set for us-east-1 region. To get the AMI ID, go to the **AWS Management Console** > **EC2** > **Launch instance** > **Application and OS Images (Amazon Machine Image)** > search for the relevant vThunder image and then copy the AMI ID. For more information on the available Thunder images with different capacities under respective regions, see List of ACOS AMI ID.
AllocationIdEIP	Specify an allocation ID of the required elastic IP.

Table 3 : JSON Parameters

Resource Name	Description
	To get the allocation ID, go to the **AWS Management Console** > **EC2** > **Elastic IPs** > <*elastic_IP_address*> > **Summary** and then copy the **Allocation ID**.
CustomTagName	Specify the custom tag name. The custom tag key is added to all the resources created using the CFT template. This tag is used to track the cost of resources in the AWS Cost Explorer.
CustomTagValue	Specify the custom tag value. The custom tag key is added to all the resources created using the CFT template. This tag is used to track the cost of resources in the AWS Cost Explorer. The default custom tag value is **a10-vthunder-adc**.
EC2Instance	Select an instance/compute type supported for vThunder from the available list. By default, instance type is m4.xlarge which contains 4 vCPU and 16 GiB memory. The minimum requirement for vThunder is 4 vCPU and 16 GiB memory. For more information on product pricing, see AWS Marketplace. If the required instance type is not available in the list, go to **AWS Management Console** > **EC2** > **Launch instance** > Find and copy the relevant instance type in the template. For more information, see Supported Instance Types.
KeyPair	Select a keypair from the available list. A key pair is a set of security credentials which consist of a public key and a private key. This keypair is used to verify your identity when connecting to an Amazon EC2 instance.
SecurityGroupData	Select the required security group ID from the available list for data interface within the selected

Table 3 : JSON Parameters

Resource Name	Description
	Virtual Private Network.
SecurityGroupMgmt	Select the required security group ID from the available list for the management interface within the selected Virtual Private Network.
SubnetDataID	Select the required subnet ID from the available list for the data traffic flow inward and outward to vThunder within the selected Virtual Private Network.
SubnetMgmtID	Select the required subnet ID from the available list for management interface of the vThunder within the selected Virtual Private Network.
Tenancy	Select the required tenancy from the available list. Tenancy defines the distribution of EC2 instances across the physical hardware that affects pricing. It specifies if the Amazon EC2 instance is hosted on shared or single-tenant hardware.
VPC	Select the required Virtual Private Network ID to configure vThunder.
Zone	Select a zone from the existing availability zones. AWS offers a range of options for managing availability and resiliency for your applications. **NOTE:** Make use of the replicated VMs across the availability zones to protect your applications and data against the data center outages and maintenance events.

8. Click **Next**.

   The **Configure stack options** window is displayed.

9. Verify the other fields and change the values appropriately. (Optional)
10. Click **Next**.

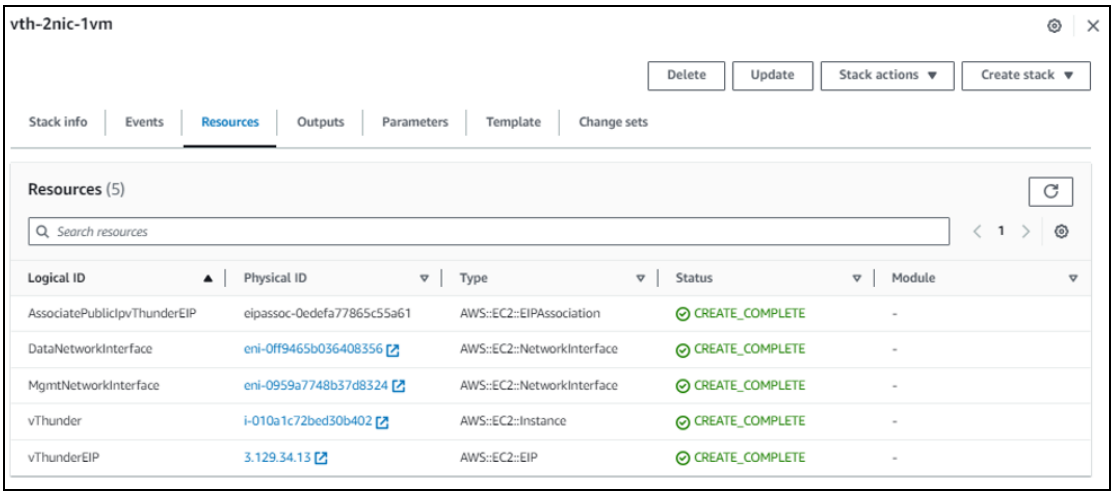
    The **Review** <*stack_name*> window is displayed.

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

NOTE: The system may take a few minutes to create the resources and display the stack status as **CREATE_COMPLETE**.

12. Verify if the elastic IPs are created in the **AWS Management Console** > **CloudFormation** > **Stacks** > <*stack_name*> > **Resources** tab.

Figure 4 : Resource listing



13. Access Thunder Virtual Machine.
14. Create Server Machine.
15. Create Client Machine.

Configure Thunder

The following configurations can be applied to the deployed vThunder instance depending on your use case and requirement, see Deployment Templates:

- Change Password
- A10 License
- SSL Certificate
- Basic Server Load Balancer
- Backend Server Autoscaling

Verify Deployment

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

1. Access the vThunder instance using CLI.
2. Run the following command to verify the running configuration:

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

If the deployment is successful with basic SLB, HTTP template, and Persist-cookie template configuration, the following output is displayed:

```
!
interface ethernet 1
  enable
  ip address dhcp
!
!
slb server server1 10.0.2.8
  port 53 udp
  port 80 tcp
  port 443 tcp
!
slb service-group sg443 tcp
  member server1 443
!
slb service-group sg53 udp
  member server1 53
!
slb service-group sg80 tcp
  member server1 80
!
slb template persist cookie persist-cookie
  expire 60
  encrypt-level 0
  name a10-cookies
  match-type service-group
!
slb template http hostname-test
```

```
  host-switching contains s1 service-group sg80
!
slb template http url-test
  url-switching regex-match s1 service-group sg80
!
slb virtual-server vip 10.0.2.148
  port 53 udp
    source-nat auto
    service-group sg53
  port 80 http
    source-nat auto
    service-group sg80
    template http url-test
    template persist cookie persist-cookie
  port 443 https
    source-nat auto
    service-group sg443
    template http url-test
    template persist cookie persist-cookie
!
!
end
```

If the deployment is successful with SLB on Backend Autoscale, HTTP template, and Persist-cookie template configuration, the following output is displayed:

```
!
interface ethernet 1
  enable
  ip address dhcp
!
!
slb server i-0177b3b4899596316 10.0.2.8
  port 53 udp
  port 80 tcp
  port 443 tcp
!
slb service-group sg443 tcp
```

```
  member i-0177b3b4899596316 443
!
slb service-group sg53 udp
  member i-0177b3b4899596316 53
!
slb service-group sg80 tcp
  member i-0177b3b4899596316 80
!
slb template persist cookie persist-cookie
  expire 60
  encrypt-level 0
  name a10-cookies
  match-type service-group
!
slb template http hostname-test
  host-switching contains s1 service-group sg80
!
slb template http url-test
  url-switching regex-match s1 service-group sg80
!
slb virtual-server vip 10.0.2.148
  port 53 udp
    source-nat auto
    service-group sg53
  port 80 http
    source-nat auto
    service-group sg80
    template http url-test
    template persist cookie persist-cookie
  port 443 https
    source-nat auto
    service-group sg443
    template http url-test
    template persist cookie persist-cookie
!
!
end
```

3. Run the following command to verify the SSL configuration:

```
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]
```

4. Run the following command on the vThunder instance to verify GLM configuration:

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

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

```
Host ID         : XXXXXXXXC264BECCCFEXXXXXXXXXXXXXXXX
USB ID          : Not Available
Billing Serials: A10XXXXecbe0000
Token           : A10f771cecbe
Product         : ADC
Platform        : vThunder
Burst           : Disabled
GLM Ping Interval In Hours : 24
-----------------------------------------------------------------
Enabled Licenses Expiry Date              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.
CYLANCE          N/A         Requires an additional Cylance license.
IPSEC_VPN        N/A         Requires an additional IPsec VPN license.
25 Mbps Bandwidth 21-December-2022
```

Verify Traffic Flow

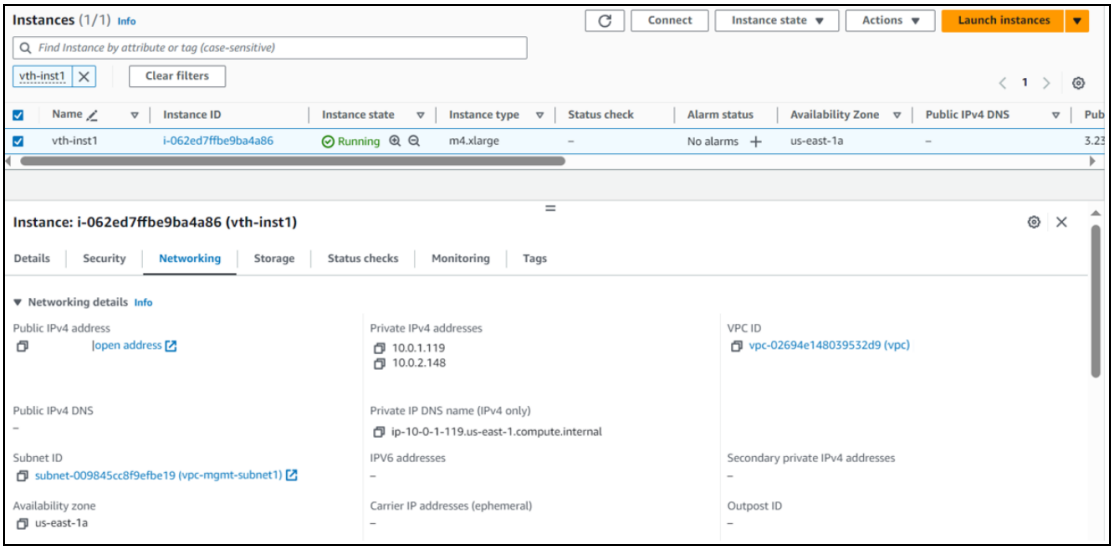
To verify the traffic flow from client machine to server machine via vThunder, perform the following:

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. Note down the IP address of the data subnet under the Private IPv4 address. Here, 10.0.2.148 is the data subnet value.

   Figure 5 : vThunder instance



4. Select your client instance from the **Instances** list.

   Here, **`client1`** 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. Replace the IP address of the data subnet noted above in the following command and then run the command in the Terminal window to send the traffic from the client machine:

```
curl <vThunder_instance_data-nic_private_ip>
```

8. Verify if a response is received from client server (For example: Apache Index page).

9. SSH your client machine and run the following command to verify the HTTP template traffic flow:

```
curl <vThunder_instance_data-nic_private_ip>:<port_number>/<host-match-string or url-match-string>/
```

Example

```
curl 10.0.2.148:80/s1/
```

Verify if a response is received from client server (For example: Apache Index page).

10. SSH your client machine and run the following commands to verify the Persist cookie template traffic flow:

    a. Verify the current cookie configuration:

    ```
    curl --head <vThunder_instance_data-nic_private_ip>
    ```

    b. Run the following commands to save the cookies in the **cookie.txt** file:

    ```
    curl -b cookie.txt -c cookie.txt <vThunder_instance_data-nic_private_ip>
    cat cookie.txt
    ```

    Example

    ```
    curl --head 10.0.2.148
    curl -b cookie.txt -c cookie.txt 10.0.2.148
    cat cookie.txt
    ```

11. Run the following command on the vThunder instance to view the persistence load-balancing statistics:

    ```
    vThunder(config)#show slb persist
    ```

    If the deployment is successful, the following summary persistence statistics is displayed:

```
                                        Total
-----------------------------------------------
URL hash persist (pri)                    0
URL hash persist (sec)                    0
URL hash persist fail                     0
SRC IP persist ok                         0
SRC IP persist fail                       0
SRC IP hash persist(pri)                  0
SRC IP hash persist(sec)                  0
SRC IP hash persist fail                  0
DST IP persist ok                         0
DST IP persist fail                       0
DST IP hash persist(pri)                  0
DST IP hash persist(sec)                  0
DST IP hash persist fail                  0
SSL SID persist ok                        0
SSL SID persist fail                      0
Cookie persist ok                         1
Cookie persist fail                       0
Persist cookie not found                  2
Persist cookie Pass-thru                  0
Enforce higher priority                   0
```

If the Persist-cookie configuration is successful, a value is displayed for the **`Cookie persist ok`** parameter, else the value is 0.

Thunder-3NIC-2VM

This template creates two vThunder instances in the same or different availability zone with HA setup and each vThunder has one management and two data NICs (data-in and data-out). It configures data-in network interface card (NIC) with Private VIP or Public VIP. It also provides feature to do the AWS HA failover using Private IP as Alien IP.

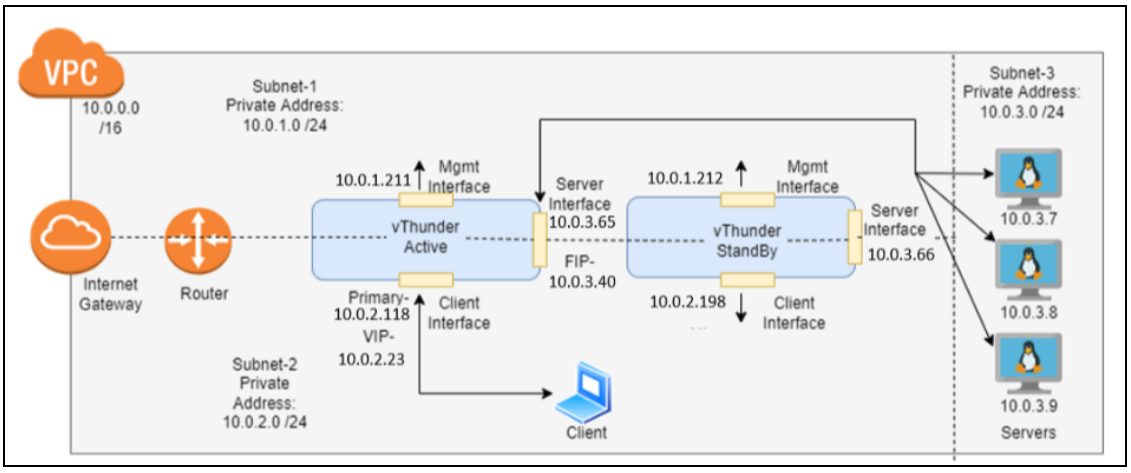
NOTE: Use a suitable VM size that supports at least three NICs. For VM sizes, see List of Supported Instance Types.

High Availability within same Availability Zone

To create two vThunder instances in the same availability zone with HA setup, see Create Thunder Virtual Machines for Same Availability Zone.

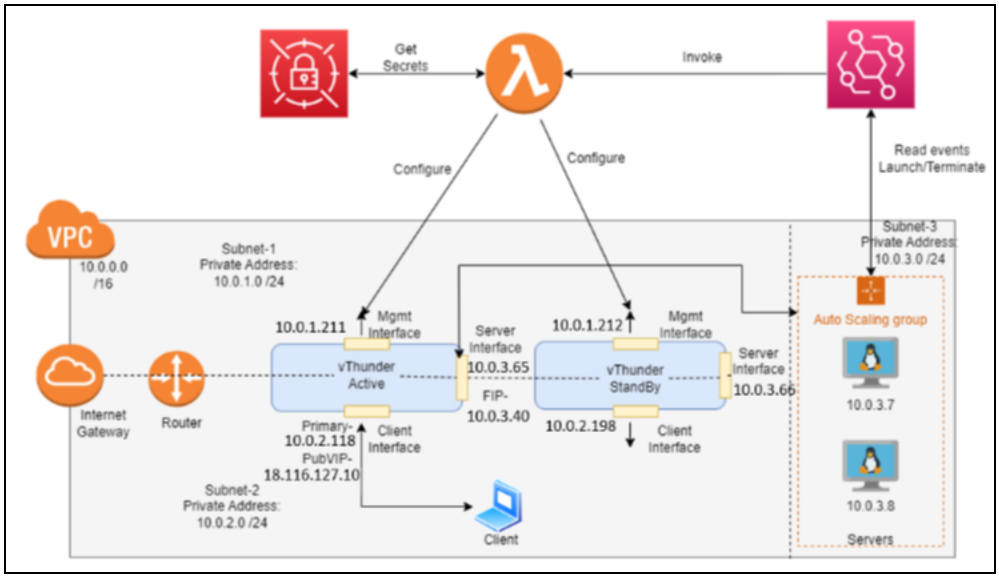
The following figure shows SLB Thunder ADC in High Availability mode with Private VIP:

Figure 6 : Thunder-3NIC-2VM with Private VIP



The following figure shows SLB Thunder ADC in High Availability mode with Public VIP:

Figure 7 : Thunder-3NIC-2VM with Public VIP



The following additional Thunder configurations are available that can be applied as needed:

- Change Password
- A10 License
- SSL Certificate
- Basic Server Load Balancer
- Backend Server Autoscaling
- High Availability

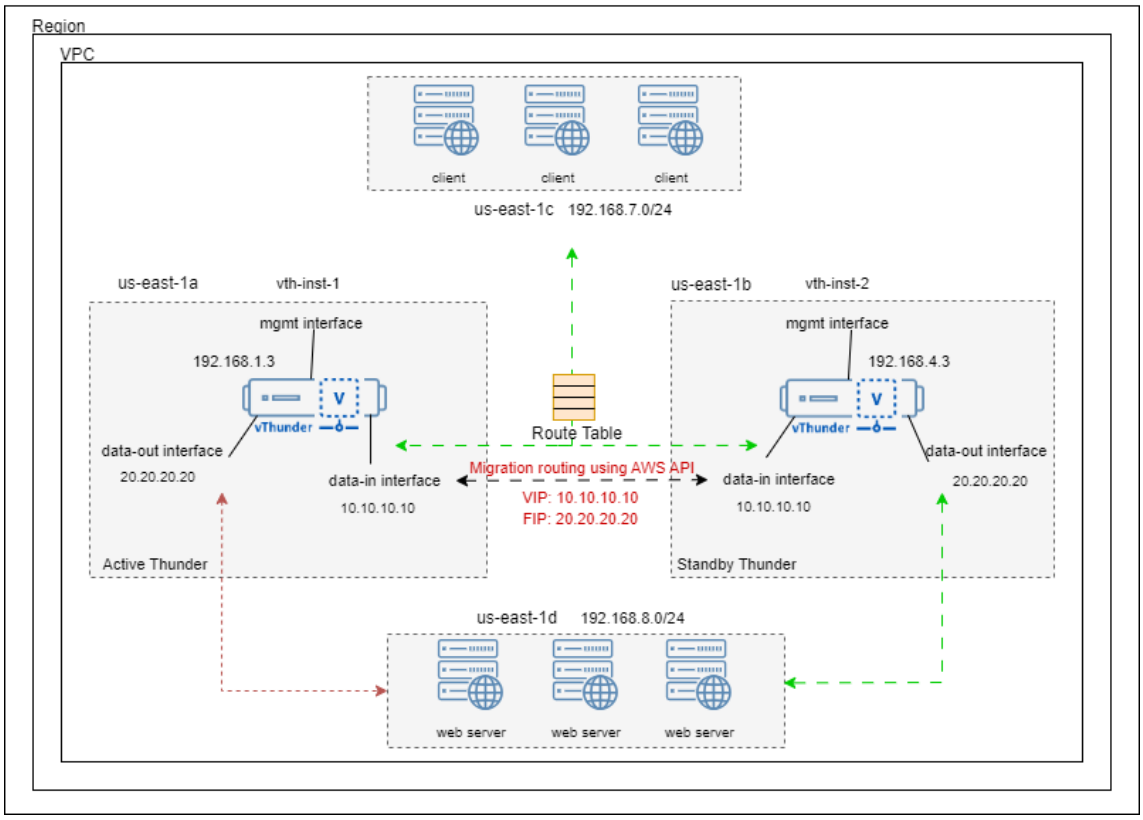
High Availability across different Availability Zones

The HA Across AZs only supports IPv4 and shared partitions with two Thunder devices at a time. It does not support more than two Thunder devices with multiple partitions.

To create two vThunder instances across different availability zones with HA setup, see Create Thunder Virtual Machines for Different Availability Zones.

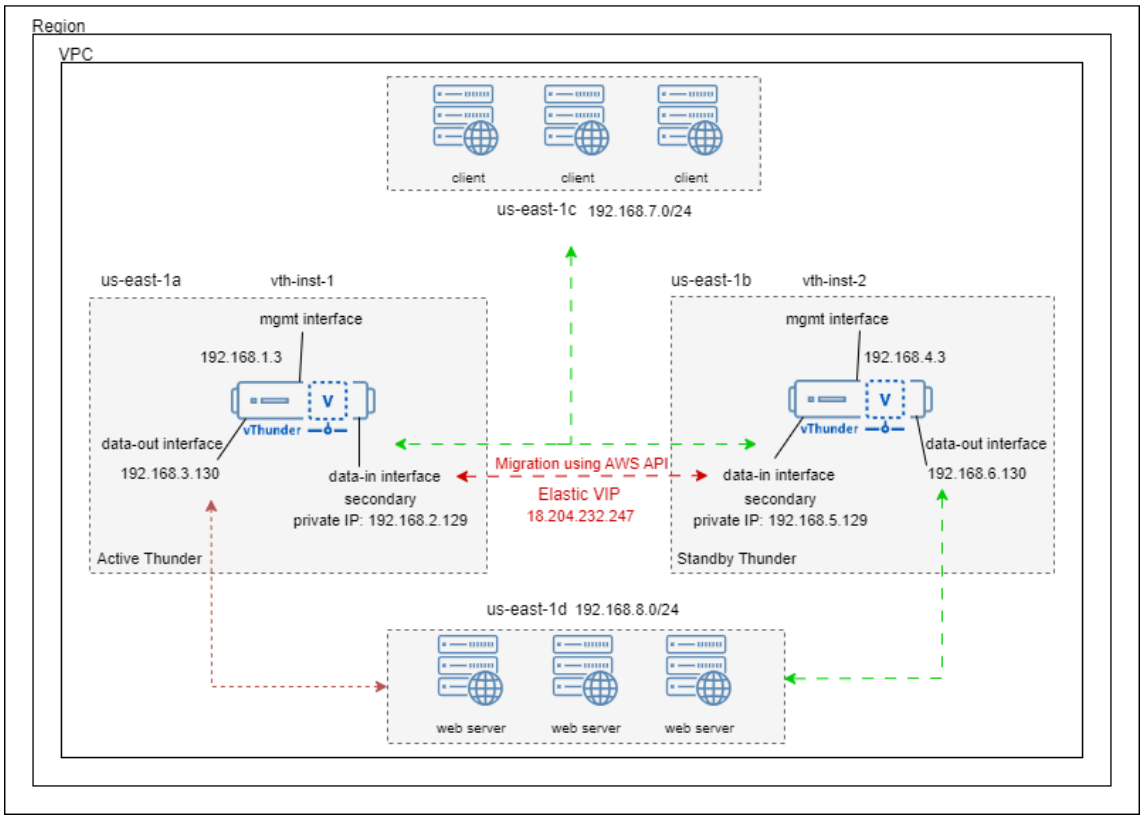
The following figure shows AWS HA using Private IP as Alien IP (Traffic from same VPC):

Figure 8 : Thunder-3NIC-2VM with Private VIP



The following figure shows AWS HA using Elastic IP (Traffic from internet):

Figure 9 : Thunder-3NIC-2VM with Public VIP



The following additional Thunder configurations are available that can be applied as needed:

- Change Password
- A10 License
- SSL Certificate
- Basic Server Load Balancer
- Backend Server Autoscaling
- High Availability Across Availability Zones

Various templates are available for different deployment needs.

For more information, see Deployment Templates.

The following topics are covered:

Create Thunder Virtual Machines for Same Availability Zone 41
Create Thunder Virtual Machines for Different Availability Zones 63

Create Thunder Virtual Machines for Same Availability Zone

The A10-vThunder-3NIC-2VM template is used to create two Thunder virtual machines in same availability zone with three network interface cards each and configure the data-in network interface card with Private VIP or Public VIP.

Before deploying this template, it is recommended to review the Prerequisites.

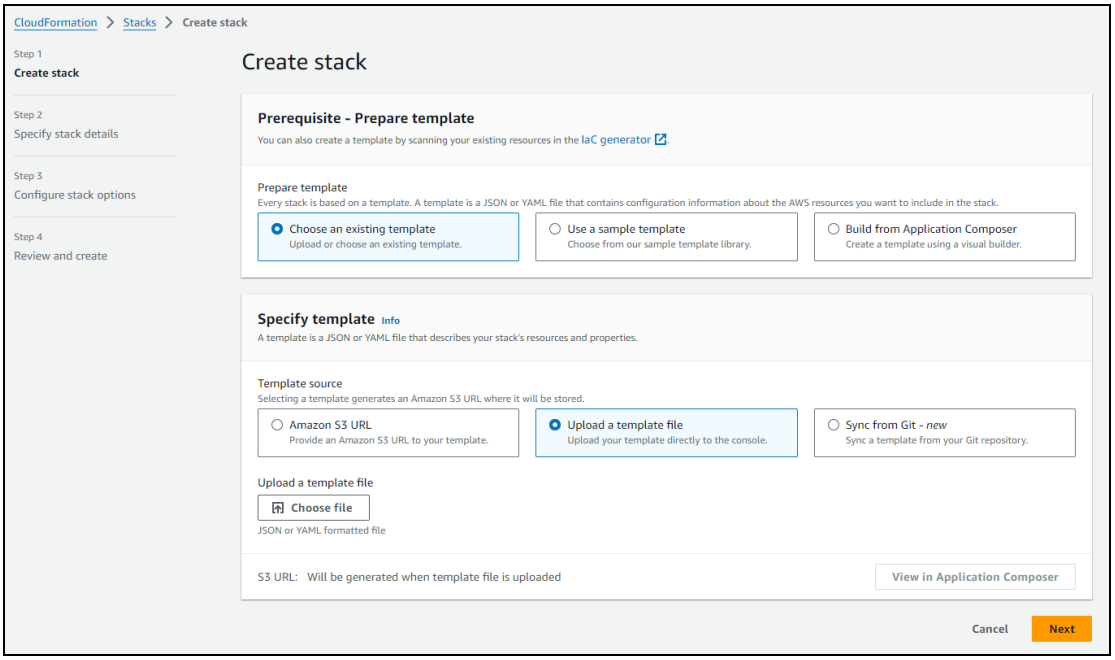
NOTE: vThunder instances should have the same versions; otherwise, traffic flow will be disrupted.

To create the Thunder virtual machines in same availability zones using A10-vThunder-3NIC-2VM template, perform the following steps:

1. Download **A10-vThunder_CFT-TEMPLATES** > **A10-vThunder-3NIC-2VM** template from GitHub.
2. From AWS Management Console, navigate to **CloudFormation** > **Stacks** > **Create Stack** > **With new resources (standard)**.

   The Create stack window is displayed.

Figure 10 : Create stack window



3. In the **Prerequisite - Prepare template** section, select **Choose an existing template**.

   After 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 the following template file from the downloaded CFT folder:
   CFT_TMPL_3NIC_2VM.json

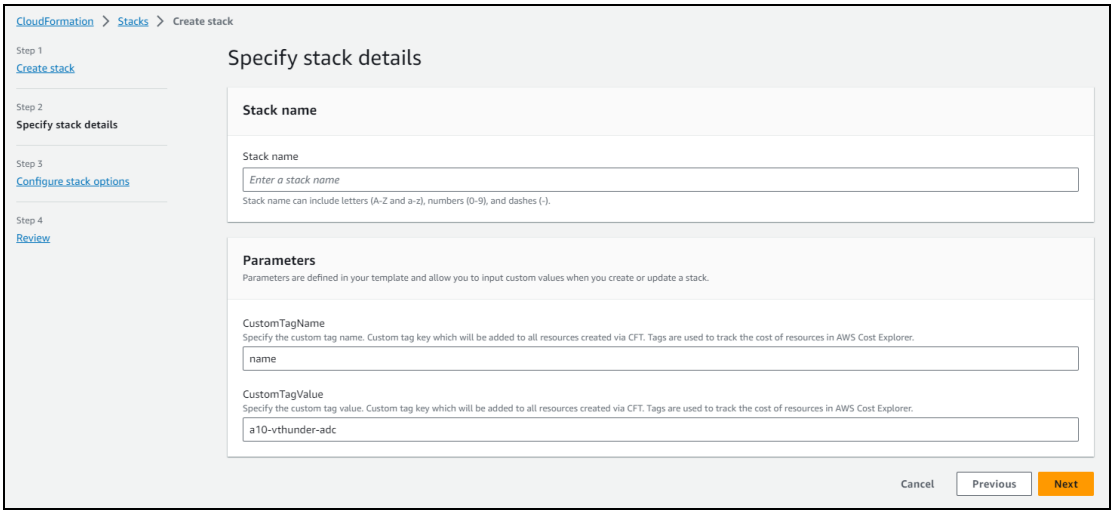
   The selected template file name is displayed as the chosen file.

NOTE: This template contains pre-populated default values that can be modified as required. It does not create a new primary private IP address, virtual private IP address, security groups, subnets, and virtual private network.

5. Click **Next**.

   The **Specify stack details** window is displayed.

Figure 11 : Specify stack details window



6. In the **Specify stack details** window, enter or select the following:

Table 4 : JSON Parameters

Resource Name	Description
Stack name	Specify a stack name containing letters (A-Z and a-z), numbers (0-9), and dashes (-). Here, the stack name is provided as **`vth-3nic-2vm`**.
AMIID	Specify the AMI ID of the required A10 vThunder image. By default, A10 Thunder ADC for Advanced Load Balancing - BYOL AMI ID is set for us-east-1 region. To get the AMI ID, go to the **AWS Management Console** > **EC2** > **Launch instance** > **Application and OS Images (Amazon Machine Image)** > search for the relevant vThunder image and then copy the AMI ID. For more information on the available Thunder images with different capacities under respective regions, see List of ACOS AMI ID.
AllocationIdEIPActive	Specify an allocation ID of the required active elastic IP.

Table 4 : JSON Parameters

Resource Name	Description
	This elastic IP address is the management public IP address of the Thunder instance 1. To get the allocation ID, go to the **AWS Management Console** > **EC2** > **Elastic IPs** > <*elastic_IP_address*> > **Summary** and then copy the **Allocation ID**.
AllocationIdEIPStandBy	Specify an allocation ID of the required standby elastic IP. This elastic IP address is the management public IP address of the Thunder instance 2. To get the allocation ID, go to the **AWS Management Console** > **EC2** > **Elastic IPs** > <*elastic_IP_address*> > **Summary** and then copy the **Allocation ID**.
CreatePubVIP	Select **Yes** if VIP is a public IP address or select **No** if VIP is a private IP address.
CustomTagName	Specify the custom tag name. The custom tag key is added to all the resources created using the CFT template. This tag is used to track the cost of resources in the AWS Cost Explorer.
CustomTagValue	Specify the custom tag value. The custom tag key is added to all the resources created using the CFT template. This tag is used to track the cost of resources in the AWS Cost Explorer. The default custom tag value is **a10-vthunder-adc**.
EC2Instance	Select an instance/compute type supported for vThunder from the available list. By default, instance type is m4.xlarge which contains 4 vCPU and 16 GiB memory. The minimum requirement for vThunder is 4 vCPU and 16 GiB memory. For more information on product pricing, see AWS

Table 4 : JSON Parameters

Resource Name	Description
	Marketplace. If the required instance type is not available in the list, go to **AWS Management Console** > **EC2** > **Launch instance** > Find and copy the relevant instance type in the template. For more information, see Supported Instance Types.
KeyPair	Select a keypair from the available list. A key pair is a set of security credentials which consist of a public key and a private key. This keypair is used to verify your identity when connecting to an Amazon EC2 instance.
PrimaryPrivateIP	Specify the primary private IP address from data-in subnet CIDR. This IP address is the primary private IP address of the Thunder instance 1.
SecondayPrivateIP	Specify the secondary private IP address from data-in subnet CIDR. This IP address is the virtual private IP (VIP) address of the Thunder instance 1.
SecurityGroupData	Select the required security group ID from the available list for data interface within the selected Virtual Private Network.
SecurityGroupMgmt	Select the required security group ID from the available list for the management interface within the selected Virtual Private Network.
SubnetDataINID	Select the required subnet ID from the available list for the data traffic flow inward to vThunder within the selected Virtual Private Network.
SubnetDataOutID	Select the required subnet ID from the available list for the data traffic flow outward from vThunder within the selected Virtual Private Network.
SubnetMgmtID	Select the required subnet ID from the available list for managing vThunder within the selected Virtual Private Network.

Table 4 : JSON Parameters

Resource Name	Description
Tenancy	Select the required tenancy from the available list. Tenancy defines the distribution of EC2 instances across the physical hardware that affects pricing. It specifies if the Amazon EC2 instance is hosted on shared or single-tenant hardware.
VPC	Select the required Virtual Private Network ID to configure vThunder.
Zone	Select a zone from the existing availability zones. AWS offers a range of options for managing availability and resiliency for your applications. **NOTE:** Make use of the replicated VMs across the availability zones to protect your applications and data against the data center outages and maintenance events.

7. Click **Next**.

   The **Configure stack options** window is displayed.

8. Verify the other fields and change the values appropriately. (Optional)
9. Click **Next**.

   The **Review** <*stack_name*> window is displayed.

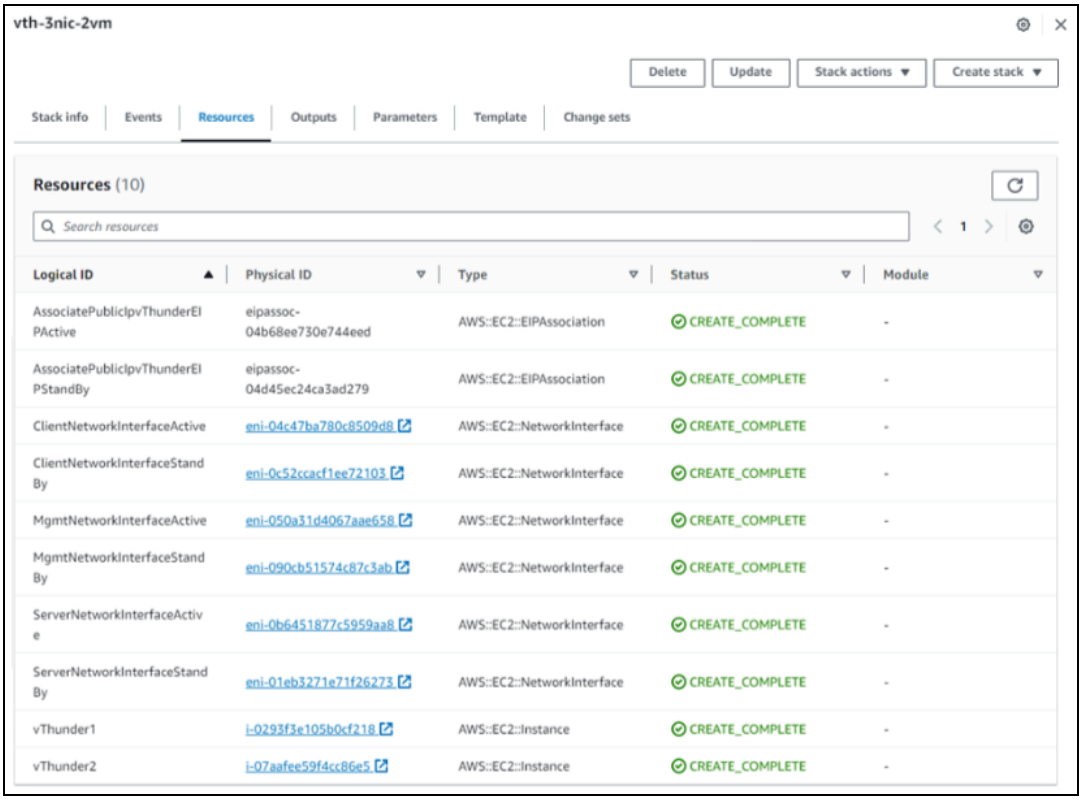
10. Verify if all the stack configurations are correct and then click **Submit**.

NOTE: The system may take a few minutes to create the resources and display the stack status as **CREATE_COMPLETE**.

11. Verify if the elastic IPs are created in the **AWS Management Console** > **CloudFormation** > **Stacks** > <*stack_name*> > **Resources** tab:

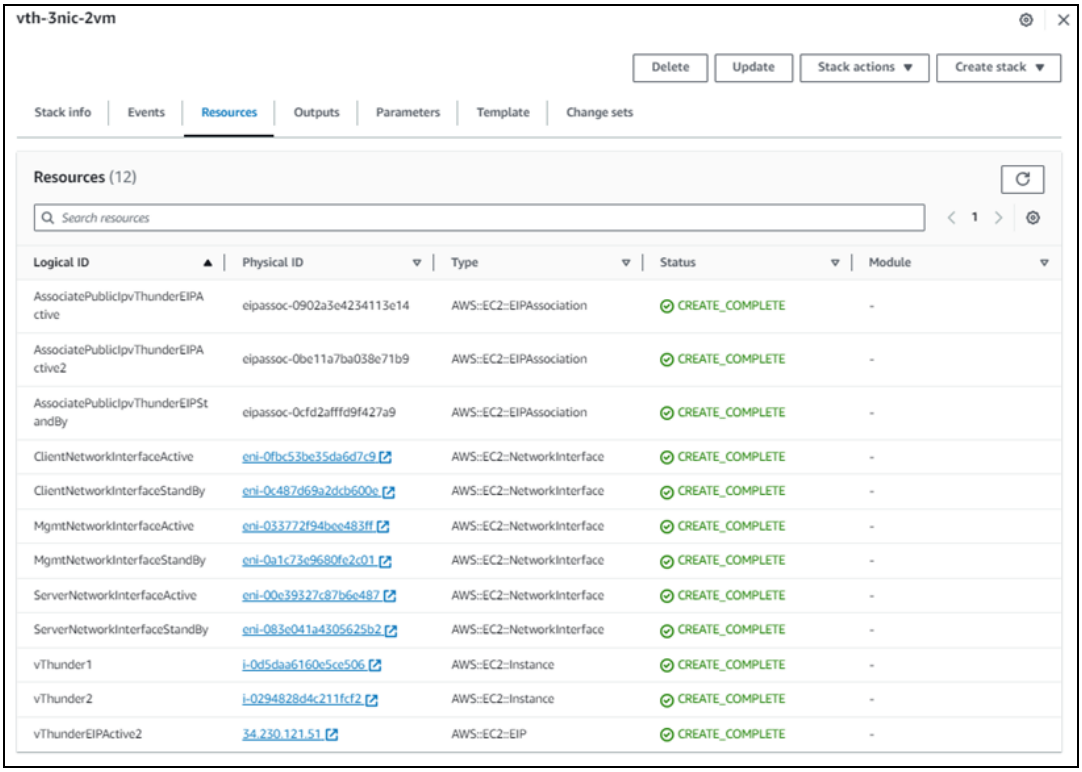
- Data-in network interface card with Private IP

Figure 12 : PVTVIP Resource listing



- Data-in network interface card with Public IP

Figure 13 : PUBVIP Resource listing



12. Access Thunder Virtual Machine.
13. Create Server Machine.
14. Create Client Machine.

Configure Thunder

The following configurations can be applied to the deployed vThunder instance depending on your use case and requirement, see Deployment Templates:

- Change Password
- A10 License
- SSL Certificate
- Basic Server Load Balancer
- Backend Server Autoscaling
- High Availability

Verify Deployment

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

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

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

If the deployment is successful with basic SLB, HTTP template, and Persist-cookie template configuration, the following output 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 10.0.3.40
  blade-parameters
    priority 99
!
vrrp-a peer-group
  peer 10.0.2.118
  peer 10.0.2.198
!
ip route 0.0.0.0 /0 10.0.1.1
ip route 0.0.0.0 /0 10.0.2.1
!
slb server server1 10.0.3.23
  port 53 udp
  port 80 tcp
  port 443 tcp
!
slb service-group sg443 tcp
  member server1 443
!
slb service-group sg53 udp
  member server1 53
!
slb service-group sg80 tcp
  member server1 80
!
slb template persist cookie persist-cookie
  expire 60
  encrypt-level 0
  name a10-cookies
  match-type service-group
!
slb template http hostname-test
  host-switching contains s1 service-group sg80
!
slb template http url-test
  url-switching regex-match s1 service-group sg80
```

```
!
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
    template http url-test
    template persist cookie persist-cookie
  port 443 https
    source-nat auto
    service-group sg443
    template http url-test
    template persist cookie persist-cookie
!
!
end
!Current config commit point for partition 0 is 0 & config mode is
classical-mode
vThunder-Active(config)#
```

If the deployment is successful with SLB on Backend Autoscale, HTTP template, and Persist-cookie template configuration, the following output 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 10.0.3.40
  blade-parameters
    priority 99
!
vrrp-a peer-group
  peer 10.0.2.118
  peer 10.0.2.198
!
ip route 0.0.0.0 /0 10.0.1.1
ip route 0.0.0.0 /0 10.0.2.1
!
slb server i-0177b3b4899596316 10.0.3.23
  port 53 udp
  port 80 tcp
  port 443 tcp
!
slb service-group sg443 tcp
  member i-0177b3b4899596316 443
!
slb service-group sg53 udp
  member i-0177b3b4899596316 53
```

```
!
slb service-group sg80 tcp
  member i-0177b3b4899596316 80
!
slb template persist cookie persist-cookie
  expire 60
  encrypt-level 0
  name a10-cookies
  match-type service-group
!
slb template http hostname-test
  host-switching contains s1 service-group sg80
!
slb template http url-test
  url-switching regex-match s1 service-group sg80
!
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
    template http url-test
    template persist cookie persist-cookie
  port 443 https
    source-nat auto
    service-group sg443
    template http url-test
    template persist cookie persist-cookie
!
!
end
!Current config commit point for partition 0 is 0 & config mode is
classical-mode
vThunder-Active(config)#
```

3. Run the following command on vThunder instance 1:

```
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 14 : vThunder instance 1 - Active

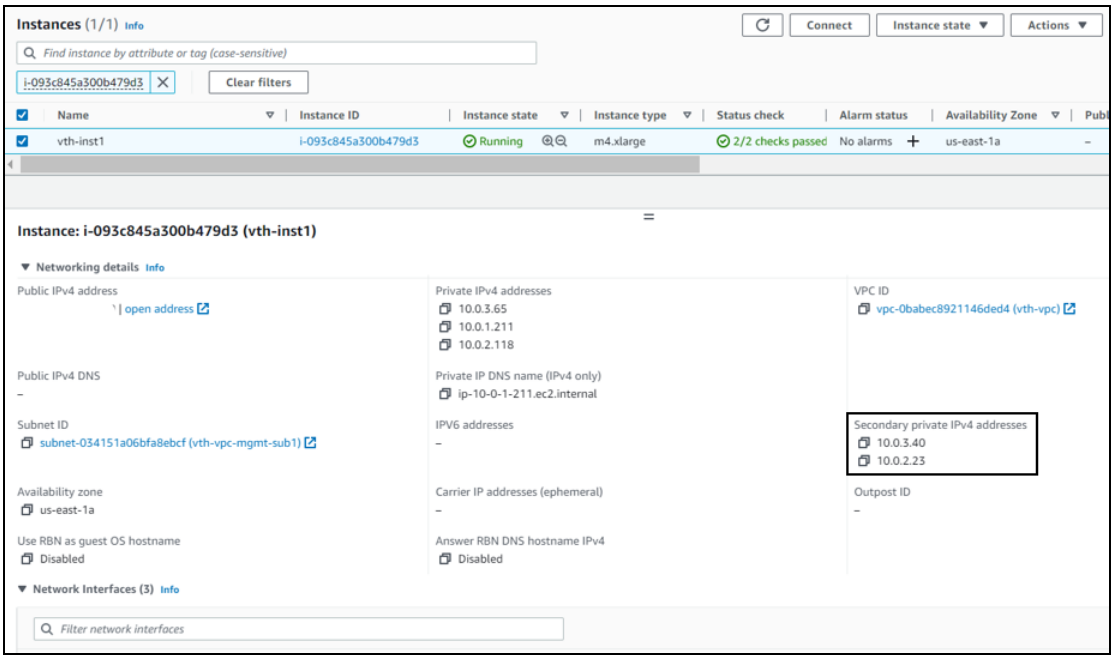
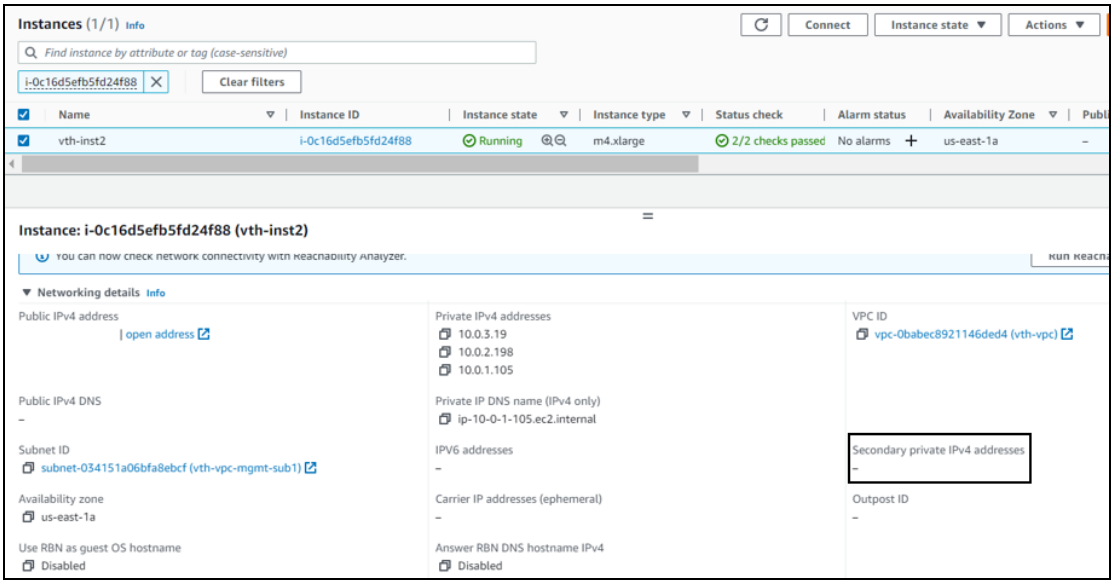


Figure 15 : vThunder instance 2 - Standby



4. Run the following command on vThunder instance 1:

```
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)#
```

5. Run the following command on vThunder instance 2:

```
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 10.0.3.40
  blade-parameters
    priority 98
!
vrrp-a peer-group
  peer 10.0.2.118
  peer 10.0.2.198
!
ip route 0.0.0.0 /0 10.0.1.1
ip route 0.0.0.0 /0 10.0.2.1
!
slb server server1 10.0.3.23
  port 53 udp
  port 80 tcp
  port 443 tcp
!
slb service-group sg443 tcp
```

```
  member server1 443
!
slb service-group sg53 udp
  member server1 53
!
slb service-group sg80 tcp
  member server1 80
!
slb template persist cookie persist-cookie
  expire 60
  encrypt-level 0
  name a10-cookies
  match-type service-group
!
slb template http hostname-test
  host-switching contains s1 service-group sg80
!
slb template http url-test
  url-switching regex-match s1 service-group sg80
!
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
```

```
    template http url-test
    template persist cookie persist-cookie
  port 443 https
    source-nat auto
    service-group sg443
    template http url-test
    template persist cookie persist-cookie
!
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)#
```

6. Run the following command on vThunder instance 2:

```
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 16 : vThunder instance 1 - Standby

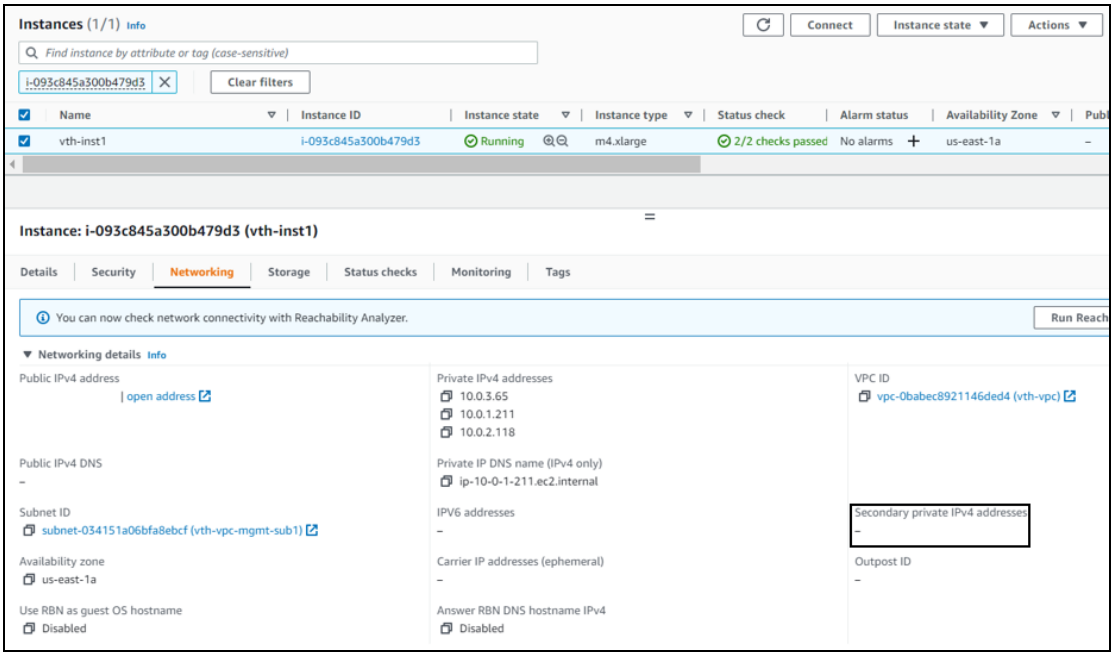
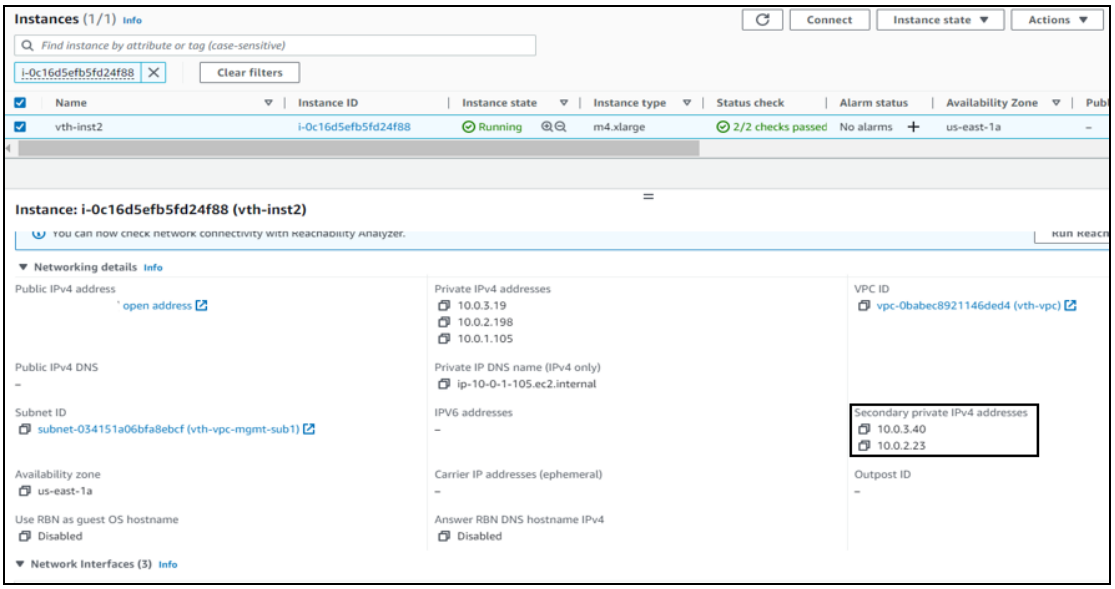


Figure 17 : vThunder instance 2 - Active



7. If you want to make vThunder instance 1 active, run the following command on vThunder instance 1:

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

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

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

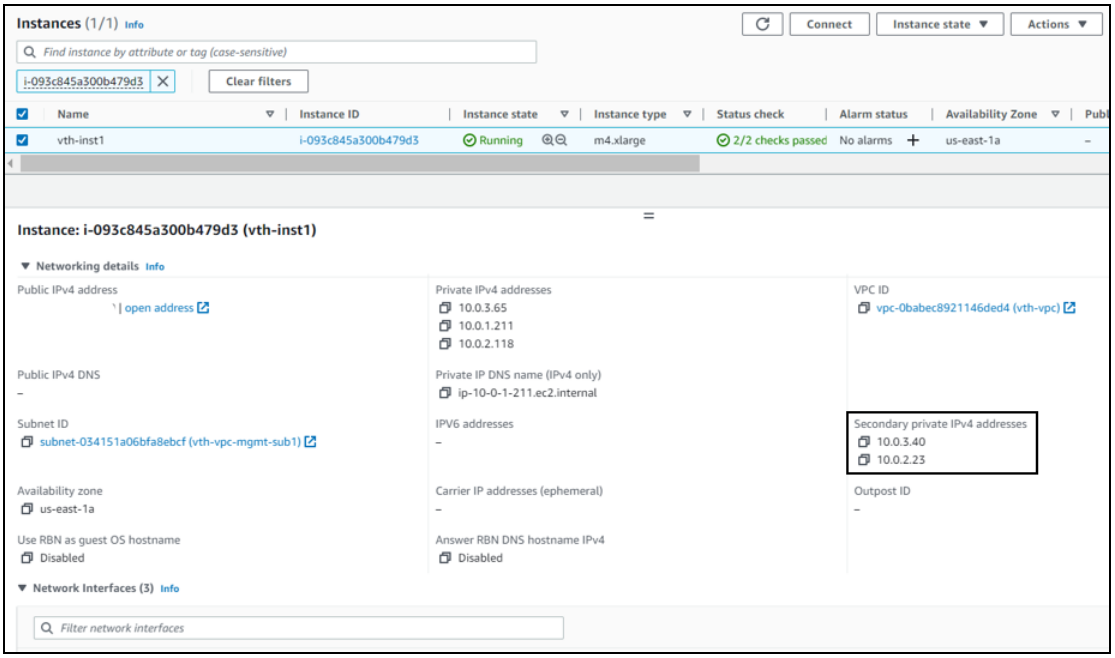
NOTE: If you want to access the vThunder instances using GUI, re-login the instances using the new admin user password and NOT the *EC2 Instance ID*.

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. Note down the Data-In Subnet VIP address of the active vThunder instance.

   Figure 18 : 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. Replace the Data-In Subnet VIP address of the active vThunder instance noted above in the following command and then run the command in the Terminal window to send the traffic from the client machine:

   ```
   curl <vThunder_instance_datain-nic_secondary_private_or_public_ip_vip>
   ```

   Example for Private VIP

   ```
   curl 10.0.2.23
   ```

   Example for Public VIP

   ```
   curl 18.116.127.10
   ```

8. Verify if a response is received.
9. SSH your client machine and run the following command to verify the HTTP template traffic flow:

   ```
   curl <vThunder_instance_datain-nic_secondary_private_or_public_ip_
   vip>:<port_number>/<host-match-string or url-match-string>/
   ```

   Example for Private VIP

   ```
   curl 10.0.2.23:80/s1/
   ```

   Example for Public VIP

   ```
   curl 18.116.127.10:80/s1/
   ```

   Verify if a response is received from client server (For example: Apache Index page).

10. SSH your client machine and run the following commands to verify the Persist cookie template traffic flow:

    a. Verify the current cookie configuration:

    ```
    curl --head <vThunder_instance_datain-nic_private_or_public_ip_vip>
    ```

    b. Run the following commands to save the cookies in the **`cookie.txt`** file:

    ```
    curl -b cookie.txt -c cookie.txt <vThunder_instance_datain-nic_
    private_or_public_ip_vip>
    cat cookie.txt
    ```

Example for Private VIP

```
curl --head 10.0.2.23
curl -b cookie.txt -c cookie.txt 10.0.2.23
cat cookie.txt
```

Example for Public VIP

```
curl --head 18.116.127.10
curl -b cookie.txt -c cookie.txt 18.116.127.10
cat cookie.txt
```

11. Run the following command on the active vThunder instance to view the persistence load-balancing statistics:

```
vThunder(config)#show slb persist
```

If the deployment is successful, the following summary persistence statistics is displayed:

```
                                        Total
-----------------------------------------------
URL hash persist (pri)                      0
URL hash persist (sec)                      0
URL hash persist fail                       0
SRC IP persist ok                           0
SRC IP persist fail                         0
SRC IP hash persist(pri)                    0
SRC IP hash persist(sec)                    0
SRC IP hash persist fail                    0
DST IP persist ok                           0
DST IP persist fail                         0
DST IP hash persist(pri)                    0
DST IP hash persist(sec)                    0
DST IP hash persist fail                    0
SSL SID persist ok                          0
SSL SID persist fail                        0
Cookie persist ok                           1
Cookie persist fail                         0
Persist cookie not found                    2
Persist cookie Pass-thru                    0
Enforce higher priority                     0
```

If the Persist-cookie configuration is successful, a value is displayed for the **`Cookie persist ok`** else the value is 0.

Create Thunder Virtual Machines for Different Availability Zones

The A10-vThunder-3NIC-2VM template is used to create two Thunder virtual machines across different availability zones with three network interface cards each and configure the data-in network interface card with Private VIP or Public VIP.

Before deploying this template, it is recommended to review the Prerequisites.

NOTE: vThunder instances should have the same versions; otherwise, traffic flow will be disrupted.

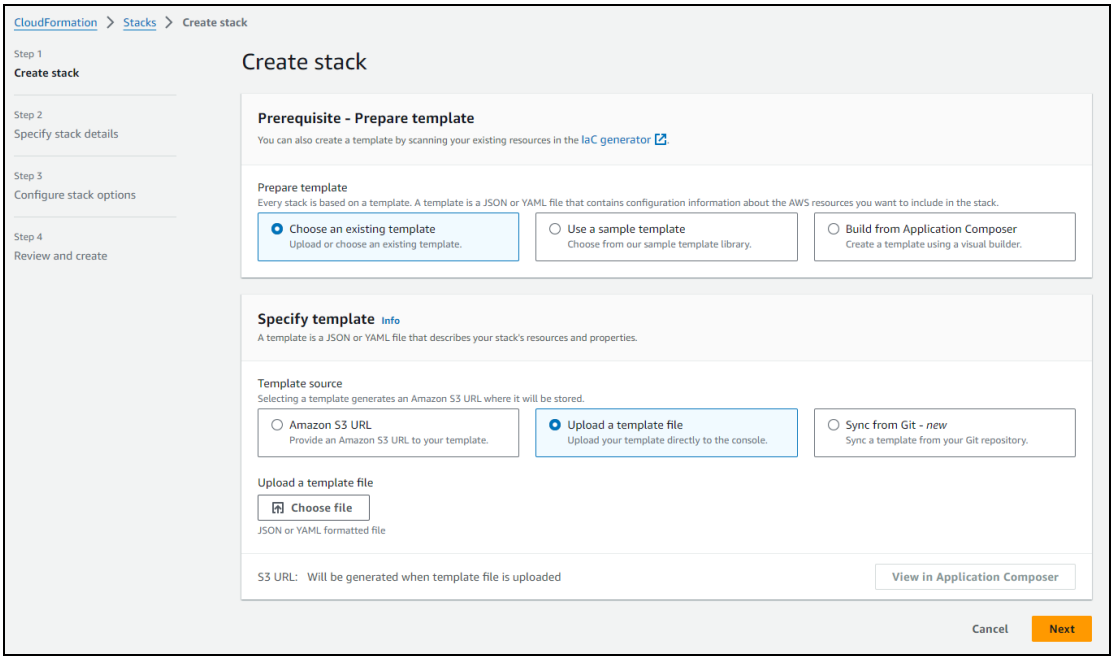
To create the Thunder virtual machines across different availability zones using A10-vThunder-3NIC-2VM template, perform the following steps:

1. Download **A10-vThunder_CFT-TEMPLATES** > **A10-vThunder-3NIC-2VM** template from GitHub.

2. From AWS Management Console, navigate to **CloudFormation** > **Stacks** > **Create Stack** > **With new resources (standard)**.

   The Create stack window is displayed.

   Figure 19 : Create stack window



3. In the **Prerequisite - Prepare template** section, select **Choose an existing template**.

   After 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 the following template file from the downloaded CFT folder:
   CFT_TMPL_3NIC_2VM_ACROSS_AZ_HA.json

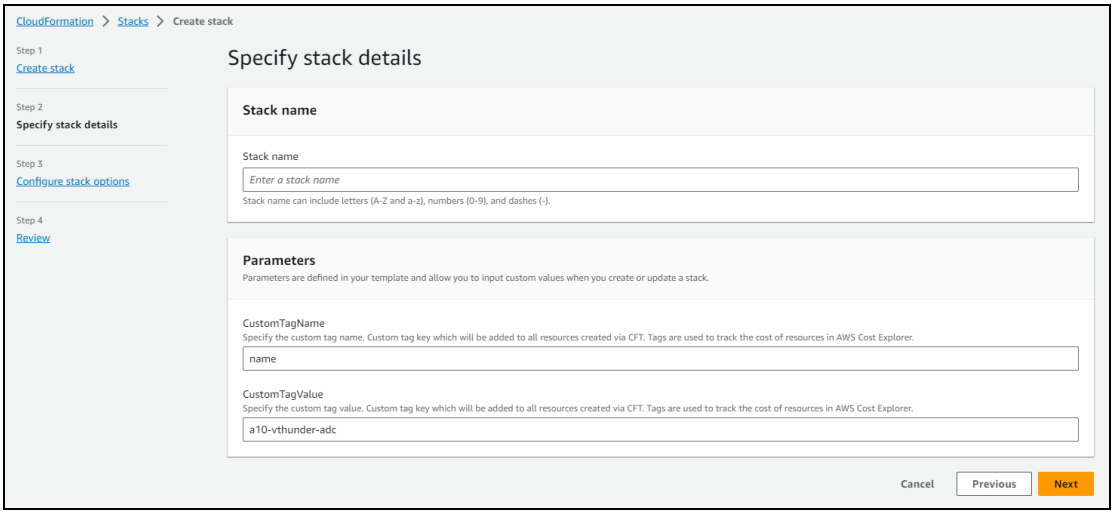
   The selected template file name is displayed as the chosen file.

NOTE: This template contains pre-populated default values that can be modified as required. It does not create a new primary private IP address, virtual private IP address, security groups, subnets, and virtual private network.

5. Click **Next**.

   The **Specify stack details** window is displayed.

   Figure 20 : Specify stack details window



6. In the **Specify stack details** window, enter or select the following:

   Table 5 : JSON Parameters

Resource Name	Description
Stack name	Specify a stack name containing letters (A-Z and a-z), numbers (0-9), and dashes (-). Here, the stack name is provided as **`vth-3nic-2vm`**.
AMIID	Specify the AMI ID of the required A10 vThunder image. By default, A10 Thunder ADC for Advanced Load Balancing - BYOL AMI ID is set for **`us-east-`**

Table 5 : JSON Parameters

Resource Name	Description
	1 region. To get the AMI ID, go to the **AWS Management Console** > **EC2** > **Launch instance** > **Application and OS Images (Amazon Machine Image)** > search for the relevant vThunder image and then copy the AMI ID. For more information on the available Thunder images with different capacities under respective regions, see List of ACOS AMI ID.
AllocationIdEIPActiveThunder	Specify an allocation ID of the required active elastic IP. This elastic IP address is the management public IP address of the Thunder instance 1. To get the allocation ID, go to the **AWS Management Console** > **EC2** > **Elastic IPs** > <*elastic_IP_address*> > **Summary** and then copy the **Allocation ID**.
AllocationIdEIPStandByThunder	Specify an allocation ID of the required standby elastic IP. This elastic IP address is the management public IP address of the Thunder instance 2. To get the allocation ID, go to

Table 5 : JSON Parameters

Resource Name	Description
	the **AWS Management Console** > **EC2** > **Elastic IPs** > <*elastic_IP_address*> > **Summary** and then copy the **Allocation ID**.
AttachAlienIP	Select **Yes** if you want to use Alien IP. This will configure the HA setup with the Private IP (Alien IP). The default set value is **No**.
CreatePubVIP	Select **Yes** if VIP is a public IP address or select **No** if VIP is a private IP address.
CustomTagName	Specify the custom tag name. The custom tag key is added to all the resources created using the CFT template. This tag is used to track the cost of resources in the AWS Cost Explorer.
CustomTagValue	Specify the custom tag value. The custom tag key is added to all the resources created using the CFT template. This tag is used to track the cost of resources in the AWS Cost Explorer. The default custom tag value is **a10-vthunder-adc**.
DifferentAvailabilityZone	Select **Yes** if you want to deploy your vThunder instances across different availability zones. The default set value is **No**.

Table 5 : JSON Parameters

Resource Name	Description
DataInDestinationCidr	If you have selected **Yes** for AttachAlienIP, specify the CIDR for data-in interface that does not overlap with the selected VPC subnets.
DataOutDestinationCidr	If you have selected **Yes** for AttachAlienIP, specify the CIDR for data-out interface that does not overlap with the selected VPC subnets.
EC2Instance	Select an instance/compute type supported for vThunder from the available list. By default, instance type is m4.xlarge which contains 4 vCPU and 16 GiB memory. The minimum requirement for vThunder is 4 vCPU and 16 GiB memory. For more information on product pricing, see AWS Marketplace. If the required instance type is not available in the list, go to **AWS Management Console** > **EC2** > **Launch instance** > Find and copy the relevant instance type in the template. For more information, see Supported Instance Types.
KeyPair	Select a keypair from the available list. A key pair is a set of security credentials which consist of a public key and a

Table 5 : JSON Parameters

Resource Name	Description
	private key. This keypair is used to verify your identity when connecting to an Amazon EC2 instance.
PrimaryPrivateIP	Specify the primary private IP address from data-in subnet CIDR. This IP address is the primary private IP address of the vThunder instance 1.
RouteTable	Specify the route table ID for the selected Virtual Private Network. This CFT template does not create a new route table.
SecondaryPrivateIPDataInActiveThunder	Specify data-in interface secondary IP for active vThunder instance. For example: **`vth-inst-data-in-active-ip`**. This CFT template does not create a new Virtual private IP address for virtual machine 1.
SecondaryPrivateIPDataInStandbyThunder	Specify data-in interface secondary IP for standby vThunder instance. For example: **`vth-inst-data-in-standby-ip`**. This CFT template does not create a new Virtual private IP address for virtual machine 2.
SecondaryPrivateIPDataOutActiveThunder	Specify data-out interface secondary IP for active

Table 5 : JSON Parameters

Resource Name	Description
	vThunder instance. For example, **`vth-inst-data-out-active-ip`**. This CFT template does not create a new Virtual private IP address for virtual machine 1.
SecondaryPrivateIPDataOutStandbyThunder	Specify data-out interface secondary IP for standby vThunder. For example, **`vth-inst-data-out-standby-ip`**. This CFT template does not create a new Virtual private IP address for virtual machine 2.
SecurityGroupData	Select the required security group ID from the available list for data interface within the selected Virtual Private Network.
SecurityGroupMgmt	Select the required security group ID from the available list for the management interface within the selected Virtual Private Network.
SubnetDataInActiveIDThunder	Select the required subnet ID from the available list for the data traffic flow inward to active vThunder instance within the selected Virtual Private Network. This CFT template does not create a new subnet.

Table 5 : JSON Parameters

Resource Name	Description
SubnetDataInStandByIDThunder	Select the required subnet ID from the available list for the data traffic flow inward to standby vThunder instance within the selected Virtual Private Network. This CFT template does not create a new subnet.
SubnetDataOutActiveIDThunder	Select the required subnet ID from the available list for the data traffic flow outward from active vThunder instance within the selected Virtual Private Network. This CFT template does not create a new subnet.
SubnetDataOutStandByIDThunder	Select the required subnet ID from the available list for the data traffic flow outward from standby vThunder instance within the selected Virtual Private Network. This CFT template does not create a new subnet.
SubnetMgmtActiveIDThunder	Select the required subnet ID from the available list for managing active vThunder instance within the selected Virtual Private Network. This CFT template does not create a new subnet.

Table 5 : JSON Parameters

Resource Name	Description
SubnetMgmtStandByIDThunder	Select the required subnet ID from the available list for managing standby vThunder instance within the selected Virtual Private Network. This CFT template does not create a new subnet.
Tenancy	Select the required tenancy from the available list. Tenancy defines the distribution of EC2 instances across the physical hardware that affects pricing. It specifies if the Amazon EC2 instance is hosted on shared or single-tenant hardware.
VPC	Select the required Virtual Private Network ID to configure vThunder instance.
ZoneActiveThunder	Select a zone from the existing availability zones for active vThunder instance. AWS offers a range of options for managing availability and resiliency for your applications. Make use of the replicated VMs across the availability zones to protect your applications and data against the data center outages and maintenance events.
ZoneStandByThunder	Select a zone from the existing availability zones for standby

Table 5 : JSON Parameters

Resource Name	Description
	vThunder instance. AWS offers a range of options for managing availability and resiliency for your applications. Make use of the replicated VMs across the availability zones to protect your applications and data against the data center outages and maintenance events.

7. Click **Next**.

   The **Configure stack options** window is displayed.

8. Verify the other fields and change the values appropriately. (Optional)
9. Click **Next**.

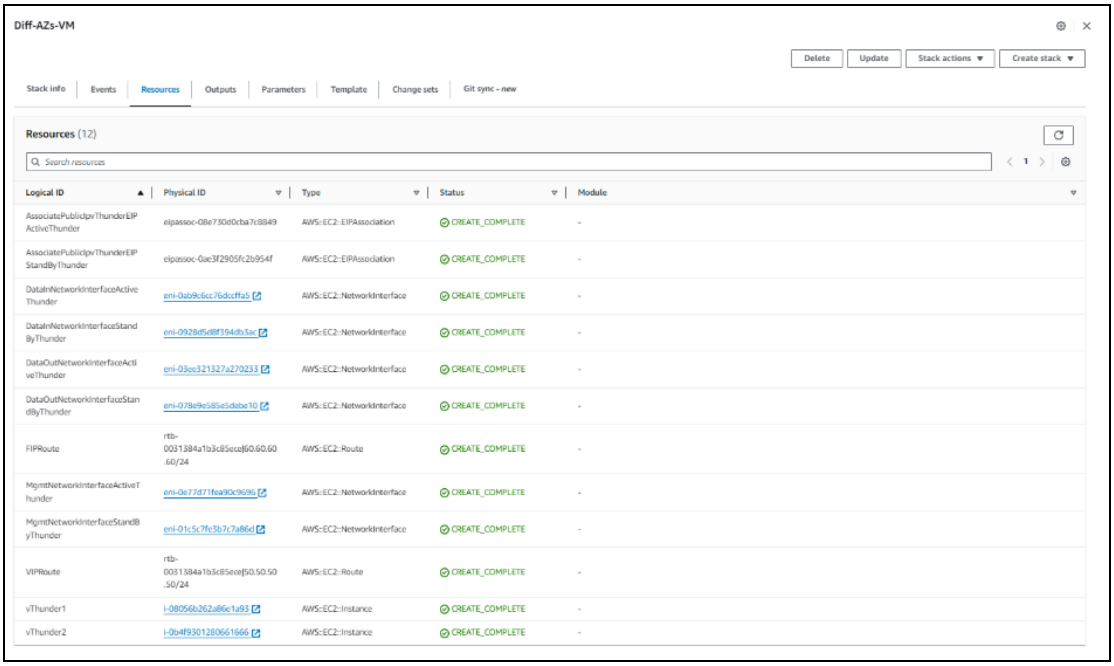
   The **Review** <*stack_name*> window is displayed.

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 and display the stack status as **CREATE_COMPLETE**.

11. Verify if the elastic IPs are created in the **AWS Management Console** > **CloudFormation** > **Stacks** > <*stack_name*> > **Resources** tab:

Figure 21 : Different AZs Resource Listing



12. Access Thunder Virtual Machine.
13. Create Server Machine.
14. Create Client Machine.

Configure Thunder

The following configurations can be applied to the deployed vThunder instance depending on your use case and requirement, see Deployment Templates:

- Change Password
- A10 License
- SSL Certificate
- Basic Server Load Balancer
- Backend Server Autoscaling
- High Availability Across Availability Zones

Verify Deployment

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

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

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

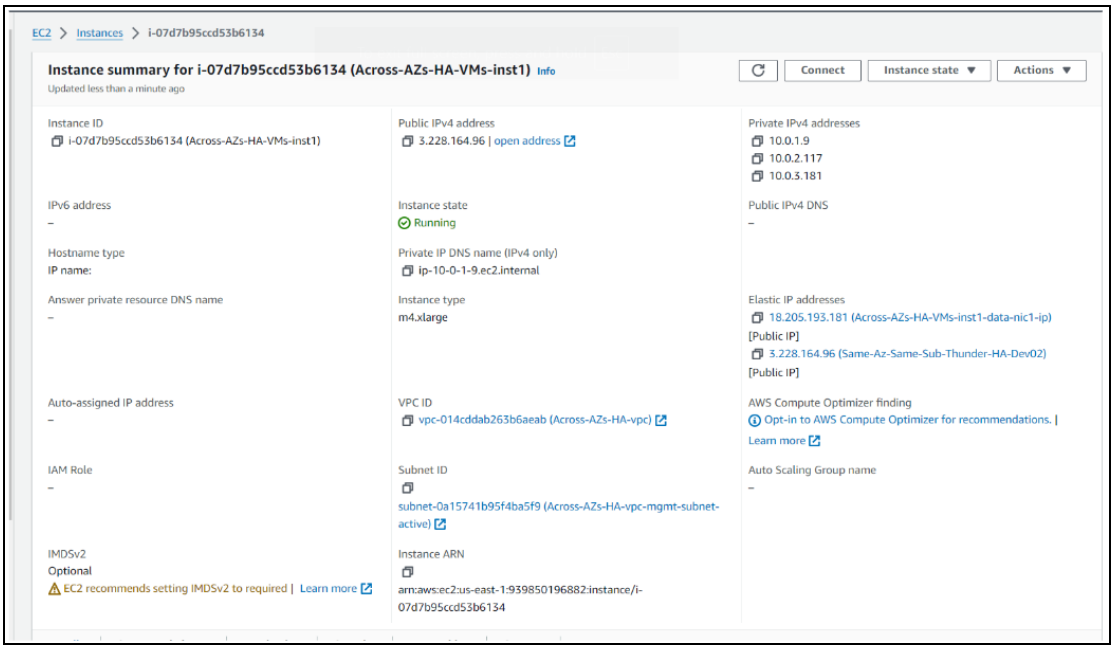
If the deployment is successful with basic SLB and HA Across AZs configuration, the following output is displayed:

```
!
!
system password-policy complexity Default username-check enable
system password-policy complexity Default repeat-character-check enable
system password-policy complexity Default forbid-consecutive-character
4
!
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 10.0.3.103
  floating-ip 10.0.2.121
  floating-ip 30.30.30.30
  floating-ip 20.20.20.20
```

```
  blade-parameters
    priority 100
!
vrrp-a peer-group
  peer 10.0.2.117
  peer 10.0.5.77
!
ip route 0.0.0.0 /0 10.0.1.1
!
ip route 10.0.8.0 /24 10.0.3.1
!
ip nat pool vrrp_0 10.0.3.103 10.0.3.103 netmask /32
!
ip nat pool vrrp_0_1 30.30.30.30 30.30.30.30 netmask /32
!
slb server ubuntu-backend-server 10.0.8.126
  health-check-disable
  port 80 tcp
    health-check-disable
  port 5004 tcp
    health-check-disable
!
slb service-group sg5004 tcp
  member ubuntu-backend-server 5004
!
slb service-group sg80 tcp
  member ubuntu-backend-server 80
!
slb virtual-server vip 10.0.2.121
  port 80 tcp
    source-nat pool vrrp_0
    service-group sg80
!
slb virtual-server vip1 20.20.20.20
  port 80 tcp
    ha-conn-mirror
    source-nat pool vrrp_0_1
```

```
    service-group sg80
!
cloud-services cloud-provider
  aws
    multi-az-failover
      vrid 0
        route-table-id rtb-0031384a1b3c85ece
        vip-interface-id eni-06bf9ec9dc4db072
        fip-interface-id eni-036149fa3aca5768e
        vip-dest 20.20.20.0/24
        fip-dest 30.30.30.0/24
        vip 0
          private-ip 10.0.2.121
          elastic-ip 34.193.11.224
!
end
vThunder-Active(config)#
```

Figure 22 : vThunder instance 1 - Active



3. Run the following command on vThunder instance 2:

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

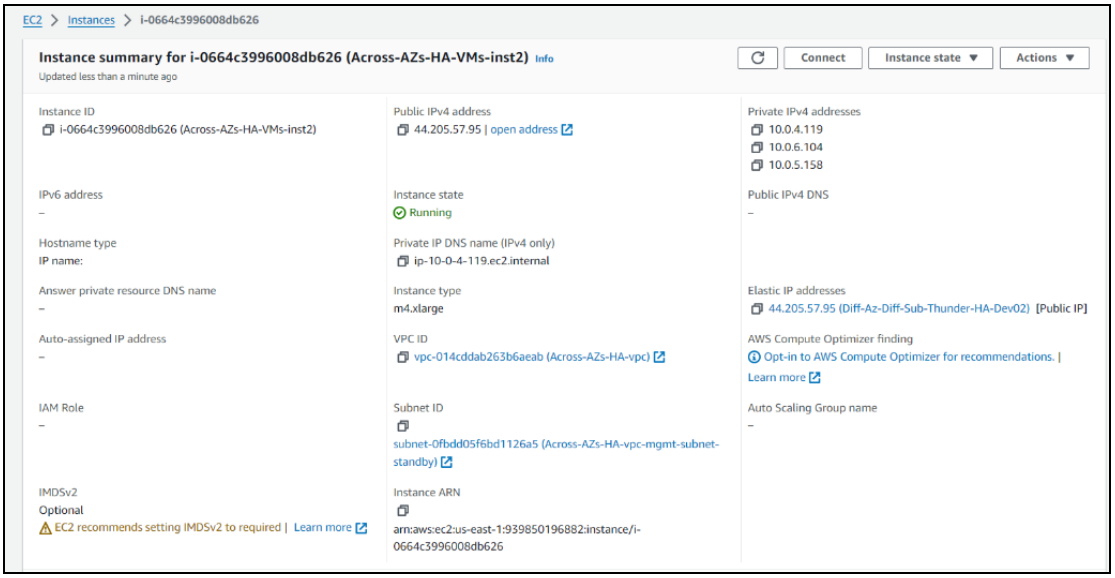
At this point, the vThunder instance 2 will also have the same configuration. If the deployment is successful with basic SLB and HA Across AZs configuration, the following output is displayed:

```
!
!
system password-policy complexity Default username-check enable
system password-policy complexity Default repeat-character-check enable
system password-policy complexity Default forbid-consecutive-character
4
!
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 10.0.6.30
  floating-ip 10.0.5.205
  floating-ip 30.30.30.30
  floating-ip 20.20.20.20
  blade-parameters
    priority 99
!
```

```
vrrp-a peer-group
  peer 10.0.2.117
  peer 10.0.5.77
!
ip route 0.0.0.0 /0 10.0.4.1
!
ip route 10.0.8.0 /24 10.0.6.1
!
ip nat pool vrrp_0 10.0.6.30 10.0.6.30 netmask /32
!
ip nat pool vrrp_0_1 30.30.30.30 30.30.30.30 netmask /32
!
slb server ubuntu-backend-server 10.0.8.126
  health-check-disable
  port 80 tcp
    health-check-disable
  port 5004 tcp
    health-check-disable
!
slb service-group sg5004 tcp
  member ubuntu-backend-server 5004
!
slb service-group sg80 tcp
  member ubuntu-backend-server 80
!
slb virtual-server vip 10.0.5.205
  port 80 tcp
    source-nat pool vrrp_0
    service-group sg80
!
slb virtual-server vip1 20.20.20.20
  port 80 tcp
    ha-conn-mirror
    source-nat pool vrrp_0_1
    service-group sg80
!
cloud-services cloud-provider
```

```
  aws
    multi-az-failover
      vrid 0
        route-table-id rtb-0031384a1b3c85ece
        vip-interface-id eni-07204a822af658600
        fip-interface-id eni-0f59395a28d1fb1c8
        vip-dest 20.20.20.0/24
        fip-dest 30.30.30.0/24
        vip 0
          private-ip 10.0.5.205
          elastic-ip 34.193.11.224
!
end
vThunder-Standby(config)#
```

Figure 23 : vThunder instance 2 - Standby

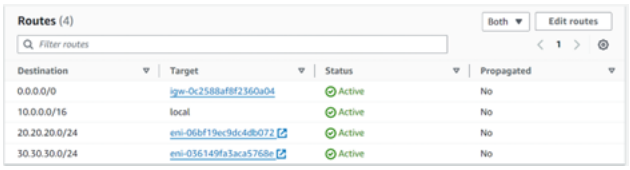


4. Run the following command on vThunder instance 1:

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

Before the failover, note the interface ID in the route table.

Figure 24 : Routes - Before Failover

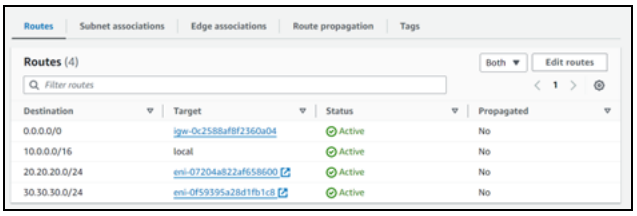


At this point, IP switching occurs and the vThunder instance 2 prompt becomes:

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

After successful failover, the interface ID is changed in the route table.

Figure 25 : Routes - After Failover



5. If you want to make vThunder instance 1 active, run the following command on vThunder instance 1:

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

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

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

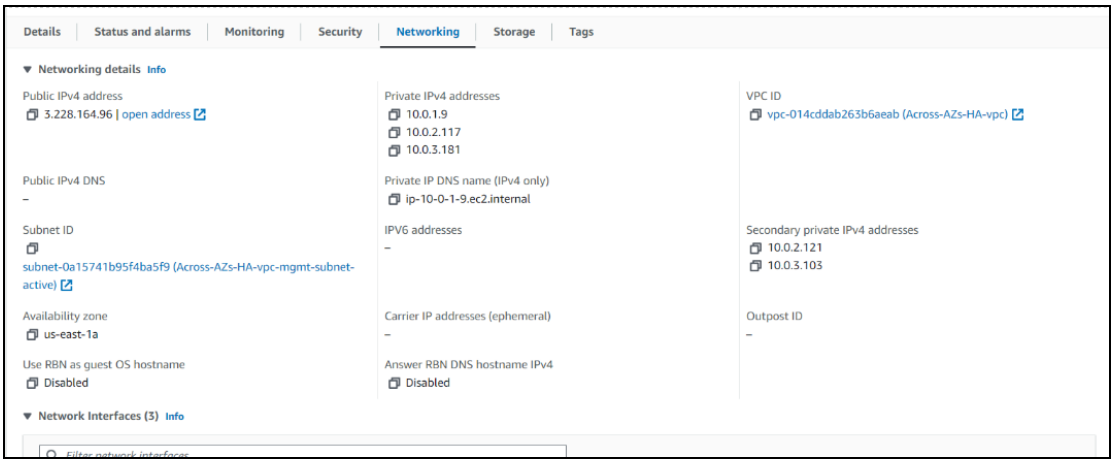
Verify Traffic Flow

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. Note down the Data-In Subnet VIP address of the active vThunder instance.

   Figure 26 : 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. Replace the Data-In Subnet VIP address of the active vThunder instance noted above in the following command and then run the command in the Terminal window to send the traffic from the client machine:

   ```
   curl <vThunder_instance_datain-nic_secondary_private_or_public_ip_vip>
   ```

   Example for Private VIP

   ```
   curl 10.0.2.121
   ```

   Example for Public VIP

   ```
   curl 3.214.248.169
   ```

   Example for Alien IP

   ```
   curl 20.20.20.20
   ```

8. Verify if a response is received.
9. SSH your client machine and run the following command to verify the HTTP

template traffic flow:

```
curl <vThunder_instance_datain-nic_secondary_private_or_public_ip_vip>:<port_number>/<host-match-string or url-match-string>/
```

Example for Private VIP

```
curl 10.0.2.121:80/s1/
```

Example for Public VIP

```
curl 3.214.248.169:80/s1/
```

Example for Alien IP

```
curl 20.20.20.20:80/s1/
```

Verify if a response is received from client server (For example: Apache Index page).

10. SSH your client machine and run the following commands to verify the Persist cookie template traffic flow:

    a. Verify the current cookie configuration:

    ```
    curl --head <vThunder_instance_datain-nic_private_or_public_ip_vip>
    ```

    b. Run the following commands to save the cookies in the **`cookie.txt`** file:

    ```
    curl -b cookie.txt -c cookie.txt <vThunder_instance_datain-nic_private_or_public_ip_vip>
    cat cookie.txt
    ```

    Example for Private VIP

    ```
    curl --head 10.0.2.121
    curl -b cookie.txt -c cookie.txt 10.0.2.121
    cat cookie.txt
    ```

    Example for Public VIP

    ```
    curl --head 3.214.248.169
    curl -b cookie.txt -c cookie.txt 3.214.248.169
    cat cookie.txt
    ```

    Example for Alien VIP

```
curl --head 20.20.20.20
curl -b cookie.txt -c cookie.txt 20.20.20.20
cat cookie.txt
```

11. Run the following command on the active vThunder instance to view the persistence load-balancing statistics:

```
vThunder(config)#show slb persist
```

If the deployment is successful, the following summary persistence statistics is displayed:

```
                                        Total
-----------------------------------------------
URL hash persist (pri)                    0
URL hash persist (sec)                    0
URL hash persist fail                     0
SRC IP persist ok                         0
SRC IP persist fail                       0
SRC IP hash persist(pri)                  0
SRC IP hash persist(sec)                  0
SRC IP hash persist fail                  0
DST IP persist ok                         0
DST IP persist fail                       0
DST IP hash persist(pri)                  0
DST IP hash persist(sec)                  0
DST IP hash persist fail                  0
SSL SID persist ok                        0
SSL SID persist fail                      0
Cookie persist ok                         1
Cookie persist fail                       0
Persist cookie not found                  2
Persist cookie Pass-thru                  0
Enforce higher priority                   0
```

If the Persist-cookie configuration is successful, a value is displayed for the **`Cookie persist ok`** else the value is 0.

Thunder-3NIC-3VM

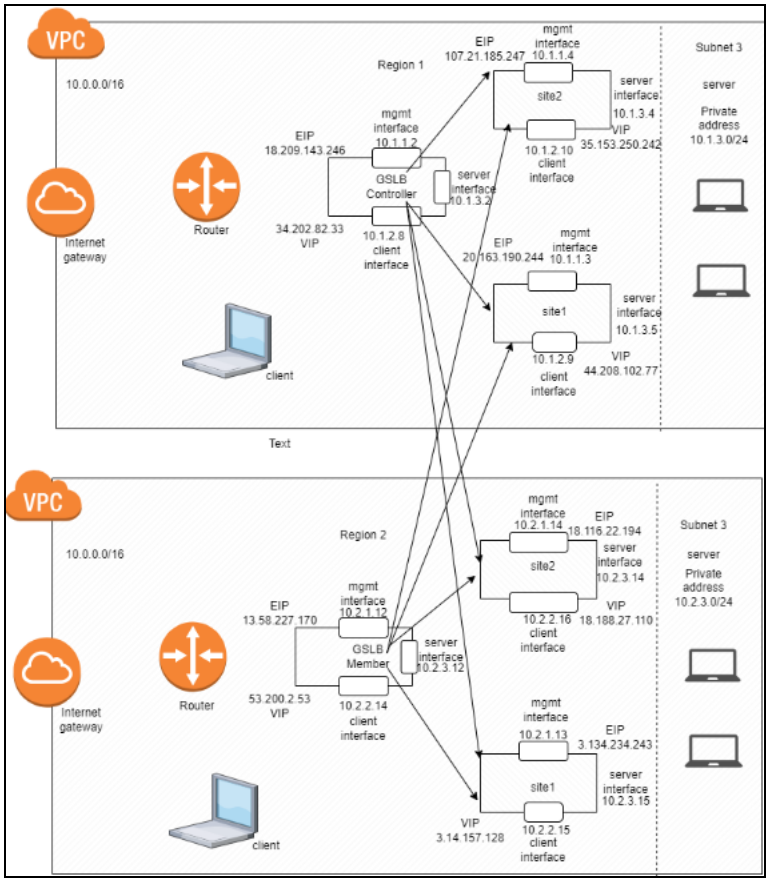
This template creates three new virtual machines with pre-loaded Thunder instance in the same region and zone and attaches three new network interface cards (NICs). These three vThunder instances are referred as Master Controller (Active), Site1 and Site2. Same template can be used to install identical number of resources in another region. The three vThunder instances in another region are referred as Member Controller, Site1 and Site2.

This template is used to setup disaster recovery site in a cross-region or hybrid cloud environment.

For more information, see Create Thunder Virtual Machines.

NOTE: Use a suitable VM size that supports at least three NICs. For VM sizes, see List of Supported Instance Types.

Figure 27 : Thunder ADC with GSLB



Additional Thunder configurations are available that can be applied as needed:

- Change Password
- A10 License
- SSL Certificate
- Hybrid Cloud GSLB

Various templates are available for different deployment needs.

For more information, see Deployment Templates.

The following topics are covered:

Create Thunder Virtual Machines 87
Configure Thunder 98
Verify Deployment 99
Verify Traffic Flow 121

Create Thunder Virtual Machines

The A10-vThunder-3NIC-3VM template is used to create three Thunder virtual machines with three network interface cards each.

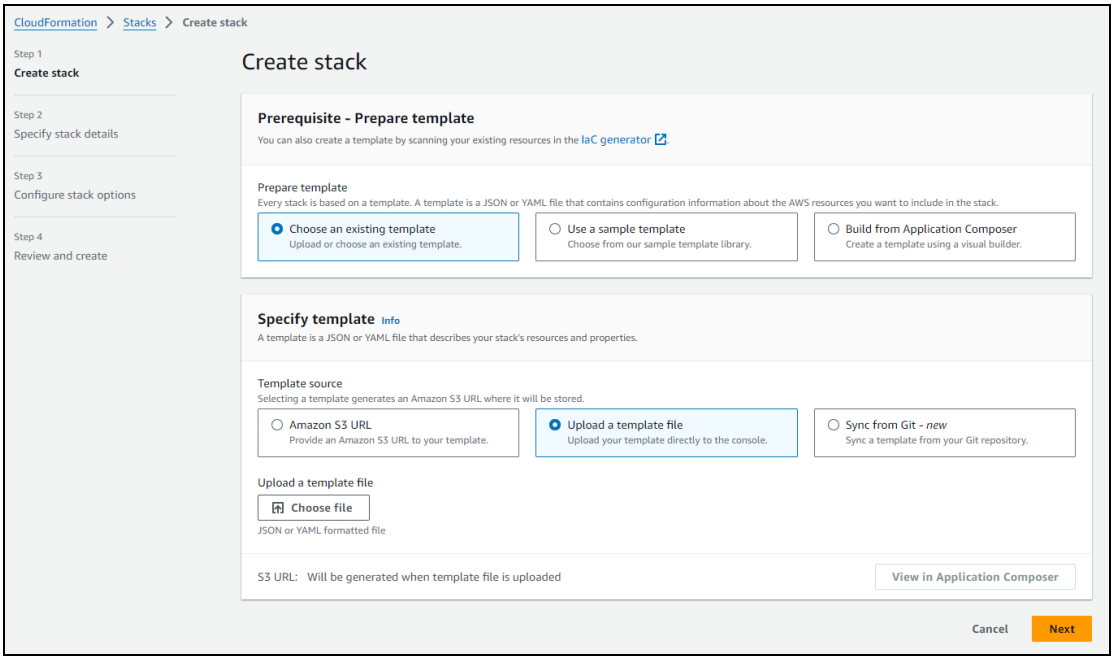
Before deploying this template, it is recommended to review the Prerequisites.

To create Thunder virtual machines using the A10-vThunder-3NIC-3VM template, perform the following steps:

1. Download **A10-vThunder_CFT-TEMPLATES** > **A10-vThunder-3NIC-3VM** template from GitHub.
2. From AWS Management Console, navigate to **CloudFormation** > **Stacks** > **Create Stack** > **With new resources (standard)**.

   The Create stack window is displayed.

Figure 28 : Create stack window



3. In the **Prerequisite - Prepare template** section, select **Choose an existing template**.

   After 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 the following template file from the downloaded CFT folder:
   CFT_TMPL_3NIC_3VM.json

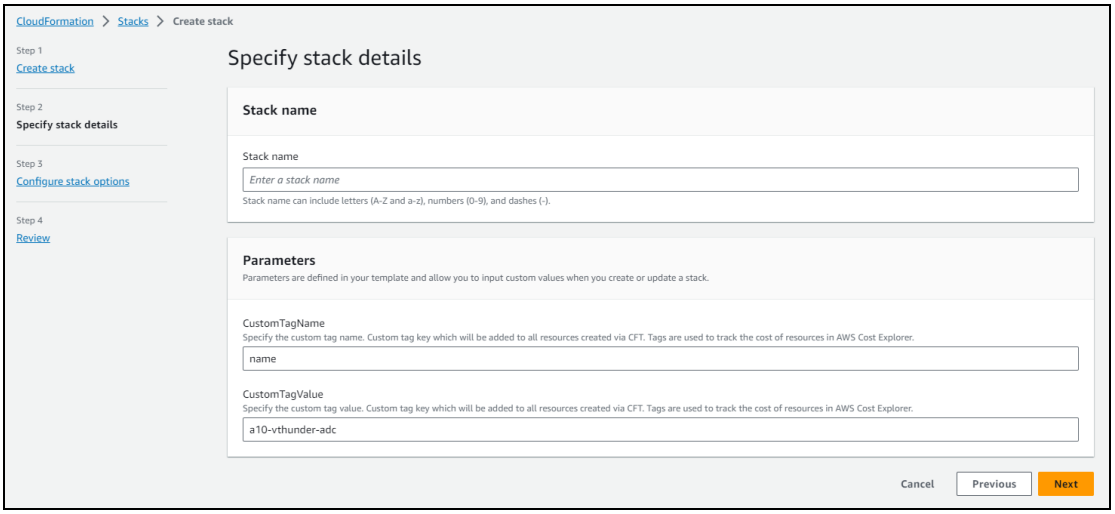
   The selected template file name is displayed as the chosen file.

   NOTE: This template contains pre-populated default values that can be modified as required. It does not create a new elastic public IP, primary private IP address, virtual private IP address, security groups, subnets, and virtual private network.

5. Click **Next**.

   The **Specify stack details** window is displayed.

Figure 29 : Specify stack details window



6. In the **Specify stack details** window, enter or select the following:

Table 6 : JSON Parameters

Resource Name	Description
Stack name	Specify a stack name containing letters (A-Z and a-z), numbers (0-9), and dashes (-). Here, the stack name is provided as `vth-rg1`.
AMIID	Specify the AMI ID of the required A10 vThunder image. By default, A10 Thunder ADC for Advanced Load Balancing - BYOL AMI ID is set for us-east-1 region. To get the AMI ID, go to the **AWS Management Console** > **EC2** > **Launch instance** > **Application and OS Images (Amazon**

Table 6 : JSON Parameters

Resource Name	Description
	Machine Image) > search for the relevant vThunder image and then copy the AMI ID. For more information on the available Thunder images with different capacities under respective regions, see List of ACOS AMI ID.
AllocationIdEIPThunderInstance11	Specify an allocation ID of Thunder instance 1's elastic IP address. This elastic IP address is the management public IP address of the Thunder instance 1. To get the allocation ID, go to the **AWS Management Console** > **EC2** > **Elastic IPs** > *<elastic_IP_address>* > **Summary** and then copy the **Allocation ID**.
AllocationIdEIPThunderInstance12	Specify an allocation ID of Thunder instance 1's elastic IP address. This elastic IP address is the virtual public IP address of the Thunder instance 1.

Table 6 : JSON Parameters

Resource Name	Description
	To get the allocation ID, go to the **AWS Management Console** > **EC2** > **Elastic IPs** > <*elastic_IP_address*> > **Summary** and then copy the **Allocation ID**.
AllocationIdEIPThunderInstance21	Specify an allocation ID of Thunder instance 2's elastic IP address. This elastic IP address is the management public IP address of the Thunder instance 2. To get the allocation ID, go to the **AWS Management Console** > **EC2** > **Elastic IPs** > <*elastic_IP_address*> > **Summary** and then copy the **Allocation ID**.
AllocationIdEIPThunderInstance22	Specify an allocation ID of Thunder instance 2's elastic IP address. This elastic IP address is the virtual public IP (VIP) address of the Thunder instance 2. To get the allocation ID, go to the **AWS Management Console** > **EC2** > **Elastic IPs** >

Table 6 : JSON Parameters

Resource Name	Description
	<*elastic_IP_address*> > **Summary** and then copy the **Allocation ID**.
AllocationIdEIPThunderInstance31	Specify an allocation ID of Thunder instance 3's elastic IP address. This elastic IP address is the management public IP address of the Thunder instance 3. To get the allocation ID, go to the **AWS Management Console** > **EC2** > **Elastic IPs** > <*elastic_IP_address*> > **Summary** and then copy the **Allocation ID**.
AllocationIdEIPThunderInstance32	Specify an allocation ID of Thunder instance 3's elastic IP address. This elastic IP address is the virtual public IP (VIP) address of the Thunder instance 3. To get the allocation ID, go to the **AWS Management Console** > **EC2** > **Elastic IPs** > <*elastic_IP_address*> > **Summary** and then copy the **Allocation ID**.

Table 6 : JSON Parameters

Resource Name	Description
CustomTagName	Specify the custom tag name. The custom tag key is added to all the resources created using the CFT template. This tag is used to track the cost of resources in the AWS Cost Explorer.
CustomTagValue	Specify the custom tag value. The custom tag key is added to all the resources created using the CFT template. This tag is used to track the cost of resources in the AWS Cost Explorer. The default custom tag value is **a10-vthunder-adc**.
EC2Instance	Select an instance/compute type supported for vThunder from the available list. By default, instance type is m4.xlarge which contains 4 vCPU and 16 GiB memory. The minimum requirement for vThunder is 4 vCPU and 16 GiB memory. For more information on product pricing, see AWS Marketplace.

Table 6 : JSON Parameters

Resource Name	Description
	If the required instance type is not available in the list, go to **AWS Management Console** > **EC2** > **Launch instance** > Find and copy the relevant instance type in the template. For more information, see Supported Instance Types.
IPThunder1	Specify the primary private IP of Thunder instance 1's data-in subnet.
IPThunder2	Specify the primary private IP of Thunder instance 2's data-in subnet.
IPThunder3	Specify the primary private IP of Thunder instance 3's data-in subnet.
KeyPair	Select a keypair from the available list. A key pair is a set of security credentials which consist of a public key and a private key. This keypair is used to verify your identity when connecting to an Amazon EC2 instance.
SecurityGroupData	Select the required

Table 6 : JSON Parameters

Resource Name	Description
	security group ID from the available list for data interface within the selected Virtual Private Network.
SecurityGroupMgmt	Select the required security group ID from the available list for the management interface within the selected Virtual Private Network.
SubnetDataINID	Select the required subnet ID from the available list for the data traffic flow inward to vThunder within the selected Virtual Private Network.
SubnetDataOutID	Select the required subnet ID from the available list for the data traffic flow outward from vThunder within the selected Virtual Private Network.
SubnetMgmtID	Select the required subnet ID from the available list for managing vThunder within the selected Virtual Private Network.
Tenancy	Select the required tenancy from the available list. Tenancy defines the distribution of EC2 instances across the

Table 6 : JSON Parameters

Resource Name	Description
	physical hardware that affects pricing. It specifies if the Amazon EC2 instance is hosted on shared or single-tenant hardware.
VIPThunder1	Specify the virtual private IP of Thunder instance 1's data-in subnet.
VIPThunder2	Specify the virtual private IP of Thunder instance 2's data-in subnet.
VIPThunder3	Specify the virtual private IP of Thunder instance 3's data-in subnet.
VPCID	Select the required Virtual Private Network ID to configure vThunder.
Zone	Select a zone from the existing availability zones. AWS offers a range of options for managing availability and resiliency for your applications.

Table 6 : JSON Parameters

Resource Name	Description
	NOTE: Make use of the replicated VMs across the availability zones to protect your applications and data against the data center outages and maintenance events.

7. Click **Next**.

   The **Configure stack options** window is displayed.

8. Verify the other fields and change the values appropriately. (Optional)
9. Click **Next**.

   The **Review** <*stack_name*> window is displayed.

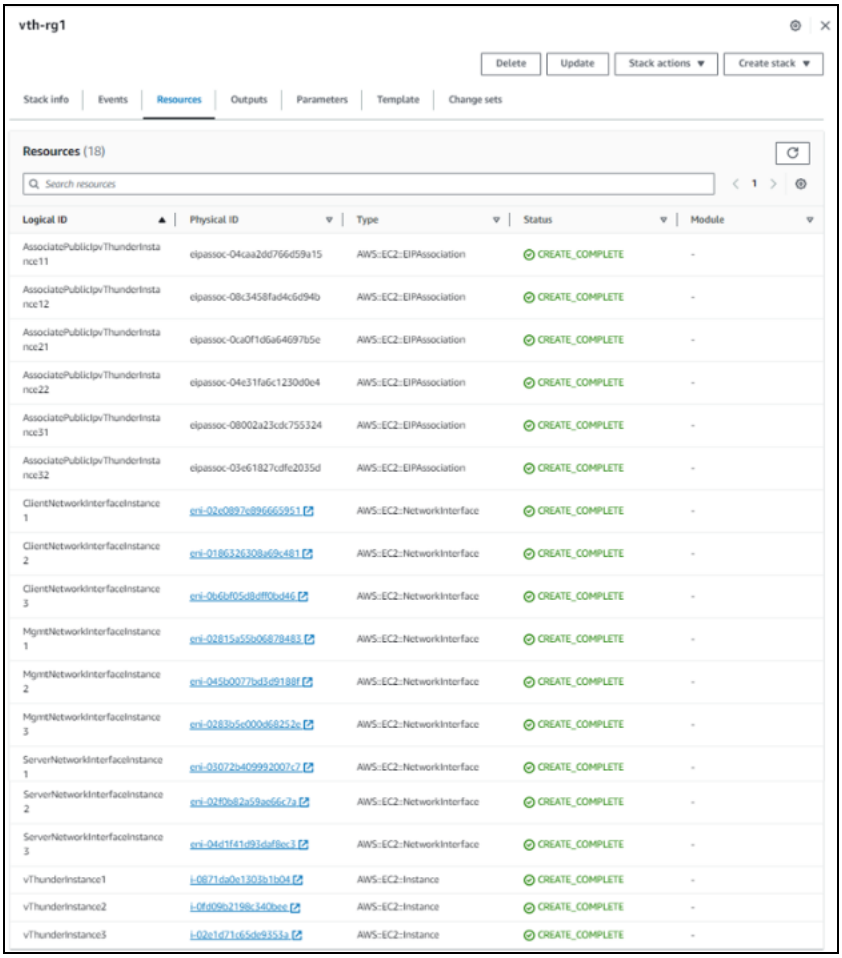
10. Verify if all the stack configurations are correct and then click **Submit**.

    NOTE: The system may take a few minutes to create the resources and display the stack status as **CREATE_COMPLETE**.

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

    - Three vThunder instances
    - One management and two data interface for each vThunder instance

Figure 30 : Resource listing



12. Access Thunder Virtual Machine.
13. Create Server Machine.
14. Create Client Machine.

Configure Thunder

The following configurations can be applied to the deployed vThunder instance depending on your use case and requirement, see Deployment Templates:

- Change Password
- A10 License

- SSL Certificate
- Hybrid Cloud GSLB

Verify Deployment

To verify deployment using CFT, perform the following steps:

1. Access the vThunder instances using CLI.
2. 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 server1 10.1.3.52
  health-check-disable
  port 80 tcp
    health-check-disable
!
slb service-group sg tcp
  member 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 server2 10.1.3.250
  health-check-disable
  port 80 tcp
    health-check-disable
!
slb service-group sg tcp
  member 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 server3 10.2.3.179
  health-check-disable
  port 80 tcp
    health-check-disable
!
slb service-group sg tcp
  member 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 server4 10.2.3.254
  health-check-disable
  port 80 tcp
    health-check-disable
!
slb service-group sg tcp
  member 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
```

3. 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
```

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
```

4. 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.
```

NOTE: If you want to access the vThunder instances using GUI, re-login the instances using the new admin user password and NOT the *EC2 Instance ID*.

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 31 : 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       35.153.250.242
www.gslb.a10.com.       10      IN      A       44.208.102.77
www.gslb.a10.com.       10      IN      A       18.188.27.110
www.gslb.a10.com.       10      IN      A       3.14.157.128

;; 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
```

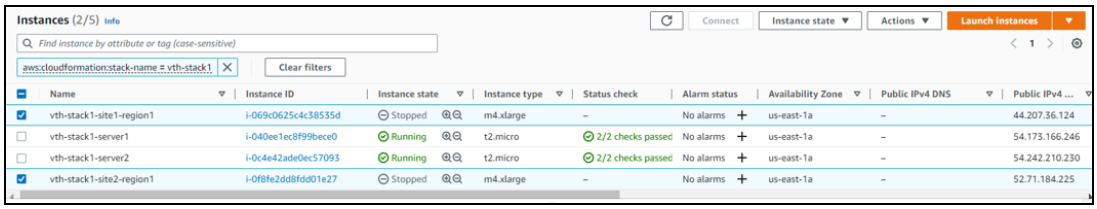
2. Perform the DNS lookup again.

```
$ 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       44.208.102.77
www.gslb.a10.com.       10      IN      A       18.188.27.110
www.gslb.a10.com.       10      IN      A       3.14.157.128
www.gslb.a10.com.       10      IN      A       35.153.250.242

;; 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 32 : 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 instances 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
```

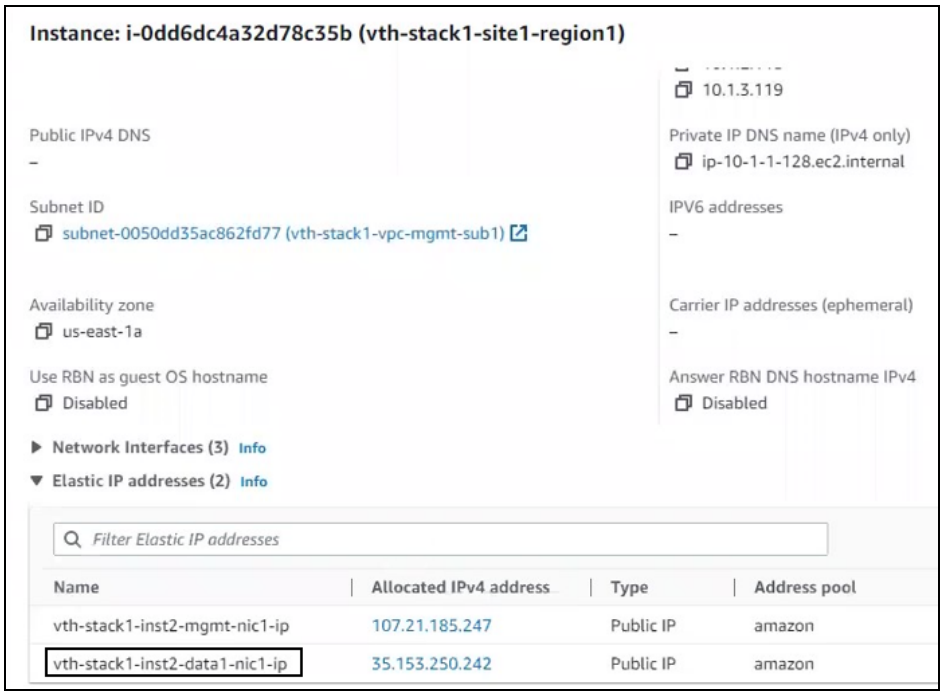
   While the Apache server is getting installed, you get a prompt to continue further. Enter 'Y' to continue. After the installation is complete, a newline prompt is displayed.

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 33 : 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
[10671/10671]
```

ADC Configuration Templates

This section guides you in applying new Application Delivery Controller (ADC) configurations on Thunder using Python scripts.

Table 7 provides an overview of the different supported Thunder configurations. These configurations are optional. You can choose to apply them based on your specific use cases. For more information, see Deployment Templates.

Table 7 : Supported Thunder configurations

Configuration	Description
Change Password	Applies a new vThunder instance password. **NOTE:** After the deployment of vThunder instance, it is highly recommended to change the default password for `admin` user.
Basic Server Load Balancer	Applies an SLB configuration for inbound traffic, outbound traffic, policies, server grouping, and routing to destination virtual servers.
Backend Server Autoscaling	Applies an SLB configuration automatically whenever backend app/web servers are autoscaled. When the backend web/app servers are in an autoscale group within the AWS cloud,

Table 7 : Supported Thunder configurations

Configuration	Description
	autoscale-in or autoscale-out of the server triggers a Lambda function for applying or removing SLB server configuration in Thunder.
A10 License	Applies an A10 license to the vThunder instance. **NOTE:** A10 Thunder is a proprietary software that requires either a trial or BYOL (Bring Your Own License) subscription. However, pre-licensed subscription-based images from the AWS Marketplace do not require this configuration.
SSL Certificate	Applies a server connection certificate configurations. A Secure Sockets Layer (SSL) certificate is a digital certificate that verifies a website's identity and facilitates an encrypted connection. SSL is a security protocol that establishes an encrypted link between a web server

Table 7 : Supported Thunder configurations

Configuration	Description
	and a web browser.
High Availability	Applies high availability configuration. This configuration automatically synchronizes Thunder configuration between the active and standby Thunder instances. In the event of a failover, the other standby Thunder instance is designated as active to ensure uninterrupted traffic routing. For this functionality, it is essential for both Thunder instances to have identical resources and configurations.
High Availability Across Availability Zones	Applies high availability configuration. In this configuration, both vThunder instances are either in same availability zone having six subnets for each interface or across different availability zones have 3 subnets each in same region. This configuration automatically synchronizes vThunder configuration between the active and standby vThunder instances. In the event of a failover, the other standby

Table 7 : Supported Thunder configurations

Configuration	Description
	vThunder instance is designated as active to ensure uninterrupted traffic routing. For this functionality, it is essential for both vThunder instances to present in same VPC.
Autoscale Server	Creates Autoscaling Group (ASG) and Lambda Function. It configures S3 Bucket to store the template resources.
Hybrid Cloud GSLB	Applies a disaster recovery configuration using a global server load balancer across any two regions or locations, whether within the same cloud, hybrid-cloud or on-premise. It requires a minimum of two Thunder instances in each region or location —one acting as the master controller and the other as a site device. Multiple site devices can be configured but it is recommended to have a minimum of three site devices for seamless failover and effective disaster recovery. For a configuration with

Table 7 : Supported Thunder configurations

Configuration	Description
	three Thunders instances, the recommended setup includes one as the master controller and the other two as site devices. Ensure that both regions have identical set of resources. To create and install three thunder instances in one region use Thunder-3NIC-3VM template. The same template can be used to install Thunder instances in another region.

Change Password

After provisioning the vThunder instance, you can change the vThunder instance password at any given time.

NOTE: It is highly recommended to change the default password. For default password, see Support Information.

To change the password of the deployed vThunder instance, perform the following steps:

1. Download **A10-vThunder_ADC-CONFIGURATION** > **CHANGE-PASSWORD** template from GitHub.
2. From the Start menu, open command prompt and navigate to the downloaded template.

3. Open the CHANGE_PASSWORD_PARAM with a text editor.

NOTE: Each parameter has a default value mentioned in the parameter file, which can be modified as required.

4. Configure the following parameters:

Table 8 : JSON Parameters

Resource Name	Description
Public IP List	Specify the Public IP address of one or more vThunder instance/s of vThunders to change password. `"publicIpList": ["X.X.X.X","X.X.X.X"],`
Secret manager name	Specify the Secret manager name for Autoscaling to add servers only. `"secretManagerName": "secret-manager-name"`

5. From the Start menu, open the command prompt and navigate to the downloaded template to run the following command:

```
C:\Users\TestUser\A10-vThunder_ADC-CONFIGURATION\CHANGE-PASSWORD>
python ./CHANGE_PASSWORD.py
```

A message prompt displaying the primary conditions for password validation appears:

```
Primary conditions for password validation, user should provide the new
password according to the given combination:

Minimum length of 9 characters
Minimum lowercase character should be 1
Minimum uppercase character should be 1
Minimum number should be 1
Minimum special character should be 1
Should not include repeated characters
Should not include more than 3 keyboard consecutive characters.
```

6. Provide the vThunder instance's default password as the EC2 instance ID, new

password, and then confirm the new password when prompted:

```
Enter vThunder [x.x.x.x] password: ***
Enter vThunder new password: *********
Confirm new password: *********
```

NOTE: The default password is provided by the A10 Networks Support. The new password should meet the default password policy criteria. For more information, see Default Password Policy.

7. If the password is successfully updated in Secret Manager, the following message is displayed:

```
vThunder [x.x.x.x] Password changed successfully.
Configurations are saved on partition: shared
```

8. Enter 'yes' to update the password in the Secret Manager.

```
Do you want to update password in Secret Manager? [yes/no]yes
```

9. If the password is changed successfully, the following message is displayed:

```
Successfully updated password in Secret Manager.
```

Basic Server Load Balancer

This template configures the vThunder instance as a Server Load Balancer (SLB) to evenly distribute the traffic across a set of predefined servers and requires manual scaling.

To configure vThunder as an SLB, perform the following steps:

1. Download **A10-vThunder_ADC-CONFIGURATION** > **BASIC-SLB** template from GitHub.
2. From the Start menu, open the command prompt and navigate to this downloaded template.
3. Open the SLB_CONFIG_PARAM.json with a text editor.

NOTE: Each parameter has a default value mentioned in the parameter file, which can be modified as required.

4. Configure the following parameters:

Table 9 : JSON Parameters

Resource Name	Description
Template HTTP	Specify the value as 1 if you want to configure the HTTP template. For more information on SLB HTTP template, see *Command Line Interface Reference*. `"template_http": 0,` **NOTE:** By default, the template HTTP value is 0.
Template Persist Cookie	Specify the value as 1 if you want to configure the Persist-Cookie template. For more information on SLB persist cookie template, see *Command Line Interface Reference*. `"template_persist_cookie": 0,` **NOTE:** By default, the template Persist-Cookie value is 0.
vThunder instance username	Specify a 'Read/Write/HM' privilege username. `"vth_username": "admin",` **NOTE:** The vThunder instance user should have 'Read/Write/HM' privilege to configure vThunder as an SLB.
Data Interface Count	Specify the number of data NICs. The value should be '1' for 2 NICs and '2' for 3 NICs. `"data_interface_count":1,`
Public IP addresses	Specify the Public IP address of one or more vThunder instance/s depending on the deployed template. `"publicIpList": ["X.X.X.X","X.X.X.X"],`
SLB	Specify name and private IP address of one or more SLB servers.

Table 9 : JSON Parameters

Resource Name	Description

server details	<pre>"server_details": {
	"value": [
	
	{
	"server-name": "server1",
	"pvt-ip-address": "X.X.X.X"
	},
	{
	"server-name": "server2",
	"pvt-ip-address": "X.X.X.X"
	}
	]},</pre>

SLB server ports	Specify the SLB Server ports details.
	<pre>"slbServerPortList": {
	"value": [
	{
	"port-number": 53,
	"protocol": "udp"
	},
	{
	"port-number": 80,
	"protocol": "tcp"
	},
	{
	"port-number": 443,
	"protocol": "tcp"
	}
	]
	},</pre>

Persist Cookie Template	Specify the Persist Cookies template details if **templatePersistCookie = 1.**

Table 9 : JSON Parameters

Resource Name	Description

	<pre>"cookie-list":{
	"value": [
	{
	"name": "persist-cookie",
	"expire": 60,
	"encrypt-level": 0,
	"cookie-name":"a10-cookies",
	"match-type": 1,
	"service-group":1
	}
	]},</pre>

HTTP Template	Specify the HTTP template details if **`templateHTTP = 1`**.

Table 9 : JSON Parameters

Resource Name	Description

	<pre>"httpList": {
	  "value": [
	  {
	  "name":"host-switch",
	  "host-switching": [
	  {
	  "host-switching-type":"contains",
	  "host-match-string":"s1",
	  "host-service-group":"sg80"
	  }
	  ]
	  },
	  {
	  "name":"url-switch",
	  "url-switching": [
	  {
	  "url-switching-type":"regex-match",
	  "url-match-string":"s1",
	  "url-service-group":"sg80"
	  }
	  ]
	  }
	  ]
	},</pre>

Virtual Server	Specify virtual server details. The virtual server default name is “vip”. This is the private primary (for 2NIC-1VM) and secondary (for 3NIC-2VM) IP address of Ethernet1. **NOTE:** To get the vip address after deploying the vThunder instances, **EC2** > **Instances** > <*stack_name_inst1*> > **Networking**.

Table 9 : JSON Parameters

Resource Name	Description
	If you want to configure an HTTP template (`templateHTTP = 1`), provide the HTTP template name in the `template-http` parameter. If you want to configure a Persist-Cookie template (`templatePersistCookie = 1`), provide the Persist-Cookie template name in the `template-persist-cookie` parameter.

Table 9 : JSON Parameters

Resource Name	Description

```
      "virtual_Server_List": {
        "virtual-server-name": "vip",
        "eth1-ip-address": "X.X.X.X",
        "metadata": {
          "description": "Specify ethernet 1 primary
private IP address in case of SLB without High
Availability and secondary private IP address in case of
SLB with High Availability."
        },
        "value": [
          {
            "port-number": 53,
            "protocol": "udp",
            "auto": 1,
            "service-group": "sg53"
          },
          {
            "port-number": 80,
            "protocol": "http",
            "auto": 1,
            "service-group": "sg80",
            "template-http": "url-switch",
             "template-persist-cookie": "persist-cookie"
          },
          {
            "port-number": 443,
            "protocol": "https",
            "auto": 1,
            "service-group": "sg443",
            "template-http": "url-switch",
            "template-persist-cookie": "persist-cookie"
          }
        ]
      },
```

Table 9 : JSON Parameters

Resource Name	Description
	NOTE: Either **host-switching-template-name** or **url-switching-template-name** can be used in the **template-http**.
Service Group List	Specify the SLB Service group. `"serviceGroupList": { "value": [ { "name": "sg443", "protocol": "tcp" }, { "name": "sg53", "protocol": "udp" }, { "name": "sg80", "protocol": "tcp" } ] } }}` **NOTE:** if you want to configure service groups for each server, the number of members should be equal to the total number of servers. For example, if two servers are added in the parameter file then the members in the member list should also be two, one for server1 and the other for server2.

5. Verify if all the configurations in the SLB_CONFIG_PARAM.json file are correct and save the changes.
6. Run the following command to configure the vThunder instance/s as an SLB:

```
C:\Users\TestUser\A10-vThunder_ADC-CONFIGURATION\BASIC-SLB> python
./SLB_CONFIG.py
```

7. Provide password for the vThunder instances/s whose IP address is mentioned in the SLB_CONFIG_PARAM.json file.
   If SLB is configured successfully for 2NIC-1VM, the following message is displayed:

```
Enter vThunder [x.x.x.x] Password: *********
Successfully configuring ethernet ip 1
Do you want to configure SLB Server? [yes/no]yes
Successfully configured server server1
Successfully configured service group.
Successfully configured http template.
Successfully configured slb persist cookie.
Successfully configured virtual servers.
Configurations are saved on partition: shared
Successfully logged out from vThunder.
------------------------------------------------------------------
```

   The above configuration has one server, an HTTP template, and a Persist-Cookie template configured for 2NIC-1VM.

   If SLB is configured successfully for 3NIC-2VM, the following message is displayed:

```
Enter vThunder [x.x.x.x] Password: *********
Successfully configuring ethernet ip 1
Successfully configuring ethernet ip 2
Do you want to configure SLB Server? [yes/no]yes
Successfully configured server server1
Successfully configured server server2
Successfully configured service group.
Successfully configured http template.
Successfully configured slb persist cookie.
Successfully configured virtual servers.
Configurations are saved on partition: shared
Successfully logged out from vThunder.
-----------------------------------------------------------------
Enter vThunder [x.x.x.x] Password: *********
Successfully configuring ethernet ip 1
Successfully configuring ethernet ip 2
Do you want to configure SLB Server? [yes/no]yes
Successfully configured server server1
Successfully configured server server2
Successfully configured service group.
Successfully configured http template.
Successfully configured slb persist cookie.
Successfully configured virtual servers.
Configurations are saved on partition: shared
Successfully logged out from vThunder.
-----------------------------------------------------------------
```

The above configuration has two servers, an HTTP template, and a Persist-Cookie template configured for 3NIC-2VM.

8. In case of only basic SLB configuration without High availability, run the following commands on both the Thunder instances:

```
vth-inst1#config
vth-inst1(config)# ip route 0.0.0.0 /0 10.0.2.1
vth-inst1(config)# write memory
Building configuration...
Write configuration to default primary startup-config
[OK]
vth-inst1(config)#
```

Backend Server Autoscaling

This template configures vThunder instance as a Server Load Balancer (SLB) to automate the scaling process allowing dynamic adjustment of servers based on the workload.

NOTE: Configure vThunder with basic SLB before configuring SLB on Backend Autoscale.

To configure a Backend Autoscale Server, perform the following steps:

1. Create S3 Bucket
2. Create AWS Secrets Manager secret
3. Create vThunder credentials secret
4. Create Lambda Function and Autoscaling Group
5. Configure Lambda Function and ASG

Create S3 Bucket

Amazon Web Services' (AWS) Simple Storage Service (S3) bucket is used to store the template resources. Therefore, you need to create the S3 bucket or you can use an existing one and then upload the SLB configuration file on AWS using this S3 bucket.

To create an S3 bucket, perform the following steps:

1. Download **A10-vThunder_ADC-CONFIGURATION** > **CONFIG-SLB_ON_BACKEND-AUTOSCALE** template from GitHub.
2. Run the following command to create an S3 bucket and store the Lambda function python script:

```
PS C:\Users\TestUser\A10-vThunder_ADC-CONFIGURATION\CONFIG-SLB_ON_
BACKEND-AUTOSCALE> python ./AUTOSCALE_SERVER_S3_UPLOAD_1.py
```

3. Enter a unique bucket name and the region where you want to create the new bucket.

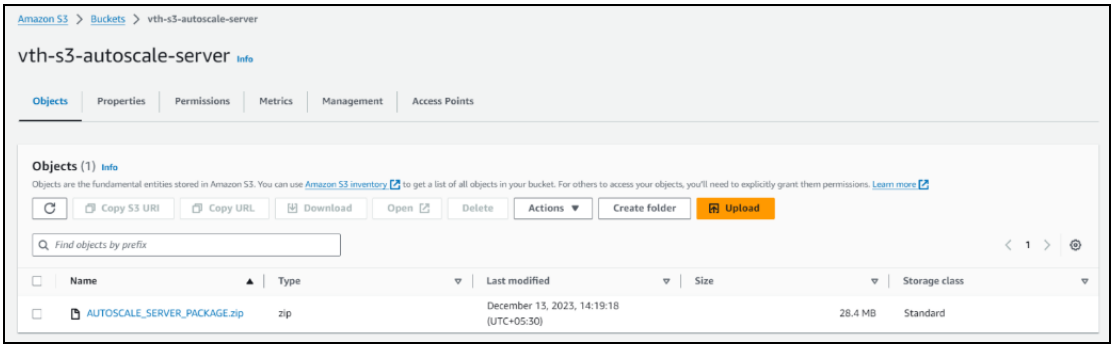
NOTE: The region for S3 bucket and ASG should be the same.

```
Enter bucket name: <bucket_name>
Enter region: <region>
```

If S3 bucket is created successfully, a message 'File uploaded in S3 bucket successfully' is displayed.

4. Verify if the S3 bucket is created in the **AWS Management Console** > **Buckets** > <*bucket_name*>.

Figure 34 : S3 Bucket



The AUTOSCALE_SERVER_PACKAGE.zip is uploaded.

If you are using an existing S3 bucket, the zip file should be uploaded in this bucket.

NOTE: To delete the S3 bucket, see Delete the resources.

Create AWS Secrets Manager secret

To create AWS Secrets Manager secret, if not created already, go to **AWS Management Console** > **Secrets Manager** > **Store a new secret** and add the following:

Table 10 : AWS Key value pair

Key	Value
Secret Type	Other type of secret
Key/value pairs	Provide the following values: • 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

Create vThunder credentials secret

To create vThunder credentials secret, if not created already, go to **AWS Management Console** > **Secrets Manager** > **Store a new secret** and add the following:

Table 11 : vThunder Key value pair

Key	Value
Secret Type	Other type of secret
Key/value pairs	Provide the following values: • username - *<vThunder username>* • password - *<vThunder password>*
Encryption key	aws/secretsmanager

Table 11 : vThunder Key value pair

Key	Value
Secret name	*<your secret manager name>*

For more information, see https://docs.aws.amazon.com/secretsmanager/latest/userguide/create_secret.html

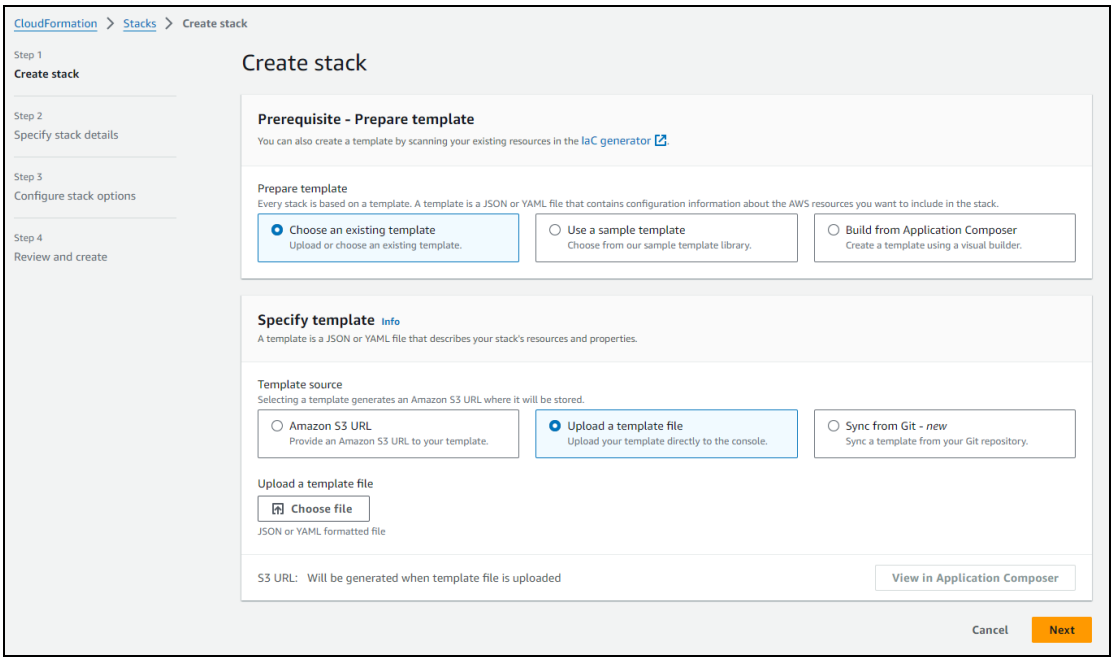
Create Lambda Function and Autoscaling Group

To create Lambda Function and Autoscaling Group, 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 **Choose an existing template**.

   After 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 template:
   AUTOSCALE_SERVER_PARAM.json

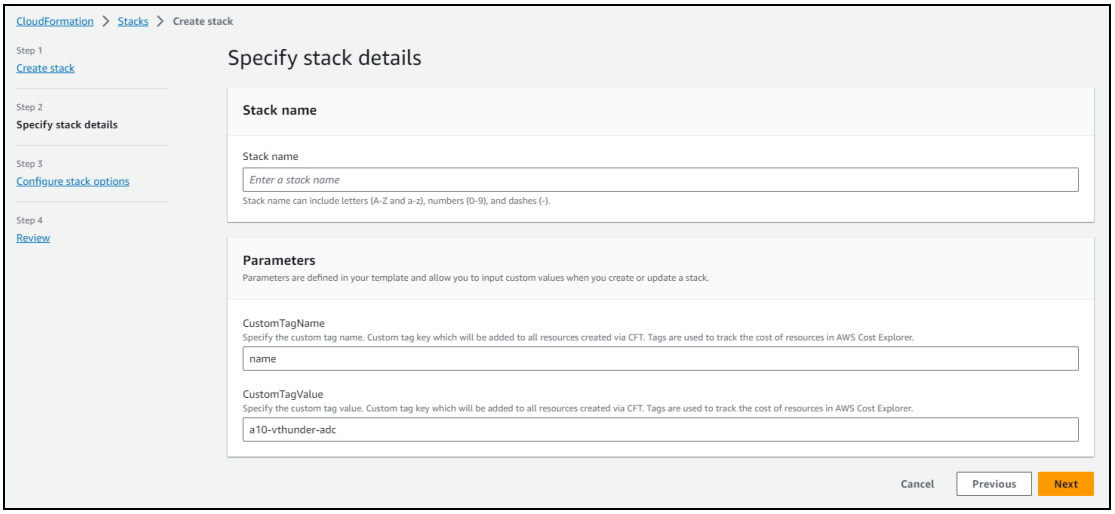
   The selected template file name is displayed as the chosen file.

NOTE:	This template contains pre-populated default values that can be modified as required and do not create a new subnet.

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:

   Table 12 : JSON Parameters

Resource Name	Description
Stack name	Specify a stack name containing letters (A-Z and a-z), numbers (0-9), and dashes (-). Here, the stack name is provided as `asg`.
AMIID	Specify the AMI ID of the required A10 vThunder image. By default, A10 Thunder ADC for Advanced Load Balancing - BYOL

Table 12 : JSON Parameters

Resource Name	Description
	AMI ID is set for us-east-1 region. To get the AMI ID, go to the **AWS Management Console** > **EC2** > **Launch instance** > **Application and OS Images (Amazon Machine Image)** > search for the relevant vThunder image and then copy the AMI ID. For more information on the available Thunder images with different capacities under respective regions, see List of ACOS AMI ID.
AWSSecretManager	Specify the existing secret manager name containing AWS access and secret access keys. AWS Secret Manager helps you to securely encrypt, store and retrieve credentials for your databases and other services.
CPUPolicyTargetValue	Specify the value to auto scale based on a target value of CPU utilization. For more information, see here.
KeyPair	Select a keypair from the available list. A key pair is a set of security credentials which consists of a public key and a private key. This keypair is used to verify your identity when connecting to an Amazon EC2 instance.
LaunchTemplateVersionNumber	Specify the auto scale group launch template version.
MgmtIPAddress	Specify the comma separated vThunder public management IP addresses.
Role	Specify the existing lambda function IAM Role ARN with lambda function permissions.

Table 12 : JSON Parameters

Resource Name	Description
	To get the value, go to **IAM Dashboard** > **Roles** and then click on **Iam Role** for which you need the ARN.
S3Bucket	Specify the existing S3 bucket name, where the AUTOSCALE_SERVER_PACKAGE.zip is uploaded.
SecurityGroupData	Select the required security group ID from the available list for data interface within the selected Virtual Private Network.
Size	Select an instance/compute type supported for Linux from the available list. A web server is configured on this Linux instance to test traffic. By default, instance type is t2.micro which contains 1 vCPU and 1 GiB memory. The minimum requirement for vThunder is 4 vCPU and 32 GiB memory. For more information on product pricing, see AWS Marketplace. If the required instance type is not available in the list, go to **AWS Management Console** > **EC2** > **Launch instance** > Find and copy the relevant instance type in the template. For more information, see Supported Instance Types.
SubnetDataID	Select the required subnet ID from the available list for the data traffic flow inward and outward to vThunder within the selected Virtual Private Network.
Zone	Select a zone from the existing availability zones. AWS offers a range of options for managing availability and resiliency for your applications.

Table 12 : JSON Parameters

Resource Name	Description
	NOTE: Make use of the replicated VMs across the availability zones to protect your applications and data against the data center outages and maintenance events.
vThunderSecretManager	Specify the existing secret manager name containing vThunder password. AWS Secret Manager helps you to securely encrypt, store and retrieve credentials for your databases and other services.

6. Click **Next**.

   The **Configure stack options** window is displayed.

7. Verify the other fields and change the values appropriately. (Optional)
8. Click **Next**.

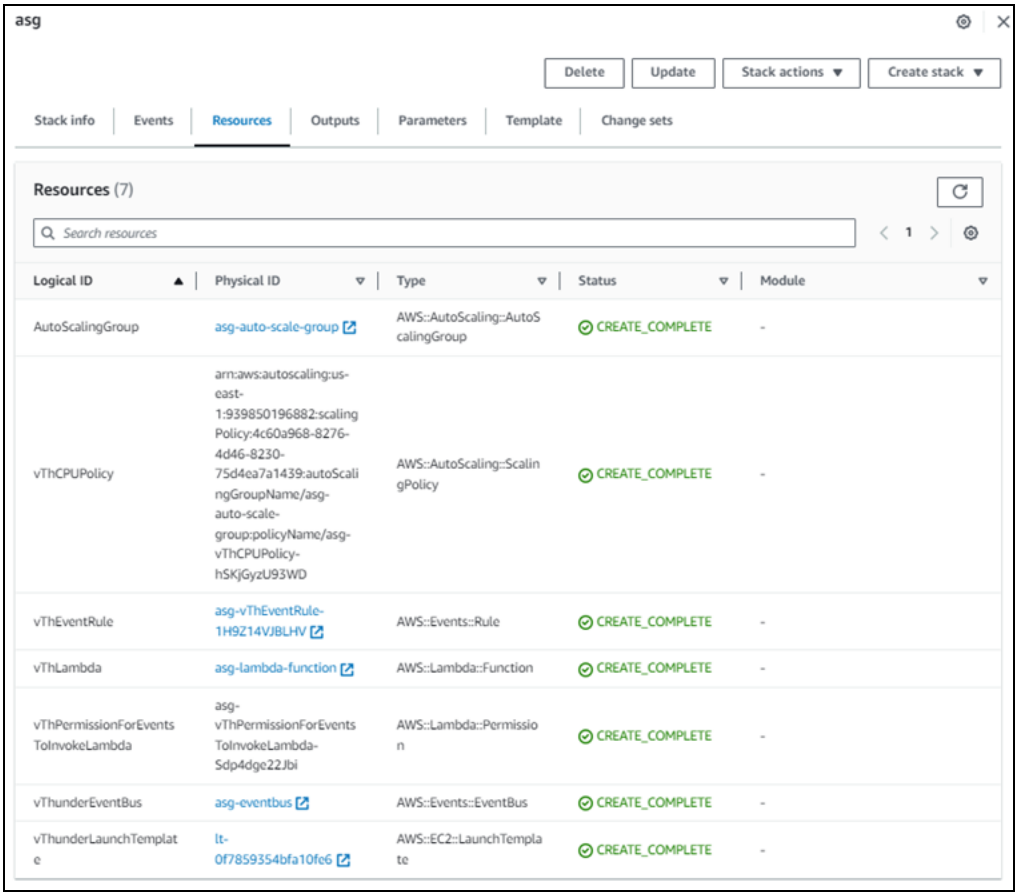
   The **Review** <*stack_name*> window is displayed.

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 and display the stack status as **CREATE_COMPLETE**.

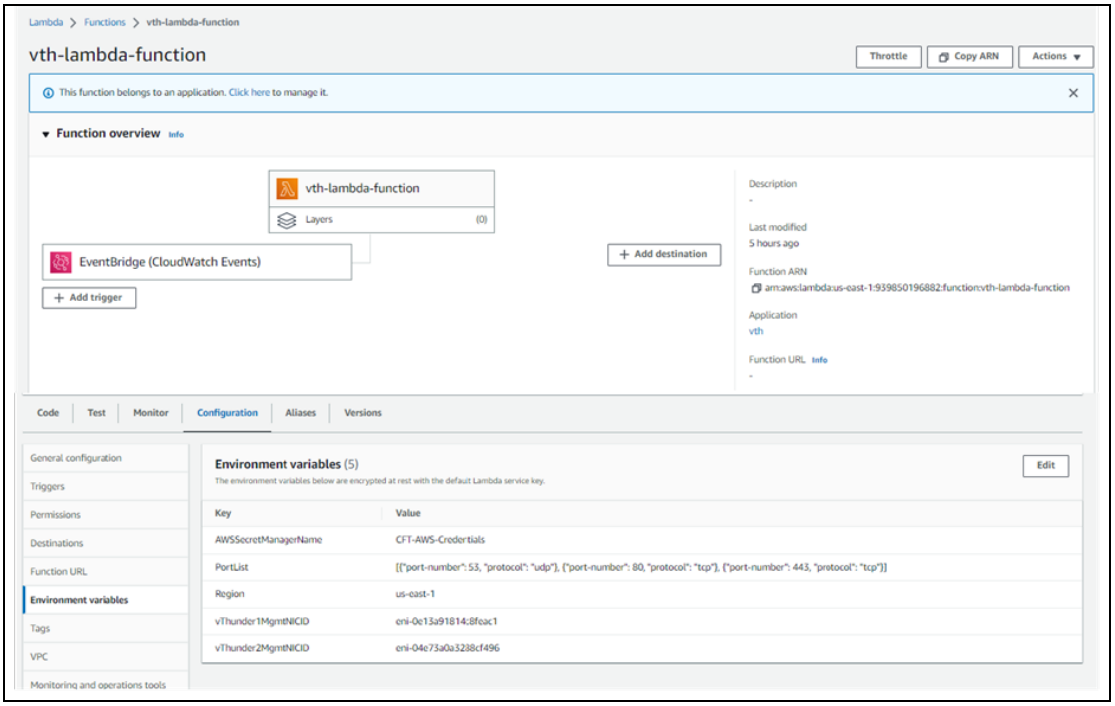
10. Verify if all the listed resources are created in the **AWS Management Console** > **CloudFormation** > **Stacks** > <*stack_name*> > **Resources** tab:

Figure 37 : Resource listing



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 38 : Lambda Function



NOTE: If you delete a stack that contains a Lambda function, the Lambda function will not be deleted automatically along with the stack. You need to manually delete the Lambda function separately. For more information, see Delete the resources.

Configure Lambda Function and ASG

To configure Lambda Function and ASG, perform the following steps:

1. Download **A10-vThunder_ADC-CONFIGURATION** > **CONFIG-SLB_ON_BACKEND-AUTOSCALE** template from GitHub.
2. From the Start menu, open command prompt and navigate to this downloaded template.
3. Open the AUTOSCALE_SERVER_PARAM.json with a text editor.

NOTE: Each parameter has a default value mentioned in the parameter file which can be modified as required.

4. Configure the following parameters:

Table 13 : JSON Parameters

Resource Name	Description
Desired Capacity	Set the capacity of the autoscale server. `"desiredCapacity": 1,`
Lambda Function	Specify the name of the lambda function. `"lambdaFunction": "lambda-function",`
autoscale Group	Specify the autoscale Group name. `"autoscaleGroupName": "auto-scale-group",`

Port List	Specify the SLB Server ports' details.
	`"port-list": [`
	`{`
	`"port-number": 53,`
	`"protocol": "udp"`
	`},`
	`{`
	`"port-number": 80,`
	`"protocol": "tcp"`
	`},`
	`{`
	`"port-number": 443,`
	`"protocol": "tcp"`
	`}`
	`]`

5. Verify if all the configurations in the AUTOSCALE_SERVER_PARAM.json file are correct and save the changes.
6. Run the following command to configure the vThunder instance/s as an SLB on

Backend AutoServer:

```
PS C:\Users\TestUser\A10-vThunder_ADC-CONFIGURATION\CONFIG-SLB_ON_
BACKEND-AUTOSCALE> python ./AUTOSCALE_SERVER_ASG_LAMBDA_UPDATE_2.py
```

If SLB is configured successfully, the following message is displayed:

```
Added PortList in asg-lambda environment variables
Updated desired capacity of autoscale group to 1
```

7. Verify the scale-in and scale-out instances in the **AWS Management Console** > **EC2** > **Auto Scaling groups** > <*asg_name*> > **Activity** tab.

SSL Certificate

This template applies Certificate Authority SSL Certificate to the vThunder instance. This certificate establishes an encrypted link between the server and your browser, ensuring that all data transferred between them remains private and secure.

To configure SSL certificate for a vThunder instance, perform the following steps:

1. Download **A10-vThunder_ADC-CONFIGURATION** > **SSL-CERTIFICATE** from GitHub.
2. From the Start menu, open command prompt and navigate to this downloaded template.
3. Open the SSL_CONFIG_PARAM.json with a text editor.

NOTE: Each parameter has a default value mentioned in the parameter file which can be modified as required.

4. Configure the following parameters:

Table 14 : JSON Parameters

Resource Name	Description
Public IP List	Specify the Public IP address of one or more vThunder instance/s and instance ID list (in the same order as public IP) of vThunders to change password.

Table 14 : JSON Parameters

Resource Name	Description
	`"publicIpList": ["X.X.X.X","X.X.X.X"],`

SSL Configuration	Specify SSL details.
	`"sslConfig": { "requestTimeOut": 40, "path": "server.pem", "file": "server", "certificationType": "pem" }`
	NOTE: By default, SSL configuration is disabled i.e. no SSL configuration is applied.
	Example
	The sample values for the SSL certificate are as shown below:
	`"sslConfig": { "requestTimeOut": 40, "Path": "C:\\.............. \\server.pem", "File": "server", "CertificationType": "pem" }`

5. Verify if the configurations in the SSL_CONFIG_PARAM.json file are correct and then save the changes.
6. Run the following command to apply SSL configuration on the vThunder instance/s:

```
PS C:\Users\TestUser\A10-vThunder_ADC-CONFIGURATION\SSL-CERTIFICATE>
python ./SSL_CONFIG.py
```

7. If the SSL certificate is uploaded successfully, the following message is displayed:

```
Configuring vThunder with ip x.x.x.x
Enter vThunder Password:
Successfully configured SSL.
Configurations are saved on partition: shared
Successfully logged out from vThunder.
```

A10 License

This template applies GLM license to the vThunder instance for legal compliance, security, all feature access, and support.

To configure GLM license for vThunder instance, perform the following steps:

1. Download **A10-vThunder_ADC-CONFIGURATION** > **GLM-LICENSE** from GitHub.
2. From the Start menu, open command prompt and navigate to this downloaded template.
3. Open the GLM_CONFIG_PARAM.json with a text editor.

NOTE: Each parameter has a default value mentioned in the parameter file, which can be modified as required.

4. Configure the following parameters:

Table 15 : JSON Parameters

Resource Name	Description
Public IP addresses	Specify the Public IP address to apply GLM. `"publicIpList": ["X.X.X.X","X.X.X.X"],`

DNS	Specify a domain namespace.
	`"dns": {`
	`"value": "8.8.8.8"`
	`},`

Entitlement Token	Specify the entitlement token.

Table 15 : JSON Parameters

Resource Name	Description

	"entitlement_token": {
	"value": "XXXXXXXXXXXXX"
	}

5. Verify if the configurations in the GLM_CONFIG_PARAM.json file are correct and then save the changes.
6. Run the following command to apply GLM license on the vThunder instance/s:

```
PS C:\Users\TestUser\A10-vThunder_ADC-CONFIGURATION\GLM-LICENSE> python
./GLM_CONFIG.py
```

7. Provide password for the vThunder instance/s whose IP address is mentioned in the GLM_CONFIG_PARAM.json file.

```
Configuring vThunder with ip x.x.x.x
Enter vThunder Password:
Successfully configured primary DNS.
Successfully configured GLM Entitlement token in vthunder.
GLM license request sent successfully.
Configurations are saved on partition: shared
Successfully logged out from vThunder.
```

High Availability

This template applies high availability configuration to the Thunder instances. It automatically synchronizes Thunder configurations between the active and standby Thunder instances. In the event of a failover, it designates the other Thunder instance as active to ensure uninterrupted traffic routing. For this functionality, it is essential for both Thunder instances to have identical resources and configurations.

High availability can be configured only within same availability zone in the same region.

Thunder instances should have the same ACOS versions; otherwise, traffic flow will be disrupted.

To configure HA for Thunder instances, perform the following steps:

1. Configure a new or an existing FTP server. For more information, see Configure the FTP server.
2. Download **A10-vThunder_ADC-CONFIGURATION** > **HIGH-AVAILABILITY** template from GitHub.
3. From Start menu, open command prompt and navigate to this downloaded template.
4. Open the HA_CONFIG_PARAM.json with a text editor.

NOTE: Each parameter has a default value mentioned in the parameter file which can be modified as required.

5. Configure the following parameters:

Table 16 : JSON Parameters

Resource Name	Description
Public IP addresses	Specify the Public IP address of one or more vThunder instance/s depending on the deployed template. `"publicIpList": ["X.X.X.X","X.X.X.X"],`
Instance IDs	Specify the EC2 instance ID list in the same order as the Public IP addresses. `"instanceIdList":["XXXXXXXXXX","XXXXXXXXXX"],` To get the EC2 Instance ID, go to **AWS** > <*region_name*> > **EC2** > **Instances** > <*instance_id*>.
ftpServerName	Specify the name of the FTP server to upload keys. `"ftpServerName":"ftp-server",`
DNS	Specify a domain namespace.

Table 16 : JSON Parameters

Resource Name	Description

	<pre>"dns": {
	"value": "8.8.8.8"
	},</pre>

Network Gateway IP	Specify 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 default management subnet configuration.
	
	The default value of next network gateway IP address is 10.0.2.1 as this is the first IP address of the data-in subnet configuration.
	
	<pre>"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"
	}
	]
	}
	],</pre>

VRRP-A	Specify the value as 1 to enable VRRP-A.
	
	<pre>"vrrp-a": {
	"set-id":1
	},</pre>

Terminal Idle Timeout	Specify the interval in minutes for closing connection when there is no input detected. The value '0' means never

Table 16 : JSON Parameters

Resource Name	Description
	timeout. `"terminal": { "idle-timeout":0 },`
VRID details	Specify the VRID details. The default value of vrid is 0. The default priority for the first vThunder instance is 100, and for second the vThunder is 99 (100-1). `"vrid-list": [ { "vrid-val": 0, "blade-parameters": { "priority": 100 } } ]`

6. Verify if all the configurations in the HA_CONFIG_PARAM.json file are correct and save the changes.
7. Import AWS access key on both the vThunder instances. For more information, refer Import the AWS Access Keys.
8. Run the following command to configure HA:

```
PS C:\Users\TestUser\A10-vThunder_ADC-CONFIGURATION\HIGH-AVAILABILITY>
python ./HA_CONFIG.py
```

9. Provide password for the vThunder instances/s whose IP address is mentioned in the HA_CONFIG_PARAM.json file.
   If HA is configured successfully, the following message is displayed:

```
Configuring vThunder with ip x.x.x.x
Enter vThunder password: *********
Successfully uploaded AWS access keys to vThunder
Successfully configured Primary DNS.
Successfully configured IP Route.
Successfully configured Vrrp-A Common.
Successfully configured Idle Timeout.
Successfully configured Vrrp Rid.
Successfully configured Peer Group.
Configurations are saved on partition: shared
Successfully logged out from vThunder.
------------------------------------------------------------------
Configuring vThunder with ip x.x.x.x
Enter vThunder password: *********
Successfully uploaded AWS access keys to vThunder
Successfully configured Primary DNS.
Successfully configured IP Route.
Successfully configured Vrrp-A Common.
Successfully configured Idle Timeout.
Successfully configured Vrrp Rid.
Successfully configured Peer Group.
Configurations are saved on partition: shared
Successfully logged out from vThunder.
------------------------------------------------------------------
```

High Availability Across Availability Zones

This template applies high availability configuration to the Thunder instances in the same availability zone having interface in different subnet or across different availability zones. It automatically synchronizes Thunder configurations between the active and standby Thunder instances. In the event of a failover, it designates the other Thunder instance as active to ensure uninterrupted traffic routing. For this functionality, it is essential for both Thunder instances to have identical resources and configurations.

Thunder instances should have the same ACOS versions; otherwise, traffic flow will be disrupted.

To configure HA Across AZs for Thunder instances, perform the following steps:

1. Configure a new or an existing FTP server. For more information, see Configure the FTP server.
   Alternatively, from ACOS 6.0.5 onwards, you can attach an IAM role to the Thunder device, which can be used to fetch the AWS access key and secret key.
2. Download **A10-vThunder_ADC-CONFIGURATION** > **HIGH-AVAILABILITY-ACROSS-AZs** template from GitHub.
3. From Start menu, open command prompt and navigate to this downloaded template.
4. Open the HA_Across_AZs_CONFIG_PARAM.json with a text editor.

NOTE: Each parameter has a default value mentioned in the parameter file which can be modified as required.

5. Configure the following parameters:

Table 17 : JSON Parameters

Resource Name	Description
Instance IDs	Specify the EC2 instance ID list in the same order as the Public IP addresses. `"instanceIdList":["XXXXXXXXXX","XXXXXXXXXX"],` To get the EC2 Instance ID, go to **AWS** > <*region_name*> > **EC2** > **Instances** > <*instance_id*>.
ftpServerName	Specify the name of the FTP server to upload keys. `"ftpServerName":"ftp-server",`

DNS	Specify a domain namespace.
	`"dns": {`
	`"value": "8.8.8.8"`
	`},`

Network	Specify a Network Gateway IP of active vThunder instance.

Table 17 : JSON Parameters

Resource Name	Description
Gateway IP for Active vThunder	The gateway IP is the first IP address of the management subnet, such as 10.0.1.1 in the configuration. You can configure multiple network gateway IPs for different routing paths. `"rib-list_active": [ { "ip-dest-addr": "0.0.0.0", "ip-mask": "/0", "ip-nexthop-ipv4": [ { "ip-next-hop": "10.0.1.1" } ] }, { "ip-dest-addr": "10.0.8.0", "ip-mask": "/24", "ip-nexthop-ipv4": [ { "ip-next-hop": "10.0.3.1" } ] } ]`
Network Gateway IP for Standby vThunder	Specify a Network Gateway IP of standby vThunder instance. The gateway IP is the first IP address of the management subnet, such as 10.0.1.1 in the configuration. You can configure multiple network gateway IPs for different routing paths.

Table 17 : JSON Parameters

Resource Name	Description

	"rib-list_standby": [
	{
	"ip-dest-addr": "0.0.0.0",
	"ip-mask": "/0",
	"ip-nexthop-ipv4": [{
	"ip-next-hop": "10.0.4.1"
	}]
	},
	{
	"ip-dest-addr": "10.0.8.0",
	"ip-mask": "/24",
	"ip-nexthop-ipv4": [{
	"ip-next-hop": "10.0.6.1"
	}]
	}
	]

VRRP-A	Specify the value as 1 to enable VRRP-A. "vrrp-a": { "set-id":1 },
Terminal Idle Timeout	Specify the interval in minutes for closing connection when there is no input detected. The value '0' means never timeout. "terminal": { "idle-timeout":0 },
VRID details for Active vThunder	Specify the VRID details for active vThunder instance. The floating IPs present should be pair of data-out IP (FIP) and data-in IP (VIP). The default value of vrid is 0. The default priority for the active vThunder instance is 100.

Table 17 : JSON Parameters

Resource Name	Description

	"vrid_list_active": [
	{
	"vrid-val": 0,
	"floating-ip": {
	"ip-address-cfg": [
	{
	"ip-address": "10.0.3.103"
	},
	{
	"ip-address": "10.0.2.121"
	}
	,{
	"ip-address":"30.30.30.30"
	},
	{
	"ip-address": "20.20.20.20"
	}
	]
	},
	"blade-parameters": {
	"priority": 100
	}
	}
	]

VRID details for Standby vThunder	Specify the VRID details for standby vThunder instance. The floating IPs present should be pair of data-out IP (FIP) and data-in IP (VIP). The default value of vrid is 0. The default priority for the standby vThunder instance is 99.

Table 17 : JSON Parameters

Resource Name	Description

	"vrid_list_standby": [
	{
	"vrid-val": 0,
	"floating-ip": {
	"ip-address-cfg": [
	{
	"ip-address": "10.0.6.30"
	},
	{
	"ip-address": "10.0.5.205"
	}
	,{
	"ip-address":"30.30.30.30"
	},
	{
	"ip-address": "20.20.20.20"
	}
	]
	},
	"blade-parameters": {
	"priority": 99
	}
	}
	]

IP Nat for Active vThunder	Specify the IP Nat for active vThunder instance. It includes the pool name, starting and ending IP addresses for outbound data, and the netmask.

Table 17 : JSON Parameters

Resource Name	Description

	<pre>"pool_list_active":[
	{
	"pool-name":"vrrp_0",
	"start-address":"10.0.3.103",
	"end-address":"10.0.3.103",
	"netmask":"/32"
	},{
	"pool-name":"vrrp_0_1",
	"start-address":"30.30.30.30",
	"end-address":"30.30.30.30",
	"netmask":"/32"
	}]</pre>

IP Nat for Standby vThunder	Specify the IP Nat for standby vThunder instance. It includes the pool name, starting and ending IP addresses for outbound data, and the netmask.
	<pre>"pool_list_standby":[
	{
	"pool-name":"vrrp_0",
	"start-address":"10.0.6.30",
	"end-address":"10.0.6.30",
	"netmask":"/32"
	},{
	"pool-name":"vrrp_0_1",
	"start-address":"30.30.30.30",
	"end-address":"30.30.30.30",
	"netmask":"/32"
	}],</pre>

Cloud Services for Active vThunder	Specify the cloud service information for the active vThunder instance. It includes details for each VRID such as the VIP number, private IP, and attached Elastic IP. If an alien IP is attached, provide the route table ID, data in interface ID, data out interface ID, data in destination CIDR,

Table 17 : JSON Parameters

Resource Name	Description
	and data out destination CIDR. If no alien IP is attached, either remove these fields or leave them blank. `"cloud_services_cloud_provider_active": { "multi-az-failover": { "vrid": { "vrid-number": 0, "route-table-id": "rtb-0031384a1b3ce", "vip-interface-id": "eni-02a4815eb4f4d", "fip-interface-id": "eni-077c93646f16", "vip-dest": "20.20.20.0/24", "fip-dest": "30.30.30.0/24", "vip-list": [{ "vip-number": 0, "private-ip": "10.0.2.121", "elastic-ip": "3.228.164.96" }] } } }`
Cloud Services for Standby vThunder	Specify the cloud service information for the standby vThunder instance. It includes details for each VRID such as the VIP number, private IP, and attached Elastic IP. If an alien IP is attached, provide the route table ID, data in interface ID, data out interface ID, data in destination CIDR, and data out destination CIDR. If no alien IP is attached, either remove these fields or leave them blank.

Table 17 : JSON Parameters

Resource Name	Description

	"cloud_services_cloud_provider_standby": {
	"multi-az-failover": {
	"vrid": {
	"vrid-number": 0,
	"route-table-id": "rtb-0031384a15ece",
	"vip-interface-id": "eni-0817fcd67e90b3",
	"fip-interface-id": "eni-050ff5e5a189e",
	"vip-dest": "20.20.20.0/24",
	"fip-dest": "30.30.30.0/24",
	"vip-list": [
	{
	"vip-number": 0,
	"private-ip": "10.0.5.205",
	"elastic-ip": "3.228.164.96"
	}
	]
	}
	}
	}

Virtual Server for Active vThunder	Specify the name of the virtual server for active vThunder instance, along with the port number and protocol, for which the pool name and ha-conn-mirror is to be added. Make sure that the pool name corresponds to the correct outbound data that pairs with the respective virtual server IP address. **NOTE:** For an alien IP, the **`ha-conn-mirror`** must always be set to 1.

Table 17 : JSON Parameters

Resource Name	Description

	<pre>"slb_virtual_server_active":[
	{
	"name":"vip",
	"port-list": [{
	"port-number":80,
	"protocol":"tcp",
	"pool":"vrrp_0",
	"ha-conn-mirror":0
	}]
	},
	{
	"name":"vip1",
	"port-list": [{
	"port-number":80,
	"protocol":"tcp",
	"pool":"vrrp_0_1",
	"ha-conn-mirror":1
	}]
	}
	]</pre>

Virtual Server for Standby vThunder	Specify the name of the virtual server for standby vThunder instance, along with the port number and protocol, for which the pool name and ha-conn-mirror is to be added. Make sure that the pool name corresponds to the correct outbound data that pairs with the respective virtual server IP address. **NOTE:** For an alien IP, the **`ha-conn-mirror`** must always be set to 1.

Table 17 : JSON Parameters

Resource Name	Description

	"slb_virtual_server_standby":[
	{
	"name":"vip",
	"port-list": [{
	"port-number":80,
	"protocol":"tcp",
	"pool":"vrrp_0",
	"ha-conn-mirror":0
	}]
	},
	{
	"name":"vip1",
	"port-list": [{
	"port-number":80,
	"protocol":"tcp",
	"pool":"vrrp_0_1",
	"ha-conn-mirror":1
	}]
	}
	]

6. Verify if all the configurations in the HA_Across_AZs_CONFIG_PARAM.json file are correct and save the changes.
7. Run the following command to configure HA:

```
PS C:\Users\TestUser\A10-vThunder_ADC-CONFIGURATION\HIGH-AVAILABILITY-ACROSS-AZs> python ./ HA_Across_AZs_CONFIG.py
```

8. Provide password for the vThunder instances/s whose IP address is mentioned in the HA_Across_AZs_CONFIG_PARAM.json file.

   If HA Across AZs is configured successfully, the following message is displayed:

```
Authorization for vThunder with IP x.x.x.x
Enter vThunder password:
Authorization for vThunder with IP x.x.x.x
Enter vThunder password:
--------------------------------------------------------------------------
----------
Configuring vThunder with IP x.x.x.x
Please note that we support generation of credential through IAM Role
attached to instance from ACOS-6.0.5 onward.
Does the instance have IAM role attached or not (YES/NO)? Yes
Successfully configured Primary DNS.
Successfully configured IP Route.
Successfully configured Vrrp-A Common.
Successfully configured Idle Timeout.
Successfully configured Vrrp-A Vrid.
Successfully configured Peer Group.
Successfully configured IP Nat pool.
Successfully configured Cloud Services.
Successfully updated SLB Virtual Server vip configuration.
Successfully updated SLB Virtual Server vip1 configuration.
Configurations are saved on partition: shared
Successfully logged out from vThunder.
--------------------------------------------------------------------------
----------
Configuring vThunder with IP x.x.x.x
Please note that we support generation of credential through IAM Role
attached to instance from ACOS-6.0.5 onward.
Does the instance have IAM role attached or not (YES/NO)? Yes
Successfully configured Primary DNS.
Successfully configured IP Route.
Successfully configured Vrrp-A Common.
Successfully configured Idle Timeout.
Successfully configured Vrrp-A Vrid.
Successfully configured Peer Group.
Successfully configured IP Nat pool.
Successfully configured Cloud Services.
Successfully updated SLB Virtual Server vip configuration.
```

```
Successfully updated SLB Virtual Server vip1 configuration.
Configurations are saved on partition: shared
Successfully logged out from vThunder.
--------------------------------------------------------------------------
-------
```

Hybrid Cloud GSLB

A hybrid cloud configuration as a Global Server Load balancer (GSLB) between two regions residing in the same or different cloud or on-premise environments. It provides flexibility to implement disaster recovery site.

It requires atleast two Thunder instances in each region or location. One instance serves as the master controller, while the other functions as the site device. It is possible to configure multiple site devices, but it is recommended to have a minimum of three site devices to ensure seamless failover and effective disaster recovery.

Both regions should maintain an equivalent number of resources, whether hosted in the cloud or on-premise.

To create and install three thunder instances in any one region use Thunder-3NIC-3VM template. The same template can be used to install Thunder instances in another region.

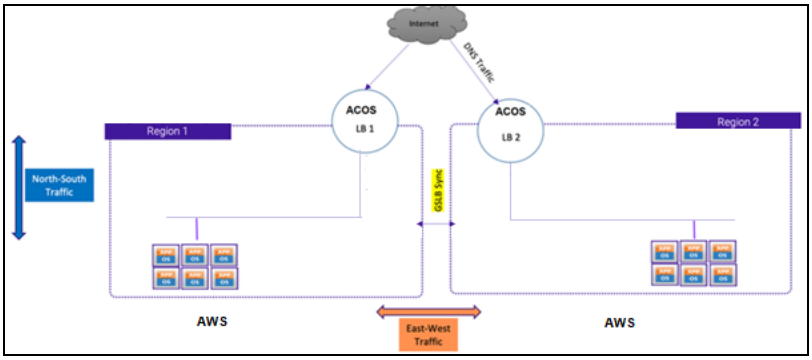
Architectural References

Refer to the following for architectural references:

- AWS-to-AWS

  Region 1 and Region 2 both are in AWS cloud.

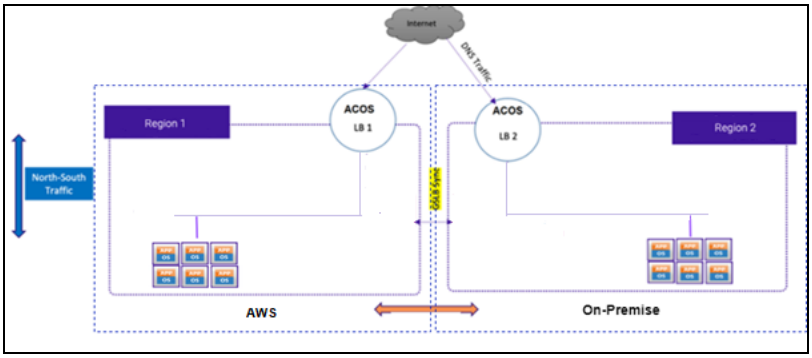
Figure 39 : AWS-to-AWS Cloud



- AWS-to-On-Premises (any)

  Region 1 is in AWS cloud and Region 2 is on-premise or vice versa.

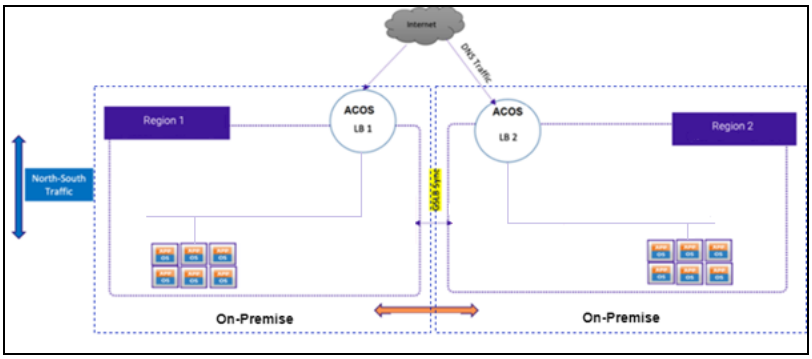
  Figure 40 : AWS-to-On-Premise



- On-Premise-to-On-Premise (any)

  Region 1 and Region 2 are on-premises.

  Figure 41 : On-Premise-to-On-Premise

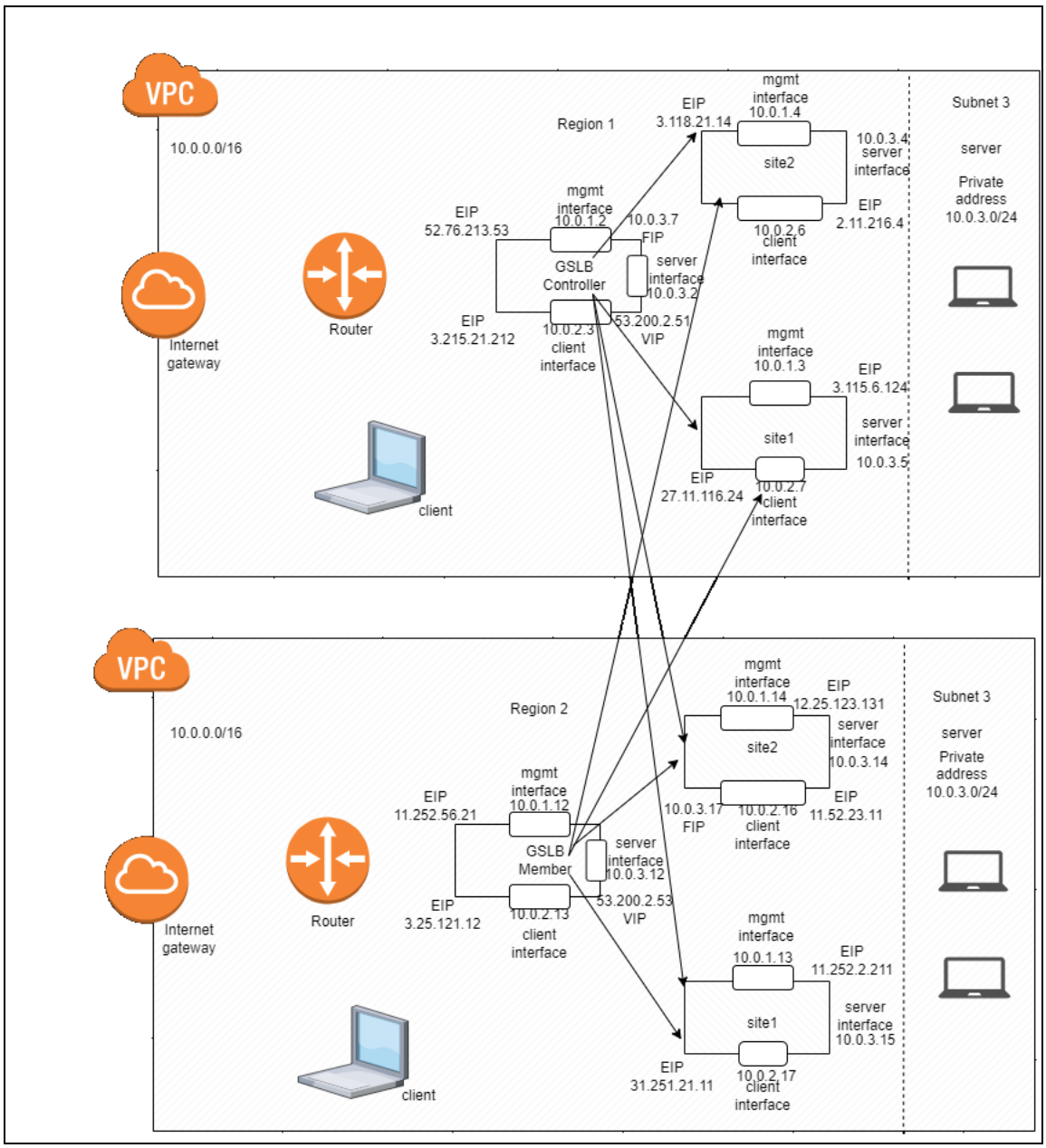


GSLB Deployment Topology

Figure 42 shows the GSLB deployment topology having two regions, Region 1 and Region 2. Both the regions must have identical number of resources:

- One GSLB controller
  This Thunder instance acts as a DNS server that directs clients to reach the active load balancer.
  GLSB controller of Region 1 is considered as the 'Master' and Region 2 is considered as a 'Member'.
- Two site devices
  These Thunder instances act as a load balancer and send traffic to the server. Each site device may have multiple app or web servers configured and route the traffic accordingly.

Figure 42 : Hybrid Cloud GSLB Deployment Topology



Configure Hybrid Cloud GSLB

A Hybrid Cloud GSLB configuration requires two regions and each region should host three Thunder instances.

To configure hybrid cloud GSLB, perform the following:

1. Create three vThunder instances if not already created. For more information, see Thunder-3NIC-3VM.
2. Download **A10-vThunder_ADC-CONFIGURATION** > **HYBRID-CLOUD-GSLB** folder from GitHub.
3. From the Start menu, open the command prompt and navigate to the downloaded template.
4. Open the HYBRID_CLOUD_CONFIG_GSLB_PARAM.json with a text editor.

NOTE: Each parameter has a default value mentioned in the parameter file which can be modified as required.

5. Configure the following parameters:

Table 18 : JSON Parameters

Resource Name	Description
Master Controller	Specify the Master Controller Parameter Details for Region 1. Master Controller is the first vThunder instance in Region 1 and it could be any vThunder instance.

Table 18 : JSON Parameters

Resource Name	Description
	"masterConfigDetails":{ "controllerMngmtPublicIp":"Public IP of Management Interface of Location1 Controller", "controllerPassword":"vThunder's Login password of Location1 Controller", "controllerSecPrivateIpData1":"Secondary Pvt IP of DataSubnet1 of Location1 Controller", "site1MngmtPublicIp":"Public IP of Management Interface of Location1 Site1", "site1Password":"vThunder's Login password of Location1 Site1 vThunder", "site2MngmtPublicIp":"Public IP of Management Interface of Location1 Site2", "site2Password":"vThunder's Login password of Location1 Site2 vThunder", "site1SecPrivateIpData1":"Secondary Pvt IP of DataSubnet1 of Location1 Site1 vThunder", "site1SecPublicIpData1":"Secondary Public IP of DataSubnet1 of Location1 Site1 vThunder", "site2SecPrivateIpData1":"Secondary Pvt IP of DataSubnet1 of Location1 Site2 vThunder", "site2SecPublicIpData1":"Secondary Public IP of DataSubnet1 of Location1 Site2 vThunder", "server1PrivateIp":"Private Ipv4 address of server1 of Location1", "server2PrivateIp":"Private Ipv4 address of server2 of Location1" },
Member Controller	Specify the Member Controller Parameter Details for Region 2. Member Controller is the first vThunder instance in Region 2 and it could be any vThunder instance.

Table 18 : JSON Parameters

Resource Name	Description
	"memberConfigDetails":{ "controllerMngmtPublicIp":"Public IP of Management Interface of Location2 Controller", "controllerPassword":"vThunder's Login password of Location2 Controller", "controllerSecPrivateIpData1":"Secondary Pvt IP of DataSubnet1 of Location2 Controller", "site1MngmtPublicIp":"Public IP of Management Interface of Location2 Site1", "site1Password":"vThunder's Login password of Location2 Site1 vThunder", "site2MngmtPublicIp":"Public IP of Management Interface of Location2 Site2", "site2Password":"vThunder's Login password of Location2 Site2 vThunder", "site1SecPrivateIpData1":"Secondary Pvt IP of DataSubnet1 of Location2 Site1 vThunder", "site1SecPublicIpData1":"Secondary Public IP of DataSubnet1 of Location2 Site1 vThunder", "site2SecPrivateIpData1":"Secondary Pvt IP of DataSubnet1 of Location2 Site2 vThunder", "site2SecPublicIpData1":"Secondary Public IP of DataSubnet1 of Location2 Site2 vThunder", "server1PrivateIp":"Private Ipv4 address of server1 of Location2", "server2PrivateIp":"Private Ipv4 address of server2 of Location2" },
SLB Server Ports	Specify the SLB server ports for site devices.

Table 18 : JSON Parameters

Resource Name	Description

	"slbServerPortList1": {
	"value": [
	{
	"port-number": 80,
	"protocol": "tcp",
	"health-check-disable":1
	}
	]
	},
	"slbServerPortList2": {
	"value": [
	{
	"port-number": 80,
	"protocol": "tcp",
	"health-check-disable":1
	}
	]
	},
	"slbServerPortList3": {
	"value": [
	{
	"port-number": 80,
	"protocol": "tcp",
	"health-check-disable":1
	}
	]
	},
	"slbServerPortList4": {
	"value": [
	{
	"port-number": 80,
	"protocol": "tcp",
	"health-check-disable":1
	}
	]
	},

Table 18 : JSON Parameters

Resource Name	Description
Service Group	Specify the SLB Service groups for site devices.

Table 18 : JSON Parameters

Resource Name	Description

```
"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
                }
            ]
        }
    ]
},
```

Table 18 : JSON Parameters

Resource Name	Description
Virtual Server	Specify the SLB virtual server for site devices. The virtual server default name is “vs1”.

Table 18 : JSON Parameters

Resource Name	Description

```
        "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"
```

Table 18 : JSON Parameters

Resource Name	Description
Service IP	Specify the 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,
```

Table 18 : JSON Parameters

Resource Name	Description

	<pre>"port-proto": "tcp"
	}
	]
	},</pre>

Sites	Specify the GSLB Site Details for controller A site is the vThunder instance in a region.

Table 18 : JSON Parameters

Resource Name	Description

	<pre>"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"
	},

Geo location	Specify the system geo location details for controller.

Table 18 : JSON Parameters

Resource Name	Description

	<pre>"geolocation": {
	"geo-location-iana": "0",
	"geo-location-geolite2-city": "1",
	"geolite2-city-include-ipv6": "0",
	"geo-location-geolite2-country": "0"
	},</pre>

dnsPolicy	Specify the GSLB DNS policy for controller.
	The default value of vrid is 0. The default priority for the first vThunder instance is 100, and for the second vThunder instance is 99 (100-1).
	<pre>"dnsPolicy": {
	"policy-name": "a10",
	"type": "health-check, geographic"
	},</pre>

GSLB Server	Specify the GSLB virtual server for controller. The default value of vrid is 0. The default priority for the first vThunder instance is 100, and for the second vThunder is 99 (100-1).

Table 18 : JSON Parameters

Resource Name	Description

	<pre>"gslbserverList1": {
	"virtual-server-name": "gslb-server",
	"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",
	"metadata": {
	"description": "gslb virtual server is using VIP from ethernet 1 secondary subnet"
	},
	"value": [
	{
	"port-number":53,
	"protocol":"udp",
	"gslb-enable": 1
	}
	]
	},</pre>

GSLB Protocol Status	Specify the GSLB protocol status for controller.
	<pre>"gslbprotocolStatus": {
	"status-interval": 1
	},</pre>

Table 18 : JSON Parameters

Resource Name	Description
GSLB Controller Protocol Status	Specify the GSLB group for controller. `"gslbcontrollerGroup1": { "name": "default", "priority": 255 }, "gslbcontrollerGroup2": { "name": "default", "priority": 100 },`
GSLB Zone	Specify the GSLB zone for controller. `"gslbzone": { "service-port": 80, "service-name": "www", "name" : "gslb.a10.com" },`
GSLB Zone	Specify the 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 default management subnet configuration. The default value of the next network gateway IP address is 10.0.2.1 as this is the first IP address of the data-in subnet configuration.

Table 18 : JSON Parameters

Resource Name	Description

	<pre>"rib-list_region1": [
	{
	"ip-dest-addr": "0.0.0.0",
	"ip-mask": "/0",
	"ip-nexthop-ipv4": [
	{
	"ip-next-hop": "10.1.2.1"
	},
	{
	"ip-next-hop": "10.1.1.1"
	}
	]
	}
	],
	
	"rib-list_region2": [
	{
	"ip-dest-addr": "0.0.0.0",
	"ip-mask": "/0",
	"ip-nexthop-ipv4": [
	{
	"ip-next-hop": "10.1.2.1"
	},
	{
	"ip-next-hop": "10.1.1.1"
	}
	]
	}
	],</pre>

Default Route	Specify the default route for vThunder instances.

Table 18 : JSON Parameters

Resource Name	Description

	<pre>"defaultroute1":
	{
	"next-hop1": "10.1.2.1",
	"next-hop2": "10.1.1.1"
	}</pre>

6. Verify if all the configurations in the HYBRID_CLOUD_CONFIG_GSLB_PARAM.json file are correct and save the changes.
7. Run the following command to configure GSLB:

```
PS C:\Users\TestUser\A10-vThunder_ADC-CONFIGURATION\HYBRID-CLOUD-GSLB>
python ./HYBRID_CLOUD_CONFIG_GSLB_1.py
```

8. If the Hybrid cloud is configured successfully, the following message is displayed:

```
Gathering public and private ip address for site devices.
-----------------------------------------------------------
configured ethernet- 1 ip
configured ethernet- 2 ip
Configuring slb server for site: site1
Successfully Configured slb server for site: site1
Configuring service group for site: site1
Successfully Configured service group for site: site1
Successfully Configured virtual server for site: site1
Successfully Configured gslb site: site1
Successfully Configured default route: site1
Configurations are saved on partition: shared
-----------------------------------------------------------
configured ethernet- 1 ip
configured ethernet- 2 ip
Configuring slb server for site: site2
Successfully Configured slb server for site: site2
Configuring service group for site: site2
Successfully Configured service group for site: site2
Successfully Configured virtual server for site: site2
Successfully Configured gslb site: site2
Successfully Configured default route: site2
Configurations are saved on partition: shared
-----------------------------------------------------------
configured ethernet- 1 ip
configured ethernet- 2 ip
Configuring slb server for site: site3
Successfully Configured slb server for site: site3
Configuring service group for site: site3
Successfully Configured service group for site: site3
Successfully Configured virtual server for site: site3
Successfully Configured gslb site: site3
Successfully Configured default route: site3
Configurations are saved on partition: shared
-----------------------------------------------------------
configured ethernet- 1 ip
configured ethernet- 2 ip
Configuring slb server for site: site4
```

```
Successfully Configured slb server for site: site4
Configuring service group for site: site4
Successfully Configured service group for site: site4
Successfully Configured virtual server for site: site4
Successfully Configured gslb site: site4
Successfully Configured default route: site4
Configurations are saved on partition: shared
------------------------------------------------------------
Configuring controller devices
configured ethernet- 1 ip
configured ethernet- 2 ip
Successfully Configuring gslb server for controller: masterController
Successfully Configured ServiceIp for site: masterController
Successfully Configured ServiceIp for site: masterController
Successfully Configured ServiceIp for site: masterController
Successfully Configured ServiceIp for site: masterController
Successfully Configured site information for: masterController
Successfully Configured site information for: masterController
Successfully Configured site information for: masterController
Successfully Configured site information for: masterController
Successfully Configured gslb policy for: masterController
Successfully Configured gslb zone for: masterController
Successfully Configured gslb controller and status interval:
masterController
Successfully Configured gslb controller group: masterController
Successfully Configured geo location: masterController
Successfully Configured default route: masterController
Configurations are saved on partition: shared
configured ethernet- 1 ip
configured ethernet- 2 ip
Successfully Configured gslb server for controller: memberController
Successfully Configured gslb controller group: memberController
Successfully Configured default route: memberController
Configurations are saved on partition: shared
```

Master Controller Parameter Details

Table 19 : Master Controller Parameter details

Parameter	Description	Sample value
controllerMngmtPublicIp	Public IP of Management Interface of Region 1 Controller.	104.45.152.126
controllerPassword	vThunder instance Login password of Region 1 Controller.	***
controllerSecPrivateIpData1	Secondary Private IP of Data Interface Subnet1 of Region 1 Controller.	10.1.20.8
site1MngmtPublicIp	Public IP of Management Interface of Region 1 Site1.	20.163.190.244
site1Password	vThunder instance Login password of Region 1 Site1.	***
site2MngmtPublicIp	Public IP of Management Interface of Region 1 Site2.	20.85.217.94
site2Password	vThunder instance Login password of Region 1 Site2.	***
site1SecPrivateIpData1	Secondary Private IP of DataSubnet1 of Region 1 Site1 vThunder.	10.1.20.9
site1SecPublicIpData1	Secondary Public IP of DataSubnet1 of Region 1 Site1 vThunder.	20.163.190.244
site2SecPrivateIpData1	Secondary Private IP of DataSubnet1 of Region 1 Site2 vThunder.	10.1.20.10
site2SecPublicIpData1	Secondary Public IP of DataSubnet1 of Region 1 Site2 vThunder.	20.85.217.94
server1PrivateIp	Private IPv4 address of Server1 of Region 1.	10.2.20.9
server2PrivateIp	Private IPv4 address of Server2	10.2.20.10

Table 19 : Master Controller Parameter details

Parameter	Description	Sample value
	of Region 1.	

Member Controller Parameter Details

Table 20 : Member Controller Parameter details

Parameter	Description	Sample value
controllerMngmtPublicIp	Public IP of Management Interface of Region 2 Controller.	20.124.0.232
controllerPassword	vThunder instance Login password of Region 2 Controller.	***
controllerSecPrivateIpData1	Secondary Private IP of Data Interface Subnet1 of Region 2 Controller.	10.1.20.14
site1MngmtPublicIp	Public IP of Management Interface of Region 2 Site1.	20.163.190.244
site1Password	vThunder instance Login password of Region 2 Site1.	***
site2MngmtPublicIp	Public IP of Management Interface of Region 2 Site2.	20.85.217.94
site2Password	vThunder instance Login password of Region 2 Site2.	***
site1SecPrivateIpData1	Secondary Private IP of DataSubnet1 of Region 2 Site1 vThunder.	10.1.20.15
site1SecPublicIpData1	Secondary Public IP of DataSubnet1 of Region 2 Site1 vThunder.	20.65.88.231
site2SecPrivateIpData1	Secondary Private IP of DataSubnet1 of Region 2 Site2 vThunder.	10.1.20.16
site2SecPublicIpData1	Secondary Public IP of DataSubnet1 of Region 2 Site2	20.65.95.155

Table 20 : Member Controller Parameter details

Parameter	Description	Sample value
	vThunder.	
server1PrivateIp	Private IPv4 address of Server1 of Region 2.	`10.2.20.9`
server2PrivateIp	Private IPv4 address of Server2 of Region 2.	`10.2.20.10`

Site Details

Table 21 : Site details

Site Name	VIP Name	Device Name	GEO Location
eastus_1	vs1	slb1	North America, United States
eastus_2	vs2	slb2	North America, United States
eastus2_1	vs3	slb3	North America.United States.California.San Jose
eastus2_2	vs4	slb4	North America.United States.California.San Jose

IP Routes

Table 22 : IP routes

RIB List Of Region	Destination IP Address	Subnet Mask	Next Hop
Region 1	0.0.0.0	/0	10.1.20.1
Region 2	0.0.0.0	/0	10.1.20.1

Appendix

Access Thunder Virtual Machine

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

- Access vThunder using CLI
- Access vThunder using GUI

Access vThunder using CLI

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

1. From **AWS Management Console**, navigate to **EC2** > **Instances**.
2. Select the vThunder instance name/s depending on your deployment template.

Figure 43 : 2NIC-1VM vThunder instance

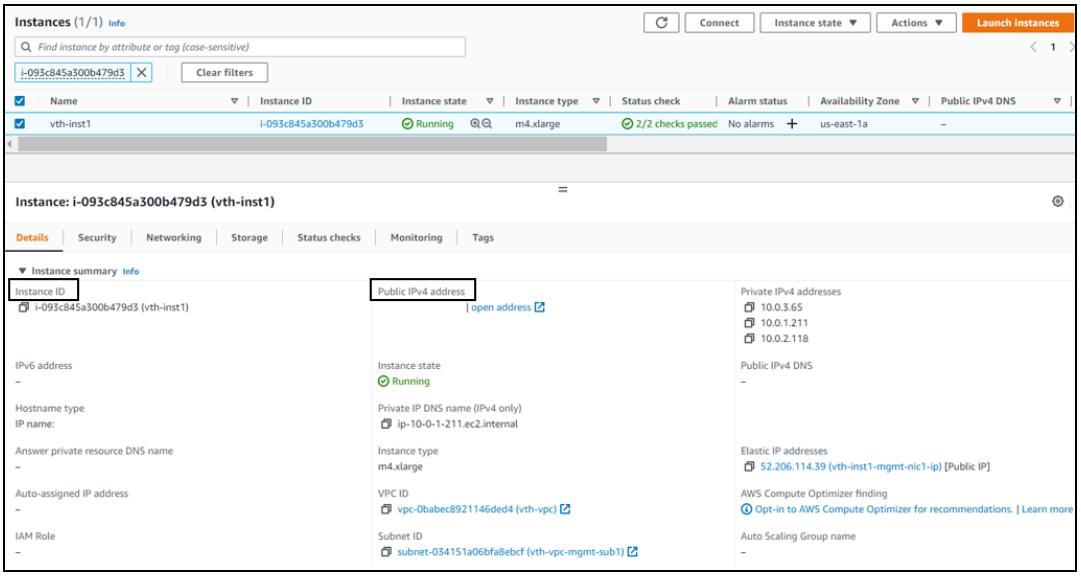


Table 23 : Sample vThunder instance name/s

CFT Template	vThunder Instance
Thunder-2NIC-1VM	`vth-inst1`

Table 23 : Sample vThunder instance name/s

CFT Template	vThunder Instance
Thunder-3NIC-2VM	**vth-inst1** **vth-inst2**
Thunder-3NIC-3VM	**vth-rg1-inst1** **vth-rg1-inst2** **vth-rg1-inst3** **vth-rg2-inst1** **vth-rg2-inst2** **vth-rg2-inst3**

3. For one or more 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 details 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 instance/s are ready to use.

Access vThunder using GUI

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

1. From **AWS Management Console**, navigate to **EC2** > **Instances**.
2. Select your instance name.

Figure 44 : 2NIC-1VM vThunder instance

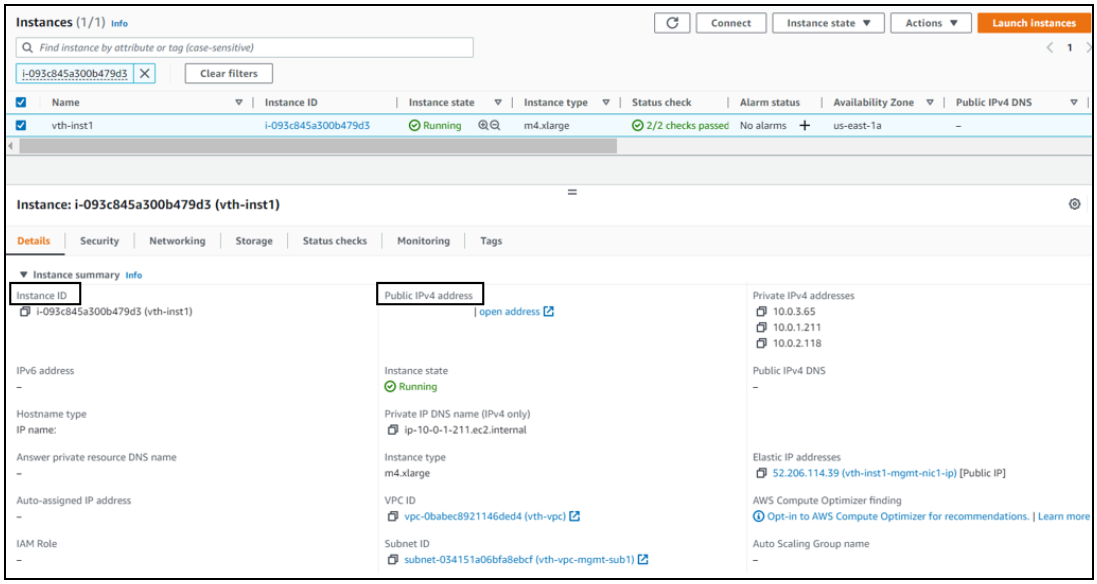


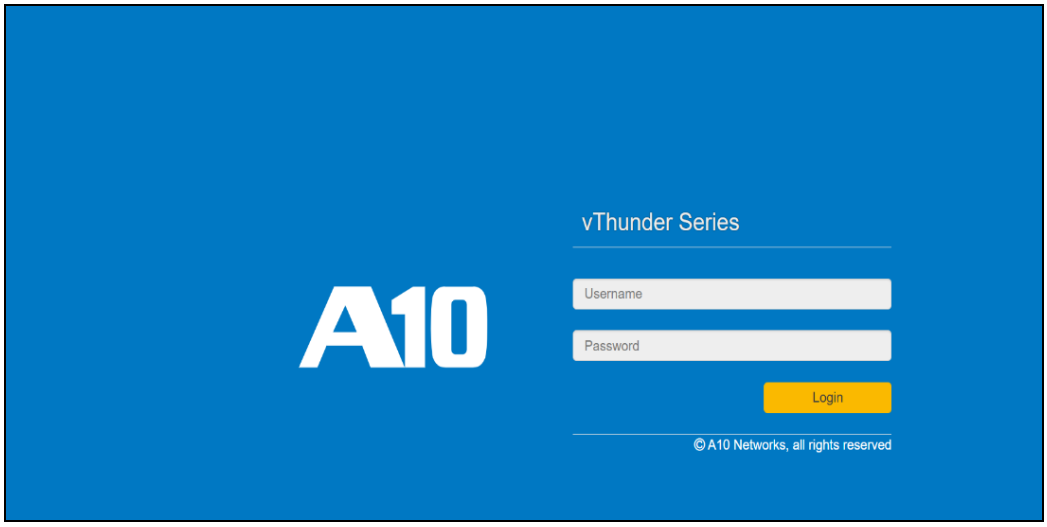
Table 24 : Sample vThunder instance name/s

CFT Template	vThunder Instance
Thunder-2NIC-1VM	`vth-inst1`
Thunder-3NIC-2VM	`vth-inst1` `vth-inst2`

Thunder-3NIC-3VM	`vth-rg1-inst1`
	`vth-rg1-inst2`
	`vth-rg1-inst3`
	`vth-rg2-inst1`
	`vth-rg2-inst2`
	`vth-rg2-inst3`

3. For one or more 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 45 : vThunder GUI



   c. Enter the following credentials and click **Login**:
      - Username – **admin**
      - Password – *EC2 Instance ID*

      The home page is displayed if the entered credentials are correct.

Create Server Machine

A server machine must be created and configured to test the traffic flow via vThunder.

To create and configure a server 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 `server1` as the server 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, enter `vpc` as the VPC.
   - Subnet: Enter value depending on your template.

     Table 25 : Subnet names

CFT Template	Subnet name
Thunder-2NIC-1VM	Data subnet `10.0.2.0/24`
Thunder-3NIC-2VM	Data subnet 2 `10.0.3.0/24`
Thunder-3NIC-3VM	Data subnet 2 `10.0.3.0/24`

   - Auto-assign public IP: Enable
   - Firewall (security groups): Select existing security group
   - Common security groups: *your data security group*
     Here, `vpc-vThunderSecurityGroupData` is the security group.
8. Click **Launch instance**.

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

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

9. Click **Connect**.
   A **Connect to instance** window with **EC2 Instance Connect** tab is displayed.
10. Click **Connect**.
    A **Terminal** window is displayed.
11. Run the following commands in the Terminal window to update all the package information:

    ```
    sudo apt update
    ```

12. 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.

13. If you want to configure HTTP template, perform the following steps:

    a. SSH the Apache Server and run the following command:

    ```
    sudo vim /etc/apache2/apache2.conf
    ```

    The Apache2 configuration file is displayed.

    b. Add the following configuration and save the file:

    ```
    Alias /<url-match-string> /var/www/html
    ```

    c. Restart the Apache server to enable the HTTP service.

    ```
    sudo systemctl restart apache2
    ```

    The server may take a few minutes to restart.

Create Client Machine

A client machine must be created and configured to test the traffic flow via vThunder.

To create and configure a client machine, perform the following steps:

1. From the **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 `client1` 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, enter `vpc` as the VPC.
   - Subnet: Enter value depending on your template.

     Table 26 : Subnet names

CFT Template	Subnet name
Thunder-2NIC-1VM	Data subnet `10.0.2.0/24`
Thunder-3NIC-2VM	Data subnet 2 `10.0.3.0/24`
Thunder-3NIC-3VM	Data subnet 2 `10.0.3.0/24`

   - Auto-assign public IP: Enable
   - Firewall (security groups): Select existing security group
   - Common security groups: *your data security group*
     Here, `vpc-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**.

Create Virtual Private Cloud

The VPC-SUBNET-NSG template is used to create virtual private cloud (VPC) containing three new subnets and two new security groups (SGs) in the same availability zones.

It is not mandatory to create new resources, the existing resources can be used in deployment and configuration.

If a Security group already exists, it should have the inbound rules as mentioned in Table 27 and Table 28; otherwise, traffic flow will be disrupted.

Table 27 : Management security rules

Security Rule	Protocol	Port
Custom TCP	TCP	4149
SSH	TCP	22
HTTPS	TCP	443
ALL ICMP -IPv4	ICMP	ALL
HTTP	TCP	80
Custom UDP	UDP	161
Custom TCP	TCP	123

Table 28 : Data security rules

Security Rule	Protocol	Port
Custom TCP	TCP	4149
SSH	TCP	22
HTTPS	TCP	443
ALL ICMP -IPv4	ICMP	ALL
HTTP	TCP	80
ALL UDP	UDP	0-65535

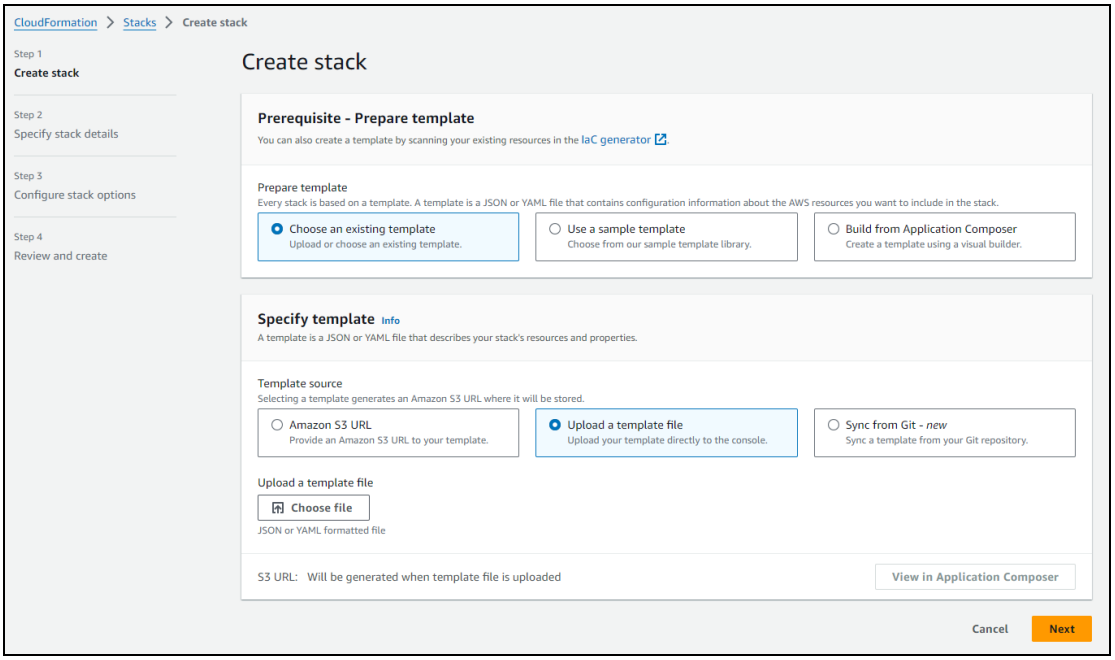
Before deploying this template, it is recommended to review the Prerequisites.

To create Virtual Private Cloud in same availability zone, perform the following steps:

1. Download **A10-vThunder_CFT-TEMPLATES** > **VPC-SUBNET-NSG** template from GitHub.
2. From AWS Management Console, navigate to **CloudFormation** > **Stacks** > **Create Stack** > **With new resources (standard)**.

   The Create stack window is displayed.

   Figure 46 : Create stack window



3. In the **Prerequisite - Prepare template** section, select **Choose an existing template**.

   After selecting this option, the Specify template section is displayed.

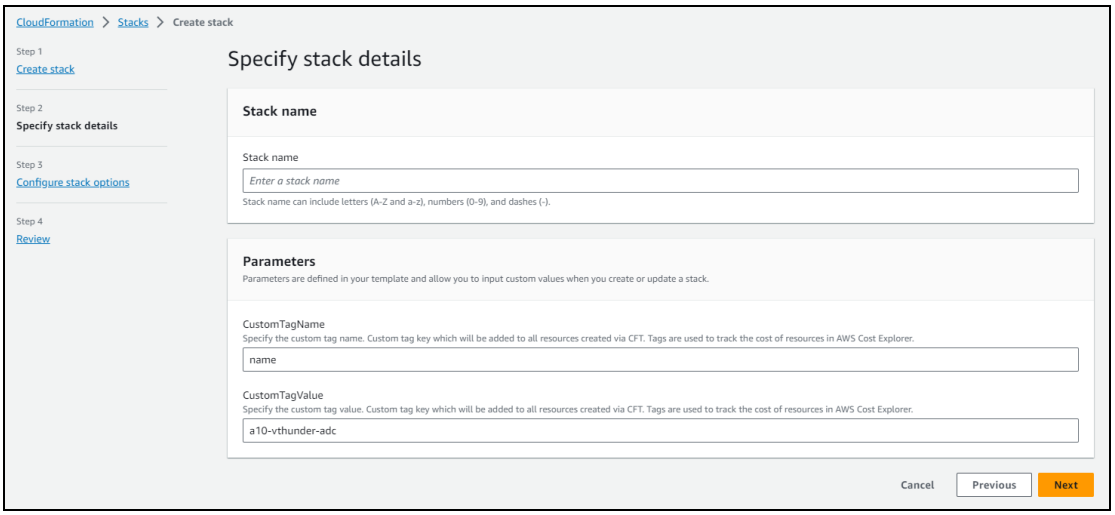
4. In the **Specify template** section, select **Upload a template file** and click **Choose file** to browse and upload the following template file from the downloaded CFT folder:
   CFT_TMPL_VIRTUAL_PRIVATE_COMPONENTS.json

   The selected template file name is displayed as the chosen file.

5. Click **Next**.

   The **Specify stack details** window is displayed.

   Figure 47 : Specify stack details window



6. In the **Specify stack details** window, enter or select the following:

   Table 29 : JSON Parameters

Resource Name	Description
Stack name	Specify a stack name containing letters (A-Z and a-z), numbers (0-9), and dashes (-). Here, the stack name is provided as `vpc`.
CidrDataSubnetIn	Specify the Classless Inter-Domain Routing (CIDR) of the data-in subnet.
CidrDataSubnetOut	Specify the CIDR of the data-out Subnet.
CidrMgmtSubnet	Specify the CIDR of the Management Subnet.
CidrVPC	Specify the IPv4 CIDR of the VPC. The allowed block size is between a /16 netmask (65,536 IP addresses) and /28 netmask (16 IP addresses).
CustomTagName	Specify the custom tag name. The custom tag key is added to all the resources created using the CFT template. This tag is used to track the cost of resources

Table 29 : JSON Parameters

Resource Name	Description
	in the AWS Cost Explorer.
CustomTagValue	Specify the custom tag value. The custom tag key is added to all the resources created using the CFT template. This tag is used to track the cost of resources in the AWS Cost Explorer. The default custom tag value is **a10-vthunder-adc**.
Zone	Select the existing availability zones. AWS offers a range of options for managing availability and resiliency for your applications. **NOTE:** Architect your solution to use the replicated VMs across the availability zones to protect your applications and data against the datacenter outages and maintenance events.

7. Click **Next**.

   The **Configure stack options** window is displayed.

8. Verify the other fields and change the values appropriately. (Optional)
9. Click **Next**.

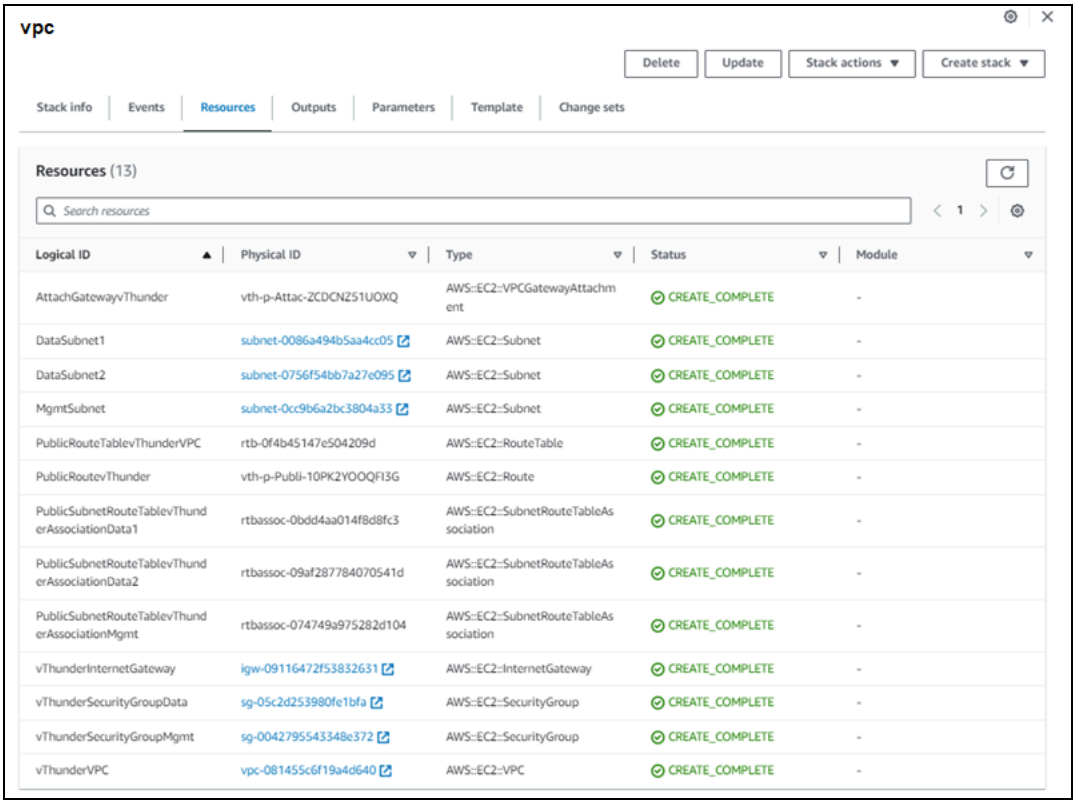
   The **Review** <*stack_name*> window is displayed.

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 and display the stack status as **CREATE_COMPLETE**.

11. Verify if the elastic IPs are created in the **AWS Management Console** > **CloudFormation** > **Stacks** > <*stack_name*> > **Resources** tab.

Figure 48 : Resource listing



Create Virtual Private Cloud for Different Availability Zones

The VPC-SUBNET-NSG template is used to create virtual private cloud (VPC) containing six new subnets and two new security groups (SGs) across different availability zones.

It is not mandatory to create new resources, the existing resources can be used in deployment and configuration.

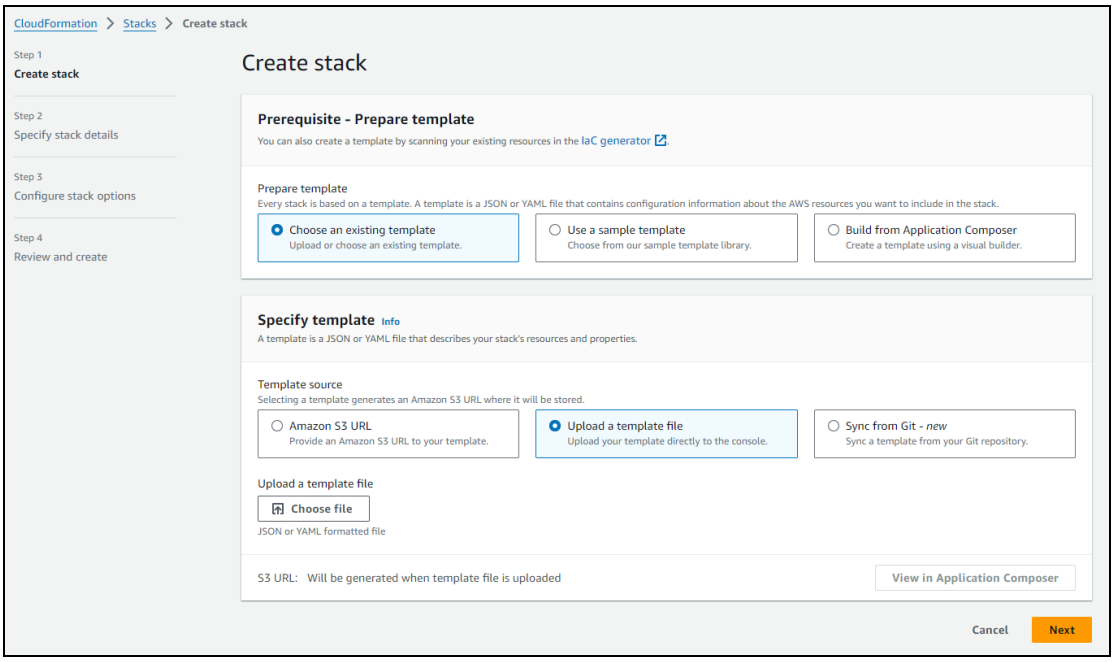
Before deploying this template, it is recommended to review the Prerequisites.

To create Virtual Private Cloud for different availability zones, perform the following steps:

1. Download **A10-vThunder_CFT-TEMPLATES** > **VPC-SUBNET-NSG** template from GitHub.
2. From AWS Management Console, navigate to **CloudFormation** > **Stacks** > **Create Stack** > **With new resources (standard)**.

   The Create stack window is displayed.

   Figure 49 : Create stack window



3. In the **Prerequisite - Prepare template** section, select **Choose an existing template**.

   After selecting this option, the Specify template section is displayed.

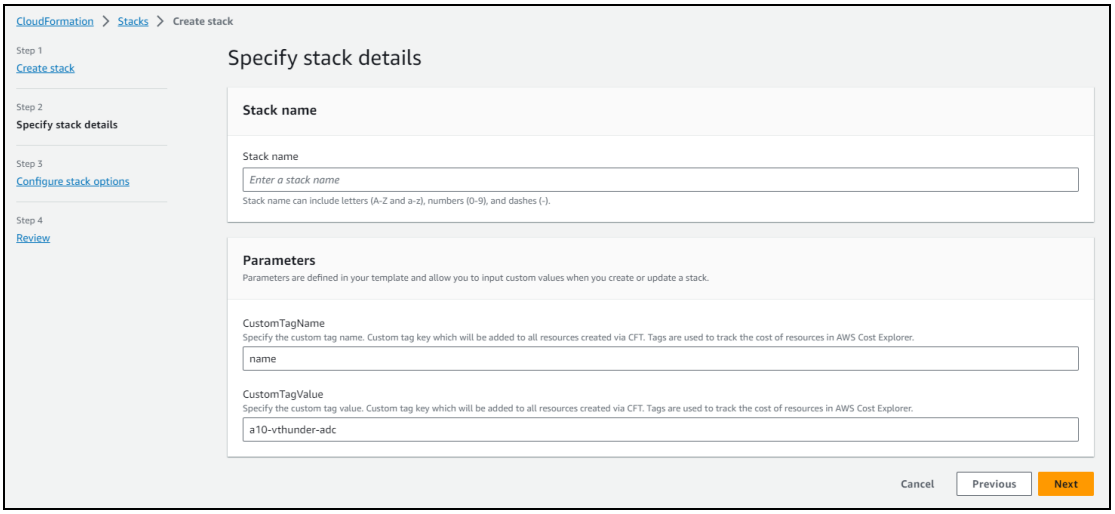
4. In the **Specify template** section, select **Upload a template file** and click **Choose file** to browse and upload the following template file from the downloaded CFT folder:
   CFT_TMPL_VIRTUAL_PRIVATE_COMPONENTS_Across_AZ_HA.json

   The selected template file name is displayed as the chosen file.

5. Click **Next**.

   The **Specify stack details** window is displayed.

Figure 50 : Specify stack details window



6. In the **Specify stack details** window, enter or select the following:

Table 30 : JSON Parameters

Resource Name	Description
Stack name	Specify a stack name containing letters (A-Z and a-z), numbers (0-9), and dashes (-). Here, the stack name is provided as `vpc`.
CidrDataInSubnetActiveThunder	Specify the Classless Inter-Domain Routing (CIDR) of the Active Data-In Subnet.
CidrDataInSubnetStandByThunder	Specify the CIDR of the StandBy Data-In Subnet.
CidrDataOutSubnetActiveThunder	Specify the CIDR of the Active Data-Out Subnet.
CidrDataOutSubnetStandByThunder	Specify the CIDR of the StandBy Data-Out Subnet.
CidrMgmtSubnetActiveThunder	Specify the CIDR of the Active Management Subnet.
CidrMgmtSubnetStandByThunder	Specify the CIDR of the StandBy

Table 30 : JSON Parameters

Resource Name	Description
	Management Subnet.
CidrVPC	Specify the IPv4 CIDR of the VPC. The allowed block size is between a /16 netmask (65,536 IP addresses) and /28 netmask (16 IP addresses).
CustomTagName	Specify the custom tag name. The custom tag key is added to all the resources created using the CFT template. This tag is used to track the cost of resources in the AWS Cost Explorer.
CustomTagValue	Specify the custom tag value. The custom tag key is added to all the resources created using the CFT template. This tag is used to track the cost of resources in the AWS Cost Explorer. The default custom tag value is **a10-vthunder-adc**.
DifferentAvailabilityZone	Select **Yes** if you want to deploy your vThunder instances across different availability zones. The default set value is **No**.
ZoneActiveThunder	Select from the existing availability zones for active vThunder instance. **NOTE:** Architect your solution to use the replicated VMs across the availability zones to protect your applications and data against the datacenter outages and maintenance events.

Table 30 : JSON Parameters

Resource Name	Description
ZoneStandByThunder	Select from the existing availability zones for standby vThunder instance. **NOTE:** Architect your solution to use the replicated VMs across the availability zones to protect your applications and data against the datacenter outages and maintenance events.

7. Click **Next**.

   The **Configure stack options** window is displayed.

8. Verify the other fields and change the values appropriately. (Optional)
9. Click **Next**.

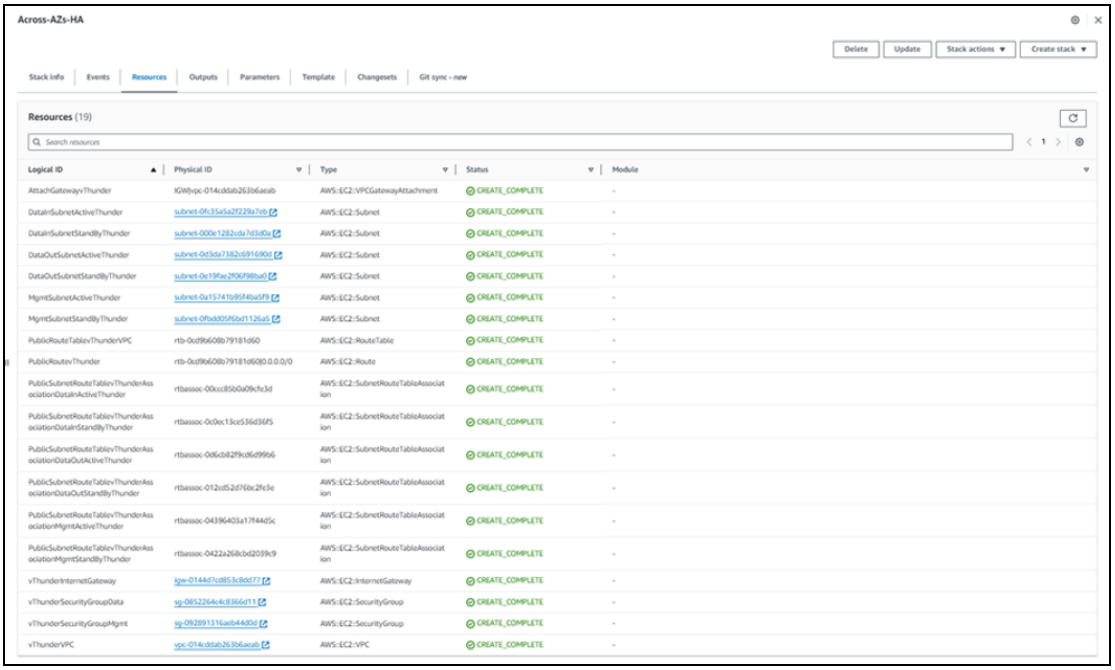
   The **Review** <*stack_name*> window is displayed.

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 and display the stack status as **CREATE_COMPLETE**.

11. Verify if the elastic IPs are created in the **AWS Management Console** > **CloudFormation** > **Stacks** > <*stack_name*> > **Resources** tab.

Figure 51 : Resource listing



Create Elastic Public IP

The PUBLIC-IP template is used to create three new elastic public IP addresses.

It is not mandatory to create new resources, the existing resources can be used in deployment and configuration.

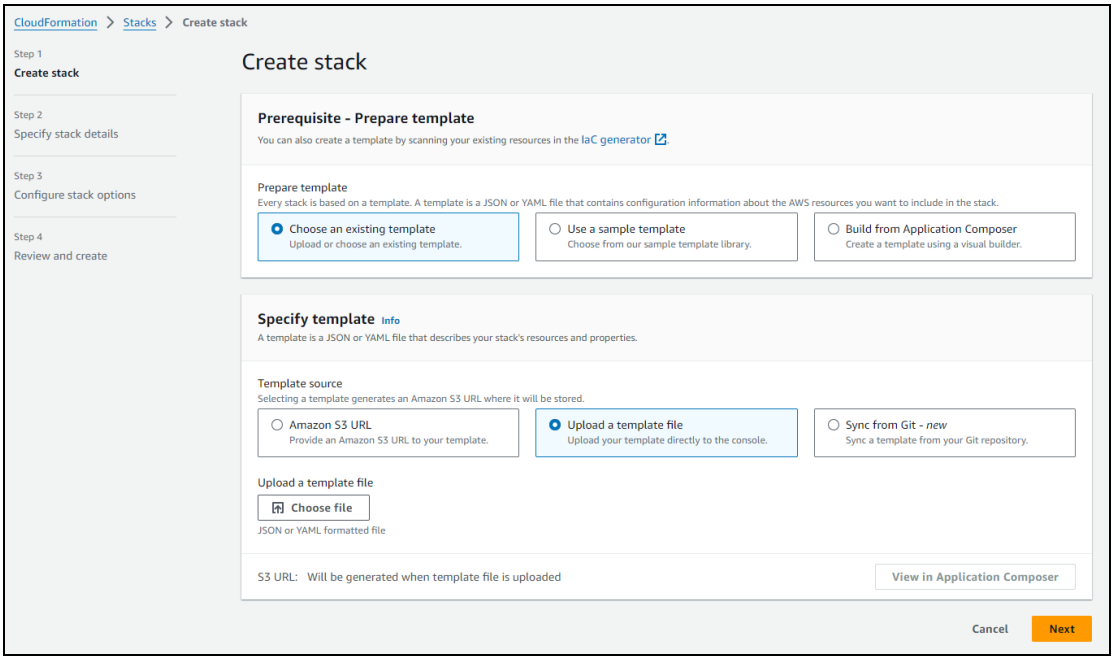
Before deploying this template, it is recommended to review the Prerequisites.

To create Elastic Public IP, perform the following steps:

1. Download **A10-vThunder_CFT-TEMPLATES** > **PUBLIC-IP** template from GitHub.
2. From the **AWS Management Console**, navigate to **CloudFormation** > **Stacks** > **Create Stack** > **With new resources (standard)**.

   The Create stack window is displayed.

Figure 52 : Create stack window



3. In the **Prerequisite - Prepare template** section, select **Choose an existing template**.

   After selecting this option, the Specify template section is displayed.

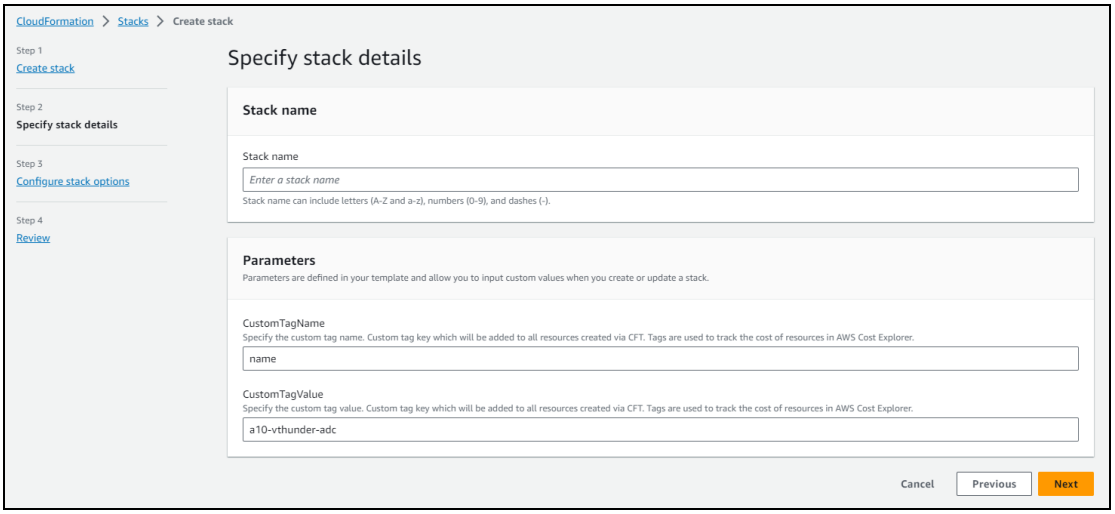
4. In the **Specify template** section, select **Upload a template file** and click **Choose file** to browse and upload the following template file from the downloaded CFT folder:
   CFT_TMPL_ELASTIC_PUBLIC_IP.json

   The selected template file name is displayed as the chosen file.

5. Click **Next**.

   The **Specify stack details** window is displayed.

Figure 53 : Specify stack details window



6. In the **Specify stack details** window, enter or select the following:

Table 31 : JSON Parameters

Resource Name	Description
Stack name	Specify a stack name containing letters (A-Z and a-z), numbers (0-9), and dashes (-). Here, the stack name is provided as `eip`.
CustomTagName	Specify the custom tag name. The custom tag key is added to all the resources created using the CFT template. This tag is used to track the cost of resources in the AWS Cost Explorer.
CustomTagValue	Specify the custom tag value. The custom tag key is added to all the resources created using the CFT template. This tag is used to track the cost of resources in the AWS Cost Explorer. The default custom tag value is **a10-vthunder-adc**.

7. Click **Next**.

   The **Configure stack options** window is displayed.

8. Verify the other fields and change the values appropriately. (Optional)

9. Click **Next**.

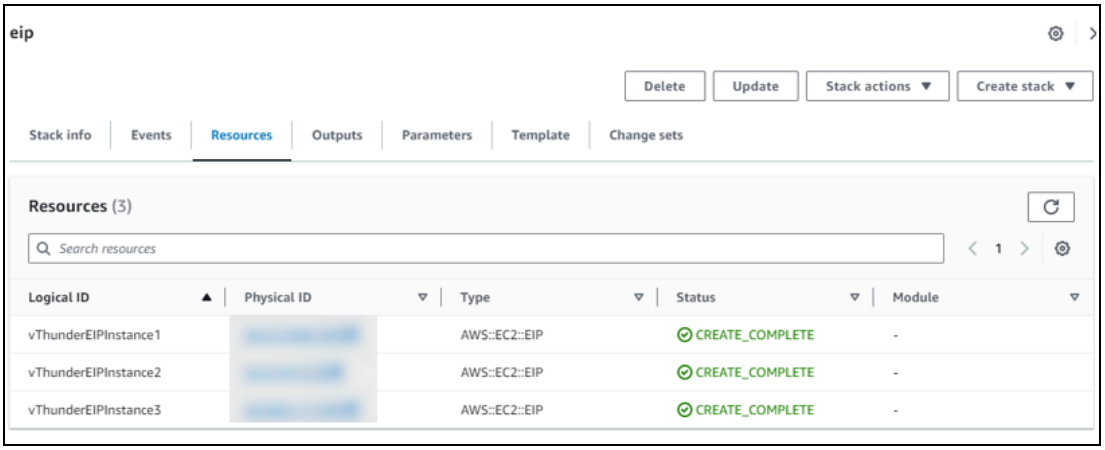
   The **Review** <*stack_name*> window is displayed.

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 and display the stack status as **CREATE_COMPLETE**.

11. Verify if the elastic IPs are created in the **AWS Management Console** > **CloudFormation** > **Stacks** > <*stack_name*> > **Resources** tab.

Figure 54 : Resource listing



Default Password Policy

The default password policy has the following criteria:

- The password should be at least nine characters in length.
- The password should contain at least one number, an uppercase letter (English), a lowercase letter (English), and a special character.
- The password should have at least one letter or number different from the previous password.
- The password should not contain its corresponding username with the same capitalization of letters.

- The password should not contain repeated characters of the same letter or number with the same capitalization of letters.
- The password should not contain the sequential row keyboard input of four letters or numbers with the same capitalization of letters.

Configure the FTP server

In a High Availability (HA) configuration, IP switching occurs between two vThunder instances. The IP switching is enabled when AWS keys are imported on the vThunder instances using the FTP server. You can configure an existing FTP server or create a new one.

Create FTP server

To create a new FTP server if an existing FTP server is not available, perform the following steps:

1. From **AWS Management Console**, navigate to **EC2** > **Instances** > <*FTP_server_name*>.
2. Click **Launch Instances**.
   A **Launch an instance** window is displayed.
3. In the **Name and tags** section, enter an instance name.
   Here, enter **`FTPserver`** as the FTP server 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, enter **`vpc`** as the VPC.
   - Subnet: Data subnet
     Here, **`10.0.3.0/24`** 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, **`vpc-vThunderSecurityGroupData`** is the security group.

8. Click **Launch instance**.

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

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

9. Click **Connect**.
   A **Connect to instance** window with **EC2 Instance Connect** tab is displayed.
10. Click **Connect**.
    A **Terminal** window is displayed.

Configure FTP server

To configure a new or existing FTP server, perform the following steps:

1. Run the following command in the Terminal window of the FTP server instance to update all the package information:

   ```
   sudo apt update
   ```

2. 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.

3. Run the following command to open the **`aws_access_key.txt`** file:

   ```
   sudo vi /var/www/html/aws_access_key.txt
   ```

4. Press **Esc** and enter **i** to enable edit/insert mode.
5. Locate and open the **credentials** file from downloaded CFT folder.

6. Copy the updated access key ID and secret access key as per your AWS account and paste in the **`aws_access_key.txt`** file. For more information, see Prerequisites.
7. After the changes, press **Esc** then type **:wq** to save the changes and exit.

   The AWS access keys are imported.

NOTE:	The FTP server should be deleted manually after the AWS access keys are imported to the vThunder instance.

Delete the resources

In case you want to delete the auto scaling group, you should also delete the Lambda Function and S3 bucket manually.

Lambda Function

To delete the Lambda Function, perform the following steps:

1. From **AWS Management Console**, navigate to **Lambda** > **Functions** and select the required lambda function.
2. Click **Action** > **Delete**.
3. Provide the confirmation in the input field and click **Delete**.

   The Lambda Function is deleted.

S3 Bucket

Before deleting your S3 Bucket, make sure the bucket is empty. To delete the S3 Bucket, perform the following steps:

1. From **AWS Management Console**, navigate to **Amazon S3** > **Buckets** and select the required bucket name from the list.
2. Click **Empty**.
3. Confirm the bucket name that you want to empty in the text field of the **Empty**

bucket window, and then click **Empty**.

4. Select the same bucket name from the list in the **Buckets** window and click **Delete**.
5. Confirm the bucket name that you want to delete in the text field of the **Delete bucket** window, and then click **Delete**.

   The S3 Bucket is deleted.

Install Python3

Depending on your operating system, install Python (3.8.5 or higher):

CentOS

To install latest Python3 from OS repository on CentOS, perform the following steps:

```
yum install -y python3
```

Linux/Ubuntu

To install Python3 on Linux or Ubuntu, perform the following steps:

```
apt update
apt-get install python3.10
apt install python3-pip
```

Windows

To install Python3 on Windows, see Using Python on Windows.

Verify Installation

To verify if the recommended Python version is installed correctly, perform the following steps:

1. Run the following command using the command prompt:

   CentOS/Linux/Ubuntu

```
$ python3 --version
```

Windows

```
$ python--version
```

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

2. Run the following command to verify if PIP is installed.

```
$ pip
```

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

Install Python dependencies

To install all Python dependencies, perform the following steps:

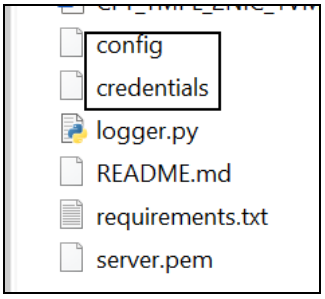
a. From the command prompt, navigate to the downloaded CFT folder path and enter the following command:

```
$ 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 55 : 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 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.

List of ACOS AMI ID

The following table lists the available ACOS AMI ID in the AWS cloud:

ACOS Image name	AMI IDs Region-wise	
	Region	**AMI ID**
A10 Thunder ADC for Advanced Load Balancing - 20 Mbps	US East (N. Virginia) - us-east-1	ami-00fbb189ff839e3f8
	US East (Ohio) - us-east-2	ami-0b059f3c25ba70d82
	US West (N. California) - us-west-1	ami-0dc2304f753c0c94f
	US West (Oregon) - us-west-2	ami-05fba450fda39ce03
	Asia Pacific (Mumbai) - ap-south-1	ami-08322b24e3046024a
	Asia Pacific	ami-

ACOS Image name	AMI IDs Region-wise	
	Region	AMI ID
	(Seoul) - ap-northeast-2	095e9bbbcaa03b1d3
	Asia Pacific (Singapore) - ap-southeast-1	ami-0b4af49b96c9cc8c8
	Asia Pacific (Sydney) - ap-southeast-2	ami-02dd0134a157ad9a2
	Asia Pacific (Tokyo) - ap-northeast-1	ami-0d06fbbb4f3b9da74
	Canada (Central) - ca-central-1	ami-004fb9c529a53d91c
	EU (Frankfurt) - eu-central-1	ami-04b07a4bac2b39f9c
	EU (Ireland) - eu-west-1	ami-07864041c98f74598
	EU (London) - eu-west-2	ami-00faba50b2d990a28
	EU (Paris) - eu-west-3	ami-0986e3946139dab92
	EU (Stockholm) - eu-north-1	ami-0f9e0a7d88ce19da3
	South America (Sao Paulo) - sa-east-1	ami-0144a14fa5ed1c046
A10 Thunder ADC for Advanced Load Balancing - 20 Mbps A10 Thunder ADC for Advanced Load Balancing – BYOL	US East (N. Virginia) - us-east-1	ami-08b72838b5b9121a9
	US East (Ohio) -	ami-

ACOS Image name	AMI IDs Region-wise	
	Region	AMI ID
	us-east-2	0b23e97b382324df5

ACOS Image name	AMI IDs Region-wise	
	Region	AMI ID
	US West (N. California) - us-west-1	ami-0c3e347bcd5642277
	US West (Oregon) - us-west-2	ami-05bd9ed4d72927b3e
	Asia Pacific (Mumbai) - ap-south-1	ami-06b7ecdfb12a4167d
	Asia Pacific (Seoul) - ap-northeast-2	ami-0014cc810cb66af81
	Asia Pacific (Singapore) - ap-southeast-1	ami-0a673d2f13bc18a36
	Asia Pacific (Sydney) - ap-southeast-2	ami-077b7ee13fd0489ee
	Asia Pacific (Tokyo) - ap-northeast-1	ami-0984eb1fa0d31d261
	Canada (Central) - ca-central-1	ami-0f20b5f0e324e0d4b
	EU (Frankfurt) - eu-central-1	ami-0dadf33ba3adad80e
	EU (Ireland) - eu-west-1	ami-0f43d27a21e631cb7
	EU (London) - eu-west-2	ami-076917d6e58ca7fc8
	EU (Paris) - eu-	ami-

ACOS Image name	AMI IDs Region-wise	
	Region	**AMI ID**
	west-3	00eaa7b84cbc3b33d
	EU (Stockholm) - eu-north-1	ami-0e8cea9ec067e213e
	South America (Sao Paulo) - sa-east-1	ami-00d46c65e7912919d
A10 Thunder ADC for Advanced Load Balancing - 1 Gbps	US East (N. Virginia) - us-east-1	ami-09de2cf3156153b78
	US East (Ohio) - us-east-2	ami-05a159f0bc348f529
	US West (N. California) - us-west-1	ami-09a4e9040933f8609
	US West (Oregon) - us-west-2	ami-0471c4c329eda84c8
	Asia Pacific (Mumbai) - ap-south-1	ami-06f79a3d0fc91e436
	Asia Pacific (Seoul) - ap-northeast-2	ami-0671bc6237cbb9cd9
	Asia Pacific (Singapore) - ap-southeast-1	ami-060466083903a7b78
	Asia Pacific (Sydney) - ap-southeast-2	ami-04144f14f23a706b5
	Asia Pacific (Tokyo) - ap-	ami-0f5a697923248645d

ACOS Image name	AMI IDs Region-wise	
	Region	AMI ID
	northeast-1	
	Canada (Central) - ca-central-1	ami-06a4deff50d12aff7
	EU (Frankfurt) - eu-central-1	ami-071bb4b3983815e9d
	EU (Ireland) - eu-west-1	ami-00aec66952e464ae9
	EU (London) - eu-west-2	ami-0fece56abc83f6442
	EU (Paris) - eu-west-3	ami-01bd1c56ebd7fdc15
	EU (Stockholm) - eu-north-1	ami-0042a12ce58f852ac
	South America (Sao Paulo) - sa-east-1	ami-0f35adf0672fdbbde
A10 Thunder ADC for Advanced Load Balancing - 100 Mbps	US East (N. Virginia) - us-east-1	ami-0c56dba1b76993160
	US East (Ohio) - us-east-2	ami-0bcba95c5a8164f5e
	US West (N. California) - us-west-1	ami-023d04c383b5903a6
	US West (Oregon) - us-west-2	ami-0dcfffd3d76ea200f
	Asia Pacific (Mumbai) - ap-south-1	ami-0ea385e99ffd4ad8a

ACOS Image name	AMI IDs Region-wise	
	Region	AMI ID
	Asia Pacific (Seoul) - ap-northeast-2	ami-01120cd1470bd4644
	Asia Pacific (Singapore) - ap-southeast-1	ami-0214f89686cda73bc
	Asia Pacific (Sydney) - ap-southeast-2	ami-0bcac10dbc56bd271
	Asia Pacific (Tokyo) - ap-northeast-1	ami-00a211b5894da58d3
	Canada (Central) - ca-central-1	ami-01a87ce35e594b7be
	EU (Frankfurt) - eu-central-1	ami-04ddc2f4ce9dc0689
	EU (Ireland) - eu-west-1	ami-0b20d63398d2e9609
	EU (London) - eu-west-2	ami-09f1f5d1b11e5092e
	EU (Paris) - eu-west-3	ami-0a9b09fb26e5b45dd
	EU (Stockholm) - eu-north-1	ami-095130384df66ec57
	South America (Sao Paulo) - sa-east-1	ami-01b2447c4558176dd
A10 Thunder ADC for Advanced Load Balancing - 500 Mbps	US East (N. Virginia) - us-east-1	ami-0b7094be8f444c220

ACOS Image name	AMI IDs Region-wise	
	Region	AMI ID
	US East (Ohio) - us-east-2	ami-06d67aa35f8d81dd6
	US West (N. California) - us-west-1	ami-0b9d10512de4a6e6d
	US West (Oregon) - us-west-2	ami-0260f43bc696f8cc6
	Asia Pacific (Mumbai) - ap-south-1	ami-0fbcbad2497704878
	Asia Pacific (Seoul) - ap-northeast-2	ami-0aa4be343c328cac4
	Asia Pacific (Singapore) - ap-southeast-1	ami-04f3835442cd6fb56
	Asia Pacific (Sydney) - ap-southeast-2	ami-00c33f6a689dbf238
	Asia Pacific (Tokyo) - ap-northeast-1	ami-0511265bee25df093
	Canada (Central) - ca-central-1	ami-03429680279cbdbd8
	EU (Frankfurt) - eu-central-1	ami-0e12796942ad43527
	EU (Ireland) - eu-west-1	ami-018e10e1e92cce19f
	EU (London) -	ami-0f18a51fcb84eddcf

ACOS Image name	AMI IDs Region-wise	
	Region	AMI ID
	eu-west-2	
	EU (Paris) - eu-west-3	ami-0eb66723e49809658
	EU (Stockholm) - eu-north-1	ami-08031b29e5e2053ab
	South America (Sao Paulo) - sa-east-1	ami-091ede470704e0f86
A10 Thunder ADC for Advanced Load Balancing - 200 Mbps	US East (N. Virginia) - us-east-1	ami-039fbf7f122430a26
	US East (Ohio) - us-east-2	ami-09590c2061aabb3fb
	US West (N. California) - us-west-1	ami-0d050f5d252f4e77a
	US West (Oregon) - us-west-2	ami-09ee52609e782b6e8
	Asia Pacific (Mumbai) - ap-south-1	ami-062c969eaf80bd52e
	Asia Pacific (Seoul) - ap-northeast-2	ami-0f1523461f1bffdce
	Asia Pacific (Singapore) - ap-southeast-1	ami-03d0e57d867342b2b
	Asia Pacific (Sydney) - ap-southeast-2	ami-067802b694a355987

ACOS Image name	AMI IDs Region-wise	
	Region	**AMI ID**
	Asia Pacific (Tokyo) - ap-northeast-1	ami-08e6f99b5944bf40f
	Canada (Central) - ca-central-1	ami-05daf3aac68ac1594
	EU (Frankfurt) - eu-central-1	ami-0fb3a7eff2a704864
	EU (Ireland) - eu-west-1	ami-0d7c72d7e25d8ba6f
	EU (London) - eu-west-2	ami-0ba549740e10d9859
	EU (Paris) - eu-west-3	ami-07670bdd530cebb85
	EU (Stockholm) - eu-north-1	ami-03a7aadd7d984f81b
	South America (Sao Paulo) - sa-east-1	ami-0c264bd1b9a5fbf0b
A10 Thunder ADC for Advanced Load Balancing - 5 Gbps	US East (N. Virginia) - us-east-1	ami-0664c818a3f9b401e
	US East (Ohio) - us-east-2	ami-0fed8d8f8e1e49a51
	US West (N. California) - us-west-1	ami-0091064539cad0767
	US West (Oregon) - us-west-2	ami-0c24363284373e522
	Asia Pacific	ami-

ACOS Image name	AMI IDs Region-wise	
	Region	**AMI ID**
	(Mumbai) - ap-south-1	091d1810b7017698e
	Asia Pacific (Seoul) - ap-northeast-2	ami-08bc88334ba7d07d1
	Asia Pacific (Singapore) - ap-southeast-1	ami-0d4c7d32167284e1c
	Asia Pacific (Sydney) - ap-southeast-2	ami-0e4f1c60e5eda85d0
	Asia Pacific (Tokyo) - ap-northeast-1	ami-0511265bee25df093
	Canada (Central) - ca-central-1	ami-01e588555b35d5bc3
	EU (Frankfurt) - eu-central-1	ami-0ae34e90d482f02ae
	EU (Ireland) - eu-west-1	ami-05fc50eca69a53bde
	EU (London) - eu-west-2	ami-0a98e268fb5cf8dfe
	EU (Paris) - eu-west-3	ami-07ab5eec19e34bc8a
	EU (Stockholm) - eu-north-1	ami-078ebf9969f6cdc2d
	South America (Sao Paulo) - sa-east-1	ami-00721643e9d8b4f5b
A10 Thunder ADC for Advanced Load	US East (N.	ami-

ACOS Image name	AMI IDs Region-wise	
	Region	AMI ID
Balancing - 10 Gbps	Virginia) - us-east-1	0bfc326ee5d8dfe4b
	US East (Ohio) - us-east-2	ami-0f6c839b8a9d83e84
	US West (N. California) - us-west-1	ami-0639218db8b128c95
	US West (Oregon) - us-west-2	ami-0b2f7afde344dde0f
	Asia Pacific (Mumbai) - ap-south-1	ami-08c712c86ba571efa
	Asia Pacific (Seoul) - ap-northeast-2	ami-09f1fcef9e84ad598
	Asia Pacific (Singapore) - ap-southeast-1	ami-0cff9443ca2a44b62
	Asia Pacific (Sydney) - ap-southeast-2	ami-05682f746b33553c3
	Asia Pacific (Tokyo) - ap-northeast-1	ami-0074bc14bb0153943
	Canada (Central) - ca-central-1	ami-029977816a132e419
	EU (Frankfurt) - eu-central-1	ami-013c4cadeecbbd5ed
	EU (Ireland) -	ami-

ACOS Image name	AMI IDs Region-wise	
	Region	AMI ID
	eu-west-1	0577b007d2ea9355a
	EU (London) - eu-west-2	ami-0f47f257e60021f2c
	EU (Paris) - eu-west-3	ami-06569a738d70381fd
	EU (Stockholm) - eu-north-1	ami-0619d5d0735302d52
	South America (Sao Paulo) - sa-east-1	ami-043d6ad1ffe36dedc

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": "*"
        }
    ]
}
```

Supported Instance Types

Table 32 provides detailed information about the supported instance types.

Table 32 : 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

Instance	vCPU	Memory	Number of Network Interfaces
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
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

Instance	vCPU	Memory	Number of Network Interfaces
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
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

License Information

This product includes software covered by the Apache License.

For more information, see Apache License.

Support Information

For any issues or queries related to CFT templates, open a case at A10 Networks Support or reach out to support@a10networks.com and mention "A10-AWS-CFT-templates" in the subject line.

What's New

1.3.0

This release has the following enhancements for Thunder® Application Delivery Controller (ADC):

- Added a template for creating new virtual private cloud (VPC) containing six new subnets and two new security groups (SGs) in same or different availability zone in the same region.
- Added a template for deploying two vThunder instances in the same or different availability zones, including resources such as three network interface cards, Private IPs (Alien IPs), and Elastic IPs.
- Added new HA Across AZs configuration to enable high availability across availability zones.

1.2.0

This release has the following enhancements for Thunder® Application Delivery Controller (ADC):

- Added support for ACOS 6.0.2, ACOS 6.0.1, and ACOS 5.2.1-P8.
- Added a template for creating new virtual private cloud (VPC) containing three new subnets and two new security groups (SGs).
- Added a template for creating new elastic public IP addresses.
- Separated the deployment and configuration parameters to ensure a clear distinction between the resources needed for initial deployment and those required for subsequent configuration and customization.
- Introduced two new SLB templates, SLB HTTP and Persist Cookie to enhance the functionality and performance of the Server Load Balancer (SLB) by optimizing HTTP traffic distribution and implementing efficient cookie persistence.

- Added new hybrid cloud GSLB configuration to optimize performance, reliability, and ease of use in hybrid cloud environments.

1.1.0

This release has the following enhancements for Thunder® Application Delivery Controller (ADC):

- Added support for ACOS 5.2.1-P7, ACOS 6.0.0-P1, and ACOS 6.0.0-P2.
- Added Thunder password change capability.

1.0.0

This release has the following enhancements for Thunder® Application Delivery Controller (ADC):

- Added support for ACOS 5.2.1-P6.
- Added GLM, HA, SLB, and SSL vThunder configuration.
- Added the following deployment templates:
  - A10-vThunder_ADC-2NIC-1VM-GLM
  - A10-vThunder_ADC-2NIC-1VM
  - A10-vThunder_ADC-3NIC-2VM-HA-GLM-PUBVIP-BACKAUTO
  - A10-vThunder_ADC-3NIC-2VM-HA-GLM-PVTVIP
  - A10-vThunder_ADC-3NIC-2VM-HA
  - A10-vThunder_ADC-3NIC-6VM-2RG-GSLB
  - A10-vThunder_ADC-3NIC-VMSS

Contact Us

©2024 A10 Networks, Inc. All rights reserved. A10 Networks, the A10 Networks logo, ACOS, A10 Thunder, Thunder TPS, A10 Harmony, SSLi and SSL Insight are trademarks or registered trademarks of A10 Networks, Inc. in the United States and other countries. All other trademarks are property of their respective owners. A10 Networks assumes no responsibility for any inaccuracies in this document. A10 Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice. For the full list of trademarks, visit: www.a10networks.com/company/legal/trademarks/.