Deploy the Threat Defense Virtual Auto Scale Solution on Alibaba Cloud

The following sections describe how the Threat Defense Virtual Auto Scale solution is implemented on the Alibaba Cloud.

- About the Auto Scale Solution, on page 1
- Guidelines and Limitations, on page 2
- Use Case, on page 2
- Download the Deployment Package, on page 3
- Solution Components, on page 3
- Licenses, on page 5
- Prerequisites, on page 5
- Input Parameters, on page 6
- Deploy the Auto Scale Solution, on page 11
- Verify Deployment, on page 20
- Enable Memory Metrics Trigger for Threat Defense Virtual Deployment, on page 20
- Monitor Memory Metrics, on page 21
- Disable Auto Scaling, on page 21
- Post-Deployment Logs, on page 21
- Troubleshoot, on page 22

About the Auto Scale Solution

The auto scale solution helps organisations to automatically scale up the number of Threat Defense Virtual instances if there is a spike in traffic and also scale down the number of instances during a lull in traffic. This solution leads to efficient handling of network resources, improves high availability, and reduces operational costs.

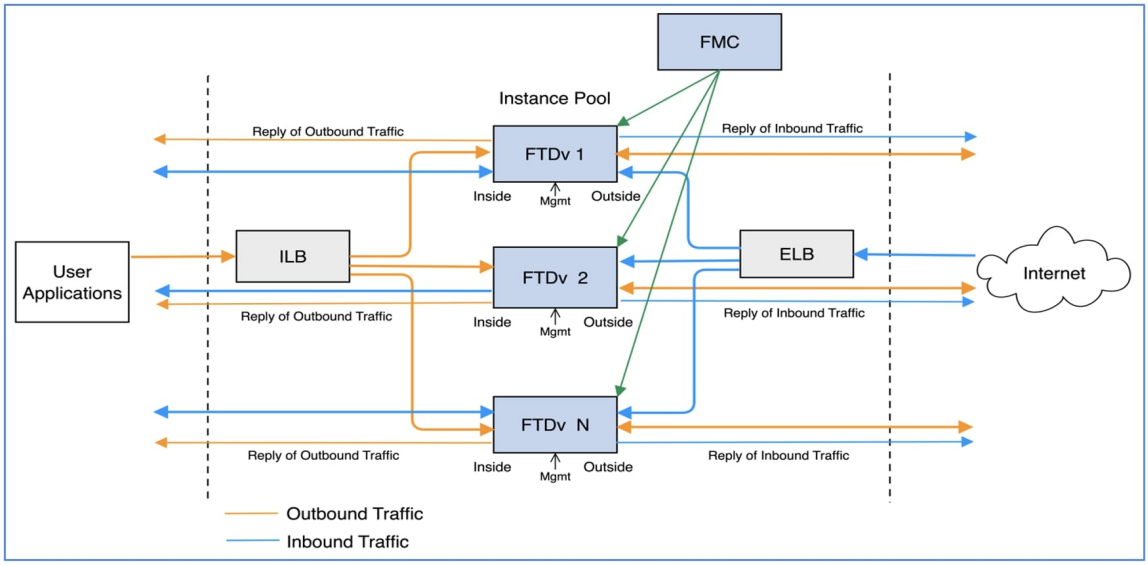
Starting from Secure Firewall version 7.4.1, Cisco provides terraform templates to enable deployment of a horizontal auto scaling solution for Threat Defense Virtual firewalls deployed in an Alibaba Cloud environment. The auto scaling solution enables an automated increase or decrease in the number of Threat Defense Virtual firewalls based on CPU and memory utilisation.

Guidelines and Limitations

- Only IPv4 is supported.
- The customer's auto scaling infrastructure requirement has to fit in the sandwich model of implementation.
- Reboot of any Threat Defense Virtual instance in the Scaling Group may result in packet loss.
- Any specific DNS requirements of the load balancer's public IP address are out of scope as the external load balancer is created by the terraform template.
- Device logs obtained by using the **show tech** command do not contain auto scale logs. The auto scale logs can be checked in the Alibaba Function Compute logs.
- Schedule based scaling is not supported.
- FDM and CSM is not supported.

Use Case

The Threat Defense Virtual Auto Scale for Alibaba Cloud is an automated horizontal scaling solution that positions an Threat Defense Virtual instance group sandwiched between an Internal load balancer (ILB) and an External load balancer (ELB).



Inbound traffic (Internet->ELB->FTDv->Inside-Application)

Outbound traffic (Application->ILB->FTDv->Outside-Web-Server)

- The ELB distributes traffic from the Internet to Threat Defense Virtual instances in the instance group; the firewall then forwards traffic to the application.
- The ILB distributes outbound Internet traffic from an application to Threat Defense Virtual instances in the instance group; the firewall then forwards traffic to the Internet.

- The Threat Defense Virtual instances are managed by the connected management center virtual.
- The number of Threat Defense Virtual instances in the scale set will be scaled and configured automatically based on CPU utilization.

Download the Deployment Package

The Threat Defense Virtual Auto Scale for Alibaba Cloud solution is a template-based deployment that makes use of the serverless infrastructure provided by Alibaba Cloud (Function Compute, Resource Orchestration Service (ROS), Event Bridge, Object Storage Service (OSS), etc.).

Download the following deployment scripts and templates that are required to launch the Threat Defense Virtual Auto Scale for Alibaba Cloud solution from the GitHub repository:

- *main.tf* and *variables.tf* function terraform templates - Used to set up Alibaba Cloud resources
- *main.tf* and *function.tf* auto scale terraform templates - Used to set up Auto scale functionality
- *fmc_functions.py*, and *index.py* functions in the *scalein* folder - Contains parameters to set up the scale-in functionality
- *alibaba_lib.py*, *fmc_functions.py*, *basic_functions.py*, and *index.py* functions in the *scaleout* folder - Contains parameters to set up the scale-out functionality
- *alibaba_lib.py*, *fmc_functions.py*, and *index.py* functions in the *memory_metrics* folder - Contains parameters to set up collection of memory metrics

Note Cisco-provided deployment scripts and templates for auto scale are provided as open source examples, and are not covered within the regular Cisco TAC support scope.

Solution Components

The following components make up the Threat Defense Virtual Auto Scale for Alibaba Cloud solution.

Resource Orchestration Service

Alibaba Cloud Resource Orchestration Service (ROS) is a fully managed service for cloud computing resource orchestration and automated deployment. Resource Orchestration Servvice (ROS) is used to manage multiple stacks as a single unit called a Stack group with high efficiency and cost-effectiveness. ROS uses terraform templates to create multiple stacks across accounts and regions and enables unified stack deployment in different directory folders and accounts.

Cisco provides terraform templates that can be deployed using Alibaba Cloud ROS. These templates create the following resources on Alibaba. These resources are used to set up the auto scale solution.

- Elastic Compute Service (ECS) Launch Template
- Elastic Scaling Service (ESS) Scaling Group
- ESS Alarms for adding and removing instances from scaling groups

- Server load balancer (SLB) for deploying external load balancer (ELB) and internal load balancer (ILB)
- SLB vServer groups, and listeners
- NAT gateway
- Elastic IP address
- Route Table
- Function Compute Service and functions for scale-out and scale-in actions
- Scale-out and scale-in action triggers
- Log Project and Log Store

Auto Scaling Group

A scaling group is a basic management unit in Auto Scaling. It is used to manage Elastic Compute Service (ECS) instances that are applied to similar scenarios and can be associated with multiple Server Load Balancer (SLB) instances. After a scaling group is associated with SLB, ECS instances that are added to the scaling group are automatically added as backend servers of the associated SLB instances.

The internal IP addresses of these instances are automatically added to the Allow lists of the associated RDS instances. A scaling configuration is a template used by Auto Scaling to automatically create ECS instances. A scaling rule is used to specify information such as the number of ECS instances to be scaled or to set the boundary values of a scaling group. After a scaling rule is created, a scaling task can automatically execute the scaling rule.

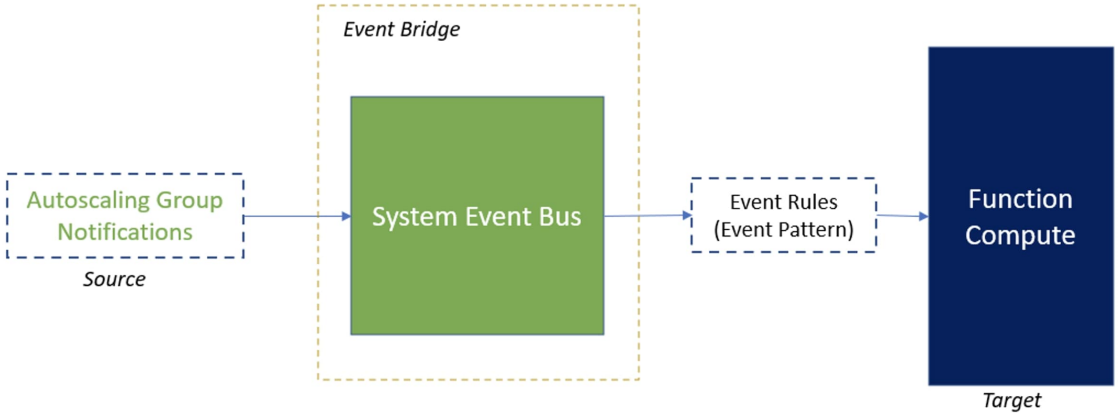
To automatically scale ECS instances based on their running metrics, use an event-triggered task that dynamically manages ECS instances in a scaling group based on monitoring metrics from CloudMonitor and EventBridge.

Classic Load Balancer

A Classic Load Balancer (CLB) distributes inbound network traffic across multiple backend servers based on forwarding rules. CLB uses virtual IP addresses to provide load balancing services for the backend pool, which consists of servers deployed in the same region. Network traffic is distributed across multiple backend servers based on forwarding rules. This ensures the performance and availability of applications. CLB monitors the health of backend servers and does not distribute network traffic to unhealthy backend servers. This eliminates single points of failure (SPOFs) and improves the availability of applications.

EventBridge

EventBridge is a serverless event bus service provided by Alibaba Cloud that can be accessed from other Alibaba Cloud services, custom applications, and software-as-a-service (SaaS) applications. EventBridge can also route events between these services and applications based on the standard CloudEvents 1.0 specification. EventBridge can be used to build loosely coupled and distributed event-driven architectures. The following figure shows the event flow in Event Bridge.



- The event Source publishes events that are generated by Alibaba Cloud services, custom applications, and SaaS applications to the System Event Bus.
- The EventBridge stores and routes received events to the event Target based on the Event Rule.
- The Event target received the events.

Function Compute

Alibaba Cloud Function Compute is a fully-managed event-driven computing service that enables you to focus on writing and uploading code, without the need to procure and manage infrastructure resources such as servers. Function Compute prepares computing resources, runs code, and provides features such as log query, performance monitoring, and alert. Function Compute integrates different services in an event-driven manner. Based on this, you can create elastic, reliable, and secure applications and services, and even complete a set of backend services for processing multimedia data within a few days. When an event source triggers an event, the associated function is automatically called to process the event. See Function Compute for more information.

Layers

Layers are used to publish and deploy custom resources such as common dependencies, runtime environments, and function extensions. The public libraries can be extracted from them and functions that depend on layers are reduced in terms of sizes of code packages when deployed.

When you build a layer, you need to package the content into a ZIP file. The Function Compute runtimes decompress and deploy the content of the layer in the /opt directory. When multiple layers are configured for a function, the content of these layers is merged and stored in the /opt directory in reverse order.

Licenses

The auto scale solution is supported with the BYOL licensing model. Ensure that the number of licenses that are reserved is greater than or equal to *max_ftd_count*. Select the entitlement based on the instance shape.

Prerequisites

- Ensure that network infrastructure such as VPCs, vSwitches, Threat Defense Virtual images, subnets and their respective security groups, and routes are available.

- Create VCN and Subnets as these are used by their applications and other elements in the resource group.
- Set up the Management Center Virtual with the required configuration.

Note The procedures to configure these prerequisites are given as part of the Deploy the Auto Scale Solution procedure.

Input Parameters

Cisco provides Terraform templates to enable deployment of a horizontal auto-scaling solution for Threat Defense Virtual firewalls deployed in the Alibaba Cloud environment.

The parameters in the main.tf function template file are given below.

Parameters	Description
resource "alicloud_log_project"	The project is the resource management unit in Log Service and is used to isolate and control resources. You can manage all the logs and the related log sources of an application by using projects.
resource "alicloud_log_store"	The log store is a unit in Log Service to collect, store, and query the log data. Each log store belongs to a project, and each project can create multiple Logstores
resource "alicloud_fc_service"	Provides an Alicloud Function Compute Service resource. The resource is the base of launching Function and Trigger configuration
resource "alicloud_log_store_index"	LogService provides the LogSearch/Analytics function to query and analyze large amounts of logs in real time. You can use this function by enabling the index and field statistics.
resource "alicloud_fc_function"	Provides an Alicloud Function Compute Function resource. This allows you to trigger execution of code in response to events in Alibaba Cloud. The Function itself includes the source code and runtime configuration.
resource "alicloud_fc_trigger"	Provides an Alicloud Function Compute Trigger resource. Based on the trigger, code execution is initiated in response to events in Alibaba Cloud.

The parameters in the variables.tf function template file are given below.

Parameters	Description
access_key and secret_key	These are the access keys and secret keys to the Alibaba Cloud account.

Parameters	Description
region	The region in which you deploy the auto scale solution.
account_id	The Enterprise Alias ID given by Alibaba Cloud.
diag_vswitch	The vSwitch ID for the diagnostic interface.
inside_vswitch and outside_vswitch	The vSwitch IDs for the inside and outside interfaces.
mgmt_vswitch	The vSwitch ID for the management interface.
scaling_group_name	Name of the auto scaling group deployed in the auto scale solution.
security_group_id	This is created for the VPC that is set up.
elb_name and ilb_name	Name of the external and internal load balancers.
fmc_ip	This is the private IP address of the management center virtual.
fmc_scaleout_username	The user name for the scaleout function. By default, this is set to scaleout.
fmc_scaleout_password	The password for the scaleout function.
fmc_scalein_username	The user name for the scalein function. By defult, this is set to scalein.
fmc_scalein_password	The password for the scalein function.
memory_metrics_password and memory_metrics_username	The user name for the memory metrics function. By default, this is set to scaleout.
fmc_group_name	The group name for the management center virtual.
fmc_access_policy	The management center virtual access policy that is created by you.
inside_gateway_name and outside_gateway_name	The names for the inside and outside gateways. By default, this is set to inside_gateway and outside_gateway.
inside_security_zone_name and outside_security_zone_name	The names for the inside and outside security zone. By default, this is set to in and out.
ftd_password	The Threat Defense Virtual password specified by you.
ftd_license_caps	The list of Threat Defense Virtual licenses that are available to use. By defeault, this is set to BASE, MALWARE, URLFilter, THREAT.
ftd_reg_id and ftd_nat_id	The registration ID and Network Address Translation (NAT) ID specified during registration of the Threat Defense Virtual instance.

Parameters	Description
memory_metrics_group_id	Memory metrics publishing group in which the memory metric function publishes metrics from the Management Center Virtual.
metric_name	Metric name in which the memory metric function publishes metrics from the Management Center Virtual.
Scaleout_threshold	The period, in seconds??, after which the scaleout function is triggered. By default, this is set to 30.
log_project_name and log_store_name	The names specified for the project log and the log storage file.
fc_service_name	The name of the service under which the function should be present.
fc_scalein_fc_name and fc_scaleout_fc_name	The scalein and scaleout function names.
fc_memory_metrics_fn_name	The memory metrics function name.
function_bucket	Specifies the name of the bucket in which the zip files are present.
scaleout_function_zip_key and scalein_function_zip_key	The zip keys that are used to unzip the scaleout and scalein function zip files.
memory_metrics_function_zip_key	The zip keys that is used to unzip the memory metrics function zip files.
launch_template_name	Specifies the name of the terraform template that is used for deployment.
scaleout_trigger and scalein_trigger	Specifies the scaleout and scalein trigger names.
memory_metrics_trigger	Specifies the memory metrics trigger name.

The parameters in the main.tf auto scale template file are given below.

Parameters	Description
resource "alicloud_ecs_launch_template"	Provides a ECS Launch Template resource.
resource "alicloud_ess_scaling_group"	Provides a ESS scaling group resource which is a collection of ECS instances with the same application scenarios. It defines the maximum and minimum numbers of ECS instances in the group, and their associated Server Load Balancer instances, RDS instances, and other attributes.
resource "alicloud_ess_scaling_rule"	Provides a ESS scaling rule resource.
resource "alicloud_ess_alarm"	Provides a ESS alarm task resource.
resource "alicloud_slb_load_balancer"	Provides an Application/Server Load Balancer resource.

Parameters	Description
resource "alicloud_slb_server_group"	A virtual server group contains several ECS instances. The virtual server group can help you to define multiple listening dimension, and to meet the personalized requirements of domain name and URL forwarding.
resource "alicloud_slb_listener"	Provides an Application Load Balancer Listener resource.
resource "alicloud_nat_gateway"	Provides a resource to create a VPC NAT Gateway.
resource "alicloud_eip_address"	Provides a EIP Address resource.
resource "alicloud_eip_association"	Provides an Alicloud EIP Association resource for associating Elastic IP to ECS Instance, SLB Instance or Nat Gateway.
resource "alicloud_snat_entry"	Provides a SNAT resource
resource "alicloud_route_table"	Provides a route entry resource. A route entry represents a route item of one VPC route table.
resource "alicloud_route_entry"	Provides a route entry resource. A route entry represents a route item of one VPC route table.
resource "alicloud_route_table_attachment"	Provides an Alicloud Route Table Attachment resource for associating Route Table to VSwitch Instance/NAT Gateway.

The parameters in the variables.tf auto scale template file are given below.

Table 1:

Variables	Description
access_key and secret_key	These are the access keys and secret keys to the Alibaba Cloud account.
region	The region in which you deploy the auto scale solution.
image_id	Specifies the Threat Defense Virtual image ID.
user_data	Base64 encoded format of Day 0 configuration – Must be applied to every instance that is to be deployed. This can be changed, if required.
instance_type	Threat Defense Virtual instance type. For example, ecs.g5ne.xlarge.
internet_charge_type	The usage payment terms as per your requirement. By default, this is set to ‘PayByBandwidth’.
internet_max_bandwidth_in and internet_max_bandwidth_out	The maximum input and output bandwidth. By default, this is set 25 and 20 Mbps (is the unit correct?) respectively.

Variables	Description
network_type	Specifies the network type. In this case, it is specified as vpc.
security_group_id	Specifies the security group ID. This is set during creation of VPC.
system_disk_name	Specifies the name of the system disk.
delete_with_instance_flag	When this is set to true, it enables you to delete the boot disk with the instance flag.
launch_template_name	The name of the deployment template.
inside_vswitch_id and outside_vswitch_id	The vSwitch IDs for the inside and outside interfaces.
mgmt_vswitch_id	The vSwitch ID for the management interface.
vpc_id	The ID that is created during the creation of the VPC.
zone_id	This specifies the region in which the instance is created.
scaling_group_name	Name of the auto scaling group deployed in the auto scale solution.
scaleout_rule_name and scalein_rule_name	Specifies the rule names for the scaleout and scalein functions.
min_ftd_number and max_ftd_number	Minimum and maximum number of Threat Defense Virtual instances that can be deployed as part of the auto-scaling group – These values can be updated as required. By default, this is set to 2 and 3 instances respectively.
cpu_threshold_scaleout and cpu_threshold_scalein	Maximum and minimum CPU utilisation values – Scale out and scale in actions are triggered based on these values. These values can be updated as per the end user's requirements. By default, this is set to CPU usage at 70% and 20% respectively.
scaleout_cooldown_period	A period after the scale out activity is initiated during which the scaling function is blocked. By default, this is set to 120 seconds.
scalein_cooldown_period	A period after the scale in activity is initiated during which scaling function is blocked. By default, this is set to 120 seconds.
memory_threshold_scaleout and memory_threshold_scalein	Maximum and minimum memory utilisation values – Scale out and scale in actions are triggered based on these values. These values can be updated as per the end user's requirements.
add_instance_cpu_event_name and rm_instance_cpu_event_name	Specifies the addition and removal of CPU event names in the scaling group.

Variables	Description
add_instance_memory_event_name and rm_instance_memory_event_name	Specifies the addition and removal of the memory function's event names in the scaling group.
memory_metrics_group_id	Memory metrics publishing group in which the memory metric function publishes metrics from the Management Center Virtual
memory_metrics_name	The memory metrics function name.
elb_name and ilb_name	Name of the external and internal load balancers.
elb_vserver_group_name and ilb_vserver _group_name	Specifies of the ELB and ILB virtual server names.
nat_gateway_name	Specifies the name of the NAT gateway.
route_table_name	Specifies the NAT gateway's route table name.
alicloud_eip_address_name	Specifies the external IP address name for the AliBaba cloud.

All VPCs, instances, and the OSS bucket, have to be in the same region for the scripts to work as intended. Ensure that you do not change the variable names as the auto scale scripts are dependent on these names. You can only change the values of the variables.

Deploy the Auto Scale Solution

Perform the steps given below to deploy the auto scale solution.

Procedure

Step 1 Clone the Git Repository to the local folder.

```
git clone <Git-Repository-URL> -b <branch-name>
```

Step 2 Create the Object Storage Service Bucket (OSS).

a) On the Alibaba Cloud OSS console, select **Object Storage Service (OSS)** in the left navigation pane.
b) Click the **Buckets** tab.
c) Click **Create Bucket** and enter the **Bucket Name** along with any other required details in the dialog box.
d) Click **OK** to create the OSS bucket.

Step 3 Set up the Virtual Private Cloud and vSwitches.

a) On the Alibaba cloud VPC console, in the top navigation bar, choose the region in which you want to create a VPC and a vSwitch.
b) On the VPC page, click **Create VPC** and enter the **VPC Name**, **IPv4 CIDR block**, and any other required details.
c) Scroll below to the **VSwitch** section and click **+Add** to add vSwitches for management, diagnostic, inside, and outside interfaces.
d) Click **OK** to create the VPC.

Step 4 Create the layers given below.

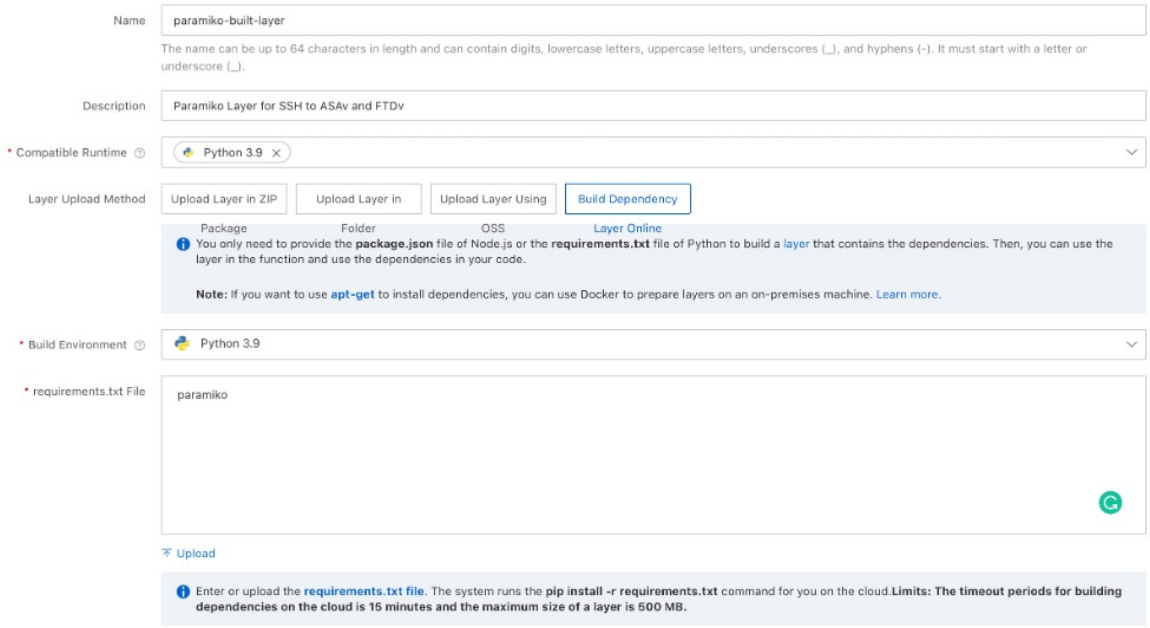
Layers to be added to the scale-out function-

- aliyun-python-sdk-slb
- paramiko-built-layer

Layers to be added to the memory_metrics function-

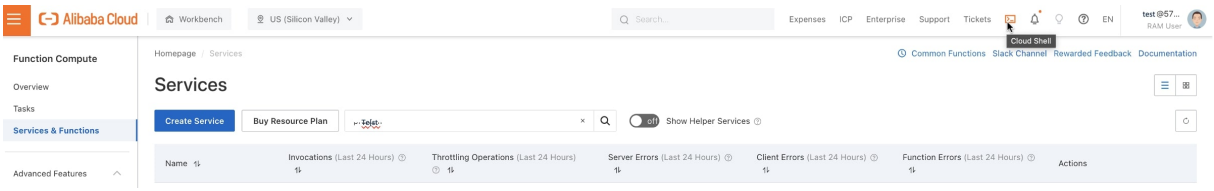
- aliyun-python-sdk-cms
- aliyun-python-sdk-ess

a) Log in to the Alibaba Cloud Function Compute Console and go to **Advanced Features > Layers**.
b) Choose a region in the top navigation bar.
c) On the **Layers** page, click **Create Layer**.
d) Enter a **Name** for the layer along with a **Description**.
e) In the **Compatible Runtime** field, choose **Python 3.9**.
f) For the **Layer Upload Method**, choose **Build Dependency**.
g) From the **Build Environment** drop-down list, choose **Python 3.9**.
h) In the **requirements.txt** File field, enter **paramiko** and click **Upload** to upload the requirements.txt file and create the paramiko-built-layer layer. Similarly, create layers for slb (scaleout function layer), ess-built, and cms-buil (memory metrics function layer).



Step 5 Create the memory metric group.

a) Go to the Alibaba Cloud Function Compute console and click the **Cloud Shell** icon on the top right of the page.



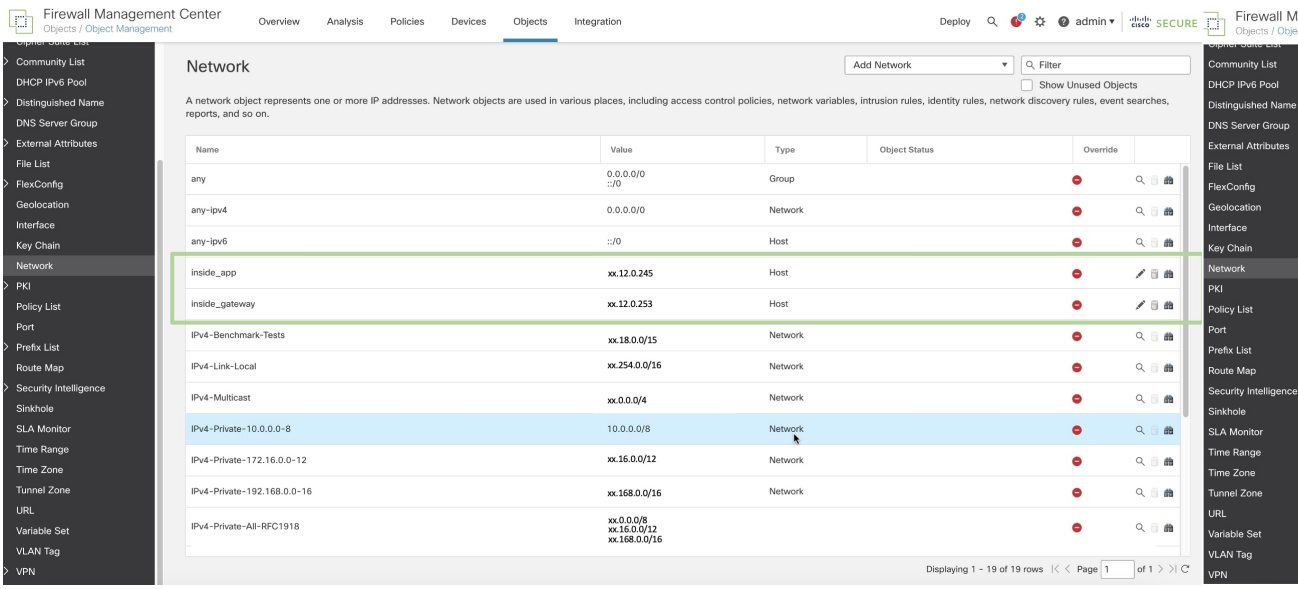
b) In the Cloud Shell terminal window, use the command given below with the required *GroupId* and *MetricName* to create the memory metric group.

```
shell@Alicloud:~$ aliyun cms PutCustomMetric --region us-west-1 --MetricList.1.GroupId 22722
--MetricList.1.MetricName memory --MetricList.1.Dimensions '{"time":"minutes"}'
--MetricList.1.Type 0 --MetricList.1.Values '{"value":1}'
{
        "Code": "200",
        "Message": "success",
        "RequestId": "EFBEAF7C-EDE7-3183-BAAF-A7E06643F02D”
}
```

Step 6 Go to the management center virtual and configure the components given below. For more information on configuring these components, see Management Center Device Configuration Guide or the Management Center REST API Quick Start Guide.

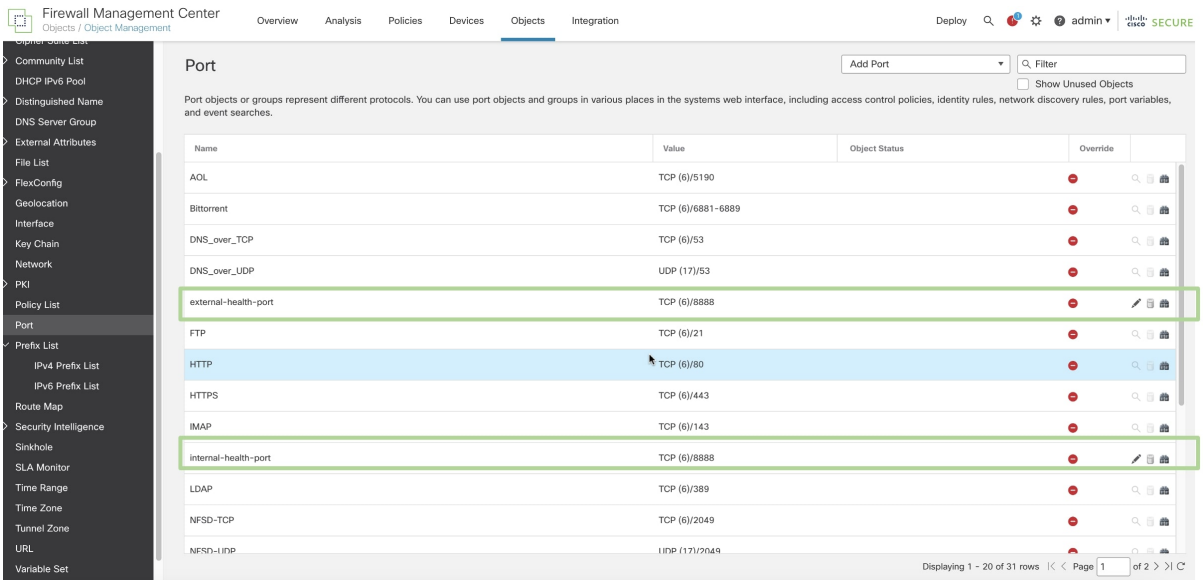
- Security Zones for the inside and outside interfaces
- Access Policy and the required Access Rules
- Device Group in management center virtual for the registration of Threat Defense Virtual instances
- Network objects for the *inside_app*, *inside_gateway*, *outside_gateway*, *outside-ext-app*, and metadata server with the IP address 100.100.100.200

  Note
  The *inside_app* is the traffic interface IP address of the internal application server. The *outside-ext-app* is the public IP address of the external application server.



- Three usernames for the management center virtual (scaleout, scalein, and memory) and login using the admin username to monitor the activities in the management center virtual.

- Port object creation for the Health Check Packets NAT Rule
  - external_health_check_port
  - internal_health_check_port



- NAT policy
- NAT policy and device group association
- NAT rules required for the traffic flow and health check.

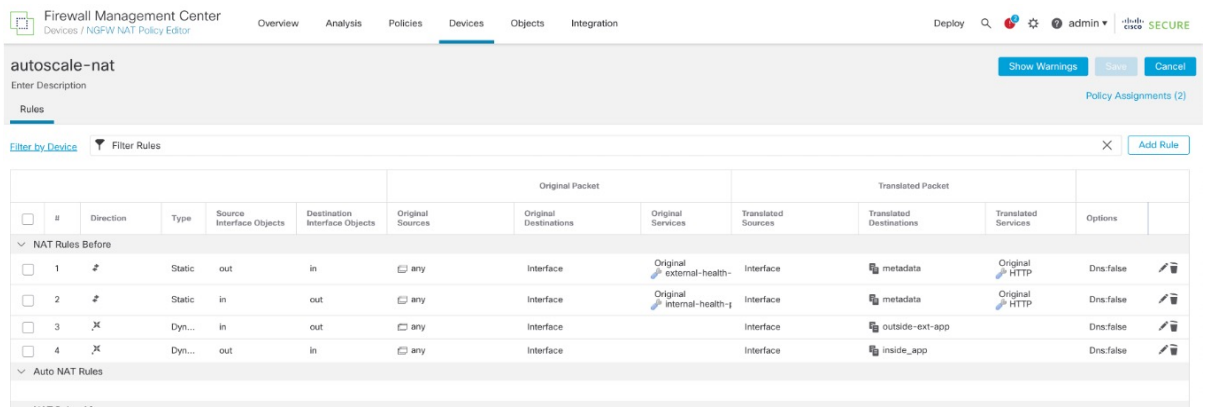
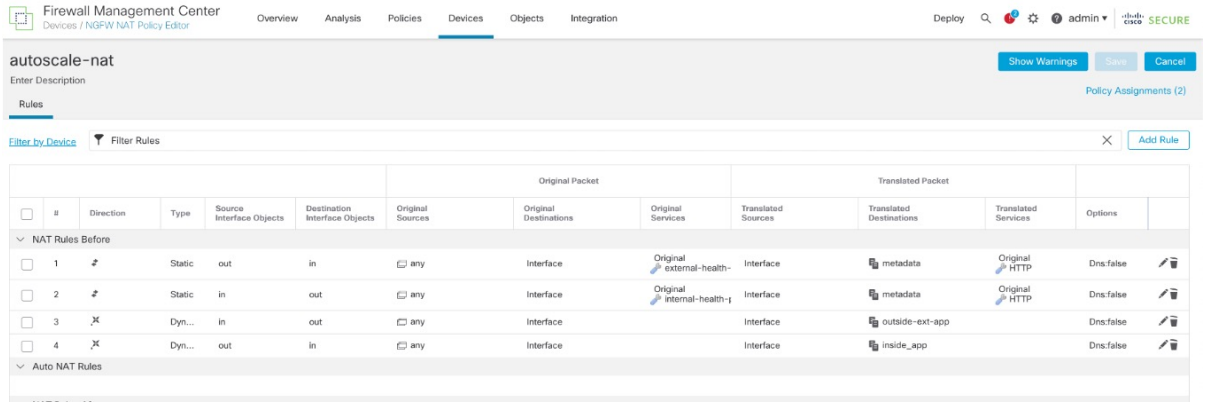


Figure 1: Sample Parameters for NAT rules 1 and 4

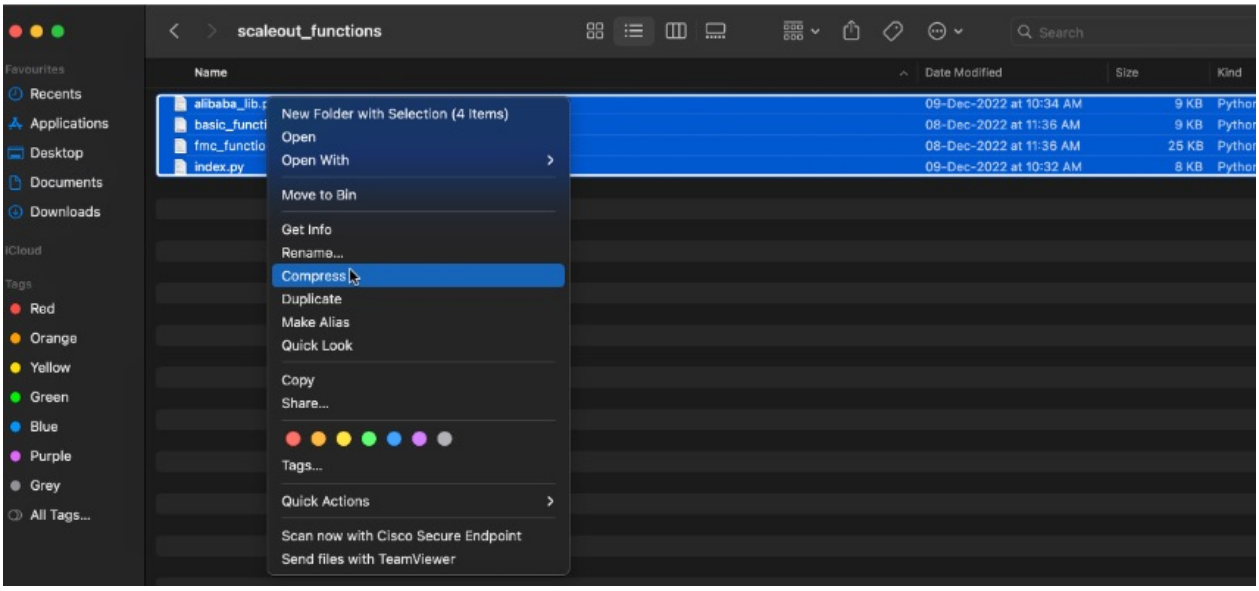


Step 7 Update the variables in the following templates to ensure that the auto scale solution is deployed using the required VPC and custom variable names.

- cloud_autoscale/Alibaba/FTDv/functions_deployment/variables.tf
- cloud_autoscale/Alibaba/FTDv/autoscale_deployment/variables.tf

See Input Parameters, on page 6 for more information on the template variables.

Step 8 Compress or zip the files in the scaleout_functions folder and rename the compressed or zipped file as scaleout_action.zip. Similarly, create the scalein_action.zip and memory_metrics_action.zip files.



The files in each of the zip files is given below.

- scaleout_action.zip:
  - index.py
  - alibaba_lib.py

- fmc_functions.py
- basic_functions.py

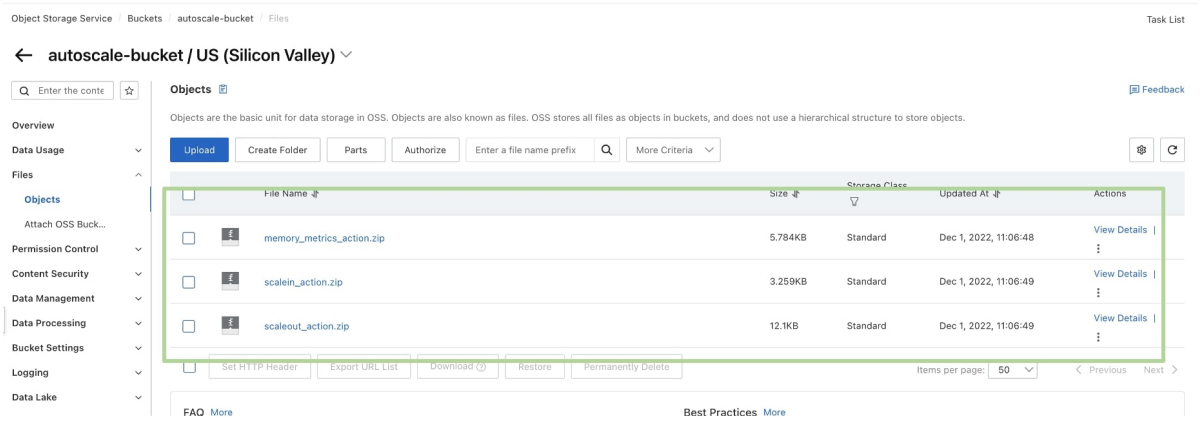
- scalein_action.zip:
  - index.py
  - fmc_functions.py

- memory_metrics_action.zip:
  - index.py
  - fmc_functions.py
  - alibaba_lib.py

Step 9 Upload the scalein_action.zip, scaleout_action.zip, and memory_metrics_action.zip files to the OSS Bucket.

a) On the Alibaba Cloud Object Storage Service console, go to **Buckets** > *OSS bucket created by you* > **Files** > **Objects**.

b) Choose the zip files that have to be uploaded from the local folder and click **Upload**. Ensure that the files are zipped and not in a folder.



Step 10 Create terraform templates.

a) In the Alibaba Cloud ROS console, click **Templates > My Templates**.

b) Click **Create Template**.

c) Click **Terraform > Open File**.

d) Select **main.tf** and **variables.tf**, and click **Open**.

e) Click **Save Template > Save as My Template**.

f) In the **Template Name** field on the **Save as My Template** window, enter **ftdv_functions**.

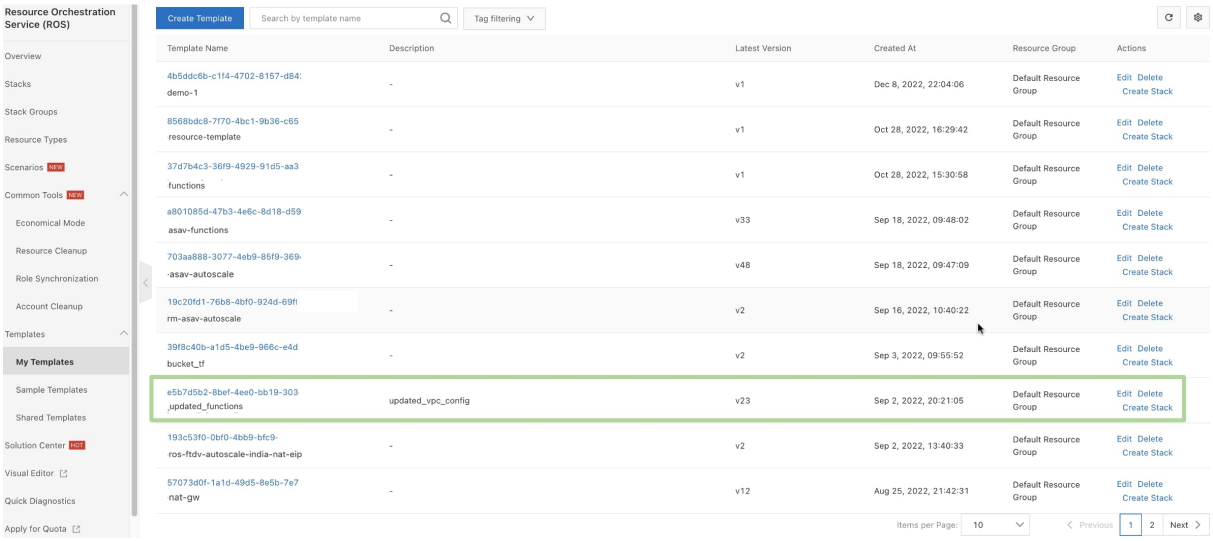
g) Click **OK**.

Repeat steps 10a) to 10g) and create the autoscale terraform template.

Step 11 Create the Functions stack.

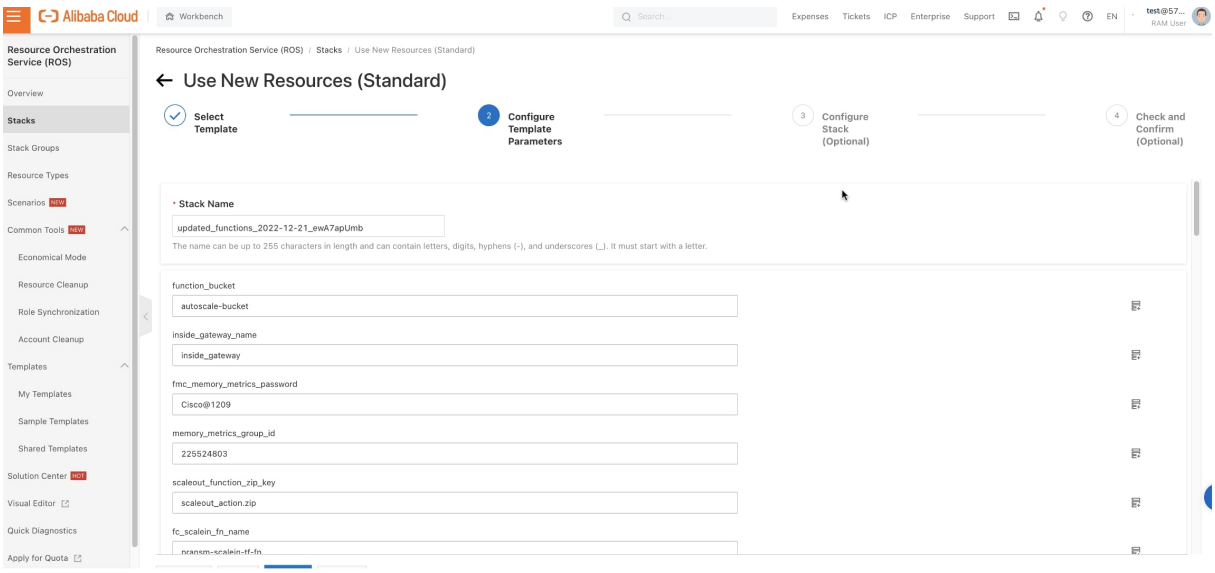
a) In the Alibaba Cloud ROS console, click **Templates > My Templates**.

b) Click **Create Stack** in the row mentioning the name of the functions template that you created.



c) In the **Use New Resources (Standard)** page, enter the **Stack Name**.

d) (Optional) You can change the values of the variables in the template as per your requirements.



e) Click **Create**.

f) (Optional) Click the **Events** tab and toggle **Auto-refresh** to see the creation of the stack resources in real-time. You can also click the **Refresh** icon to refresh the status of the events on this page. After all the resources have been created, you will see **Created** next to **Status** in the **Stack Information** tab which signifies that the stack has been created.

Step 12 Add layers to the functions stack. The Threat Defense Virtual auto-scale solution requires the layers given below.

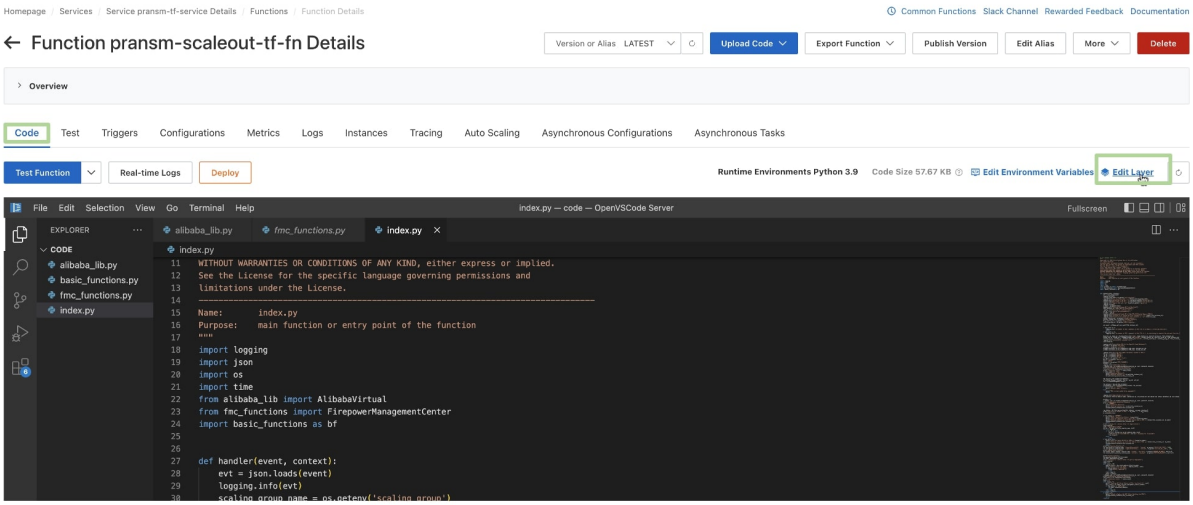
Scale out function-

- aliyun-python-sdk-slb
- paramiko-built-layer

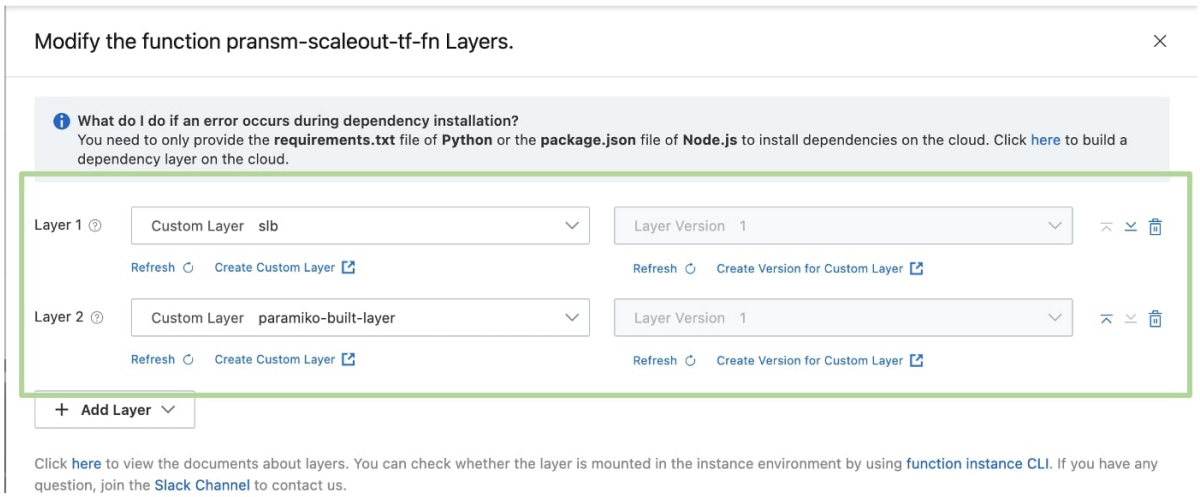
Memory metrics function-

- aliyun-python-sdk-cms
- aliyun-python-sdk-ess

a) In the Alibaba Cloud Function Compute console, click **Services and Functions**.
b) Click the service name to display the scale-out and scale-in functions.
c) Click the scale-out function name.
d) In the **Code** tab, click **Edit Layer**.

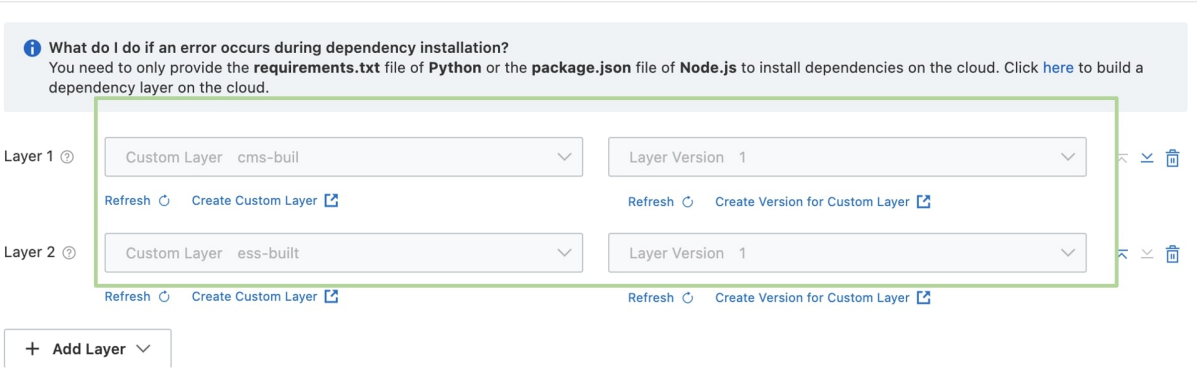


e) Click **Add Layer** > **Add Custom Layer**.
f) Add the slb and paramiko-built-layer layers from the **Layer 1** and **Layer 2** drop-down lists.



g) Click **OK**.
h) Click the memory metrics function name.
i) In the **Code** tab, click **Edit Layer**.
j) Click **Add Layer** > **Add Custom Layer**.

k) Add the cms-build and ess-built layers from the **Layer 1** and **Layer 2** drop-down lists.



l) Click **OK**.

Step 13 Create auto scale rules for all the three functions - scaleout_action, scalein_action, and memory_metrics_action, to ensure that only one function is initiated at a time.

a) In the Alibaba Cloud compute console, click **Services and Functions**.
b) Click the service name to display the functions.
c) Click the scale out function name.
d) In the **Auto Scaling** tab, click **Create Rule**.
e) Choose **LATEST** as the version.
f) Enter 0 in the **Minimum Instance Count** field and 1 in the **Maximum Instance Count** field.
g) Click **Create Rule**.

Similarly, create rules for the scale in and memory metrics functions.

Step 14 Create the Auto Scale Stack.

a) In the Alibaba Cloud ROS console, click **Templates > My Templates**.
b) Click **Create Stack** in the auto scale template row.

c) In the **Use New Resources (Standard)** page, enter the **Stack Name** and any other required parameters.

d) Click **Create**.
e) (Optional) Click the **Events** tab and toggle **Auto-refresh** to see the creation of the stack resources in real-time. You can also click the **Refresh** icon to refresh the status of the events on this page. After all the resources have been created, you will see **Created** next to **Status** in the **Stack Information** tab which signifies that the stack has been created.

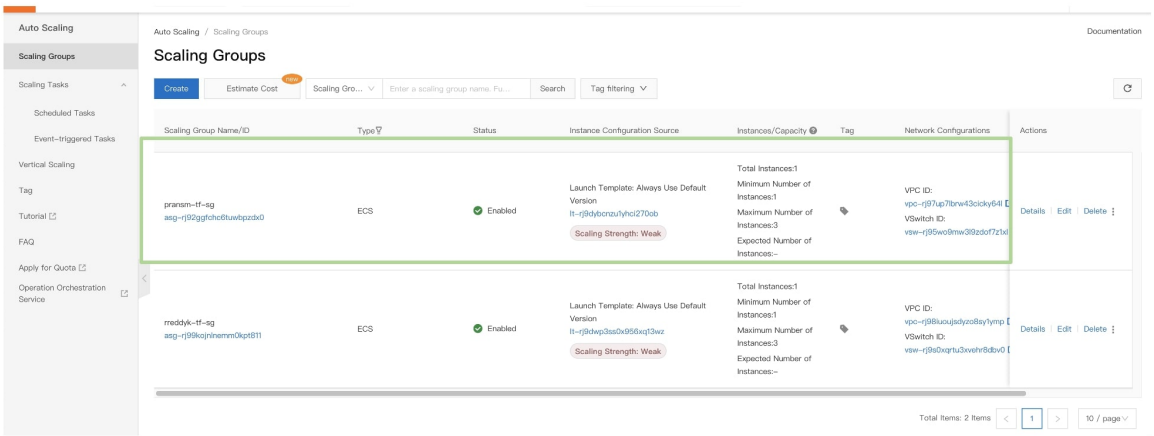
You have now created all the required resources and deployed the Threat Defense Virtual for Alibaba Cloud auto scale solution.

What to do next

Enable the memory metrics trigger and set up custom monitoring of the memory metrics.

Verify Deployment

In the Alibaba Cloud Auto Scaling console, click **Auto Scaling > Scaling Groups** to display the deployed auto scaling group.



Enable Memory Metrics Trigger for Threat Defense Virtual Deployment

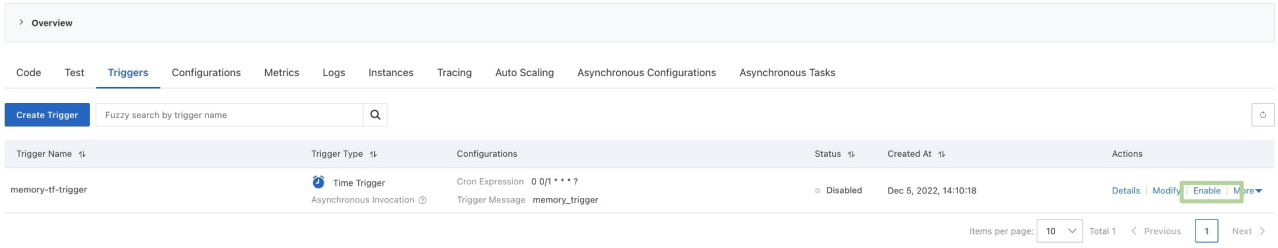
Enable the memory metric functions trigger by performing the steps given below. The trigger is a function that is enabled when certain network conditions are met. You can enable this trigger only after creating the scaling group and deploying the auto-scale terraform template.

Procedure

Step 1 In the Alibaba Cloud compute console, click **Services and Functions**.
Step 2 Click the service name to display the functions.

Step 3 Click the memory metrics function name.

Step 4 In the **Triggers** tab, click **Enable** in the trigger name row.



Monitor Memory Metrics

Procedure

Step 1 In the Alibaba Cloud CloudMonitor console, click **Custom Monitoring**.

Step 2 Select the memory metrics group name from the **Select Group** drop-down list, the memory metric name, and **Dimension** (time in minutes).

Step 3 Click **OK** to display a graph depicting minute-to-minute Threat Defense Virtual memory usage. Any increase in memory usage is depicted by a spike in the graph.

Disable Auto Scaling

Perform the steps given below to disable auto scaling.

Procedure

Step 1 In the Alibaba Cloud Auto Scaling console, click **Scaling Groups**.

Step 2 Click the **Scaling Group Name/ID** for which you want to disable auto scaling.

Step 3 Click **Disabled** to disable auto scaling.

Post-Deployment Logs

You can see the post-deployment logs to view the scale out, scale in, and memory metrics function parameters. To see the logs, go to **Alibaba Cloud Homepage > Services > Service details > Functions > Function details > Logs > Function Logs**.

Scale-out Function Logs

You can see details such as the scaling group name, access and secret keys, region along with the instance ID and public IP of the scaled out Threat Defense Virtual instance.

Scale-in Function Logs

You can see updates related to deregistration of the Threat Defense Virtual instance.

Memory Metrics Function Logs

You can see details such as the scaling group ID, public IP addresses of the instances in the scaling group, along with the average memory usage of the Threat Defense Virtual instances.

Troubleshoot

- Issue: Unable to SSH to the Threat Defense Virtual instance

  Troubleshooting: Ensure that the password of the Threat Defense Virtual instance is correct in the environment variables.

- Issue: Unable to import the module index or the 'module not found' message is displayed in the Alibaba Cloud Function logs.

  Example:

  ```
  {'errorMessage': "Unable to import module 'index'", 'errorType': 'ImportModuleError',
  'stackTrace': ["ModuleNotFoundError: No module named 'aliyunsdkslb'"]}
  ```

  Troubleshooting: Ensure that the slb layer is attached to your function.

Note The issue and troubleshooting step is similar for the other layers in the function.

- Issue: License Registration Failed

  Troubleshooting:

  - Ensure that the License ID token is correct.
  - Ensure that the Threat Defense Virtual can reach the CSSM.
  - Check the number of available licenses in the Smart Licensing Virtual Account.

- Issue: Health Check Failure

  Troubleshooting: Check the health probe NAT rule.

- Issue: Unable to connect to the Management Center Virtual.

  Troubleshooting:

  - Ensure that the Management Center Virtual is reachable.
  - Ensure that the Management Center Virtual credentials are correct.

- Issue: Failed to register with Management Center Virtual.

  Troubleshooting: Check if the management center virtual has availability to accommodate new Threat Defense Virtual instances.

  Also, check if the name of the access control policy in the POLICY_ID field in the scaleout-function environment variables matches the name of the access policy in the management center. The access policy name in both the POLICY_ID field and the management center should match.