cPacket

cPacket Cloud Suite

Quick Start Guide for AWS cVu-V Packet Broker

Version 24.2.1

Revision History

Document Version	Date	Notes
1	13Aug2024	• The original release of this document.
2	03Dec2024	• Document title/naming modifications.
3	15Apr2025	• Title update

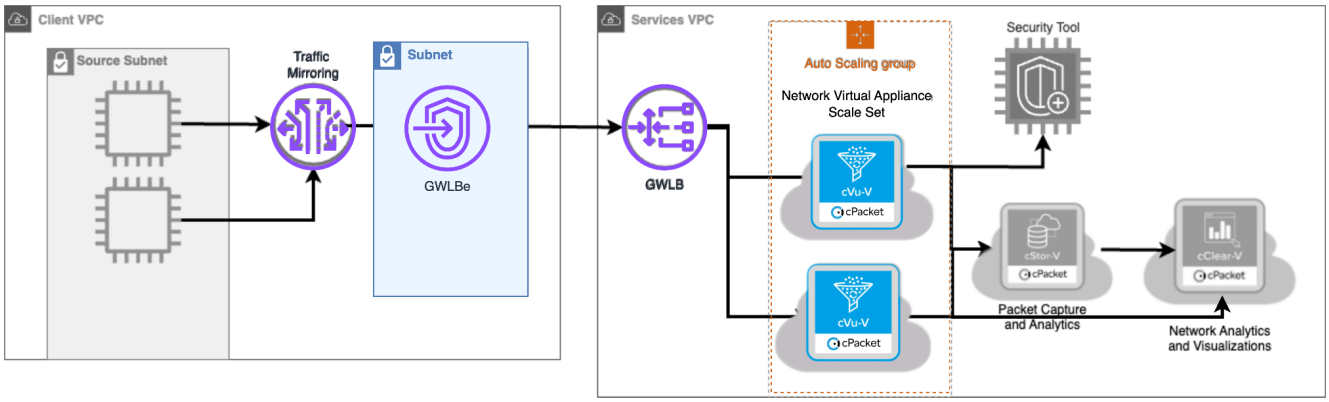
Table of Contents

Introduction

In this guide you will learn how to launch a cPacket cVu-V virtual appliance in your Amazon Web Services (AWS) environment to replicate packets from a Virtual Private Cloud (VPC) with Traffic Mirroring. We recommend using this guide to set up a basic cVu-V deployment in accounts that are primarily used for testing and evaluation. cPacket Solutions Engineering will work with you to set up cPacket solutions at scale with scripting and automation when you are ready to deploy the solutions more broadly in AWS.

Getting started

A mirror session is a connection between a mirror source and a mirror target. In the following diagram, the mirror sources on the left are EC2 instances and the mirror target is the Gateway Load Balancer Endpoint (GWLBe). On the right is a Gateway Load Balancer (GWLB) which will send packets to an auto scaling group of cPacket cVu-V virtual appliances. **Traffic is sent automatically from the Gateway Load Balancer Endpoint to the Gateway Load Balancer**. The mirror filter determines which network packets are mirrored.



Before you begin

Access to the Amazon Machine Images (AMI) for cPacket appliances is provided by sharing images to a specific AWS account ID and Region where you will be installing cPacket virtual appliances. You must provide the AWS account ID and Region where you will be installing to your cPacket representative. cPacket will share the latest AMI images for the virtual appliances to this account ID in selected regions.

We recommend that you install cClear-V and optionally cStor-V before installing cVu-V. See the ***Quick Start Guide for AWS cClear-V Control Center*** for detailed information about installing cClear-V and the ***Quick Start Guide for AWS cStor-V Packet Capture*** for detailed information about installing cStor-V.

The following table lists all the requirements necessary to begin installation in AWS.

Requirement	Detail
AWS User ID	You will need a user ID in an AWS account, permissions for the user are listed below.
AWS Account and Region	If you cannot access the AWS Marketplace, an AWS account and region must be provided to cPacket so the virtual appliance images (Amazon Machine Image - AMI's) can be shared. Account numbers are 12 digits in length and example regions names are: us-east-1, us-west-2, …
cPacket License Key	cPacket will provide you with a license key used to activate the cVu-V appliance.
AWS Organization Tagging Policies	Your organization may have requirements for tagging resources created in the cloud. Common tag requirements are: Name, Owner, and CreatedBy. If these tags are mandated by your organization, creating a device without them will fail the organizational policy. Remediation is to simply add the required tags at resource creation time.
Identity and Access Management (IAM) User/Permissions/Role	In the account used for installation, the user needs to have permissions granted to setup the cVu-V virtual appliance. See IAM Policy to install cVu-V for the minimum permissions. Your organization may already have roles defined that grant these permissions through their IAM policies, if not there are existing AWS managed policies that can be granted to the user or the group the user is a member of. They are: - AmazonEC2FullAccess - AmazonEC2RoleforSSM - AWSMarketplaceManageSubscriptions (for marketplace access)
Network Bandwidth	The network bandwidth to be captured will determine the size of the instance for the cVu-V appliance. Recommendation on instance sizing and the number of storage volumes are contained later in this guide and indexed by network bandwidth of Gbps you would like to capture.
Virtual Private Cloud (VPC)	You will need a **VPC** to install the cVu-V into. We recommend working with your organization's AWS cloud support team and requesting a role that allows creating a VPC.

<table>
<tr><td>Security Groups/Policies</td><td>The following ports will be opened on the cVu-V for inbound and outbound traffic.

Inbound:
<table><tr><td>TCP 22</td><td>SSH traffic</td></tr><tr><td>TCP 443</td><td>Encrypted HTTP traffic</td></tr><tr><td>UDP 6081</td><td>UDP port to receive GENEVE encapsulated traffic</td></tr></table>
Outbound ports:
- All TCP/UDP ports</td></tr>
<tr><td>SSH Key Pair</td><td>A SSH public/private key pair to control access to the virtual appliances and hosts. If you don’t currently have an SSH Key Pair you can create one during installation.</td></tr>
<tr><td>AWS Cloud Shell Access</td><td>The account and user you are using must have permission to use the AWS Console.</td></tr>
</table>

Installation using the cPacket cVu-V Shared AMI

You will be performing the following steps:

- Creating a service VPC and other resources
- Creating a launch template
- Creating an Auto Scaling group
- Verifying your Auto Scaling group
- Creating a Gateway Load Balancer
- Creating a Gateway Load Balancer endpoint service
- Creating a Gateway Load Balancer endpoint

Create a service VPC and other resources (Optional)

If you do not already have a service VPC with cClear-V and/or cStor-V, use the following procedure to create a service VPC plus the additional VPC resources that you need to run cVu-V, such as subnets, route tables, internet gateways, and NAT gateways.

To create a VPC, subnets, and other VPC resources using the console

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. On the VPC dashboard, choose **Create VPC**.

3. For **Resources to create**, choose **VPC and more**.
4. Keep **Name tag auto-generation** selected to create Name tags for the VPC resources or clear it to provide your own **Name** tags for the VPC resources.
5. For **IPv4 CIDR block**, enter an IPv4 address range for the VPC. A VPC must have an IPv4 address range.
6. Select the **Tenancy** option. Choose the tenancy of the VPC to be `Default`, EC2 instances launched into this VPC will use the tenancy attribute specified when you launch the instance.
7. For **Number of Availability Zones (AZs)**, we recommend that you provision subnets in at least two Availability Zones for a production environment. To choose the AZs for your subnets, expand **Customize AZs**. For this deployment, choose 1 availability zone.
8. To configure your subnets, choose values for **Number of public subnets** and **Number of private subnets**. To choose the IP address ranges for your subnets, expand **Customize subnets CIDR blocks**. For this deployment, choose 1 public subnet and 1 private subnet. The public subnet will be the management subnet through which network operators can reach the cPacket appliances. The private subnet will be the capture subnet where replicated packets will reach cVu-V instances.
9. For **NAT gateways ($)**, for this deployment, choose **1 per AZ**.
10. Select **None** for **VPC Endpoints**.
11. Enable DNS hostnames and DNS resolution.
12. (Optional) To add a tag to your VPC, expand **Additional tags**, choose **Add new tag**, and enter a tag key and a tag value.
13. When you are finished configuring your VPC, choose **Create** VPC.

Create a launch template

In this step, you create a launch template that specifies the type of EC2 instance that Amazon EC2 Auto Scaling creates for you. Include information such as the ID of the Amazon Machine Image (AMI) to use, the instance type, the key pair, and security groups.

To create a launch template

1. Open the Amazon EC2 console and go to the Launch templates page.
2. On the top navigation bar, select an AWS Region. The launch template and Auto Scaling group that you create are tied to the Region that you specify.
3. Choose **Create launch template**.
4. For **Launch template name**, enter cvu-auto-scaling.
5. Under **Auto Scaling guidance**, select the check box.
6. (Optional) Add any template tags.

7. For **Application and OS Images (Amazon Machine Image)**, choose the cVu-V image under **My AMIs** and **Shared with me**.
8. For **Instance type**, choose m5n.xlarge.
9. **For Key pair (login)**, choose an existing key pair or create one. You will use this key pair to connect to the Amazon EC2 instances via SSH.
10. For **Network settings**,
    a. Under Subnet choose **Don't include in launch template**.
    b. Under **Security group**, you can select an existing security group used by your organization or create one. The following security rules and protocols are needed for the cVu-V, you can add additional protocols by selecting **Add security group** rule.

Type	Port	Source Type
SSH	22	CIDR block for the VPC, e.g. 10.0.0.0/16
HTTPS	443	CIDR block for the VPC, e.g. 10.0.0.0/16
GENEVE	UDP 6081	CIDR block for the VPC, e.g. 10.0.0.0/16

11. For Storage, leave the defaults.
12. Add any resource tags.
13. Under Advanced Details, copy the following cloud-init user data to the **User data** text box. Ensure the value for the cstorv_ip variable is the IPV4 address of the downstream cStor-V instance or other downstream tool. Remove the section for VxLAN if not using a downstream tool.

```
#!/bin/bash
set -ex

# turn off ip forwarding: not needed when using GWLB with traffic mirroring
sed -i 's/net.ipv4.ip_forward=1/#net.ipv4.ip_forward=1/' /etc/sysctl.conf

cstorv_ip="10.0.0.20"

# Retrieve Instance Metadata
token="$(curl -s -X PUT "http://169.254.169.254/latest/api/token" -H
"X-aws-ec2-metadata-token-ttl-seconds: 21600")"

macs="$(curl -s -H "X-aws-ec2-metadata-token: $token"
http://169.254.169.254/latest/meta-data/network/interfaces/macs/)"
macs="$(printf "%s\n" "$macs" | perl -p -e 's#/$##g')"
```



```
declare -A nics
declare -A nic_ips
while read -r mac; do
  device_number="$(curl -s -H "X-aws-ec2-metadata-token: $token"
"http://169.254.169.254/latest/meta-data/network/interfaces/macs/$mac/device-number"
)"
  # Assume there is only one private IP per NIC
  private_ip="$(curl -s -H "X-aws-ec2-metadata-token: $token"
"http://169.254.169.254/latest/meta-data/network/interfaces/macs/$mac/local-ipv4s")"
  interface_name="$(ip -o link | grep "$mac" | awk '{print $2}' | perl -p -e
's/:$//')"
  nic_ips["$device_number"]="$private_ip"
  nics["$device_number"]="$interface_name"
done <<<"$macs"

if [[ -n "${nic_ips["1"]}" ]]; then
  capture_nic_ip="${nic_ips["1"]}"
  capture_nic="${nics["1"]}"
  capture_nic_index="1"
  echo "capture nic IP: $capture_nic_ip"
else
  capture_nic_ip="${nic_ips["0"]}"
  capture_nic="${nics["0"]}"
  # shellcheck disable=SC2034
  capture_nic_index="0"
  echo "capture nic IP: $capture_nic_ip"
fi

management_nic="${nics["0"]}"
echo "management nic: $management_nic"
echo "management nic ip: ${nic_ips["0"]}"

touch /home/cpacket/boot_config.toml
chmod a+w /home/cpacket/boot_config.toml

cat >/home/cpacket/boot_config.toml <<EOF_BOOTCFG
vm_type = "aws"
cvuv_mode = "endpoint"
cvuv_mirror_eth_0 = "$capture_nic"
cvuv_mgmt_dev = “$capture_nic”

# Remove if not using cStor-V or a downstream tool
cvuv_vxlan_id_0 = 1337
cvuv_vxlan_srcip_0 = "$capture_nic_ip"
cvuv_vxlan_remoteip_0 = "$cstorv_ip"

cvuv_endpoint_vxlan_id_0 = 222
EOF_BOOTCFG
```

14. Choose **Create launch template**.

Create an Auto Scaling group

To create an Auto Scaling group

1. Go to the EC2 dashboard, select Auto Scaling Groups, and select “Create Auto Scaling Group”.
2. On the **Choose launch template,** for **Auto Scaling group name**, enter **cvu-asg** or a name that matches your naming conventions.
3. Under **Launch template**, choose the launch template we created in the previous step.
4. For the version, select “Latest” and then Choose **Next**
5. The **Choose instance launch options** page appears. In the **Instance type requirements** section, use the default setting to simplify this step. (Do not **Override launch template**.) For this deployment, you will launch only one On-Demand Instance using the instance type specified in your launch template.
6. In the **Network** section, select the service VPC.
7. For **Availability Zones and subnets**, choose the private (capture) subnet.
8. Keep the rest of the defaults for this deployment and choose **Skip to review**.

Note	The initial size of the group is determined by its desired capacity. The default value is 1 instance.

9. On the **Review** page, review the information for the group, and then choose **Create Auto Scaling group**.

Verify your Auto Scaling group

Now that you have created an Auto Scaling group, you are ready to verify that the group has launched cVu-V.

To verify that your Auto Scaling group has launched an EC2 instance

1. Open the Auto Scaling groups page of the Amazon EC2 console.
2. Select the check box next to the Auto Scaling group that you just created.
   A split pane opens up in the bottom of the Auto Scaling groups page. The first tab available is the Details tab, showing information about the Auto Scaling group.
3. Choose the second tab, **Activity**. Under Activity history, you can view the progress of activities that are associated with the Auto Scaling group. The Status column shows the

current status of your instance. While your instance is launching, the status column shows Not yet in service. The status changes to Successful after the instance is launched. You can also use the refresh button to see the current status of your instance.

4. On the Instance management tab, under Instances, you can view the status of the instance.
5. Verify that your instance launched successfully. It takes a short time for an instance to launch.
    - The Lifecycle column shows the state of your instance. Initially, your instance is in the Pending state. After an instance is ready to receive traffic, its state is InService.
    - The Health status column shows the result of the Amazon EC2 Auto Scaling health checks on your instance.

Create a Gateway Load Balancer

Use the following procedure to create your load balancer, listener, and target group.

To create the load balancer, listener, and target group using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, under **Load Balancing**, choose **Load Balancers**.
3. Choose **Create load balancer**.
4. Under **Gateway Load Balancer**, choose **Create**.
5. **Basic configuration**
    a. For **Load balancer name**, enter a name for your load balancer.
    b. For **Load balancer IP address type**, choose IPv4.
6. **Network mapping**
    a. For **VPC**, select the service VPC.
    b. For **Mappings**, select the Availability Zone and the private (capture) subnet.
7. **IP listener routing**
    a. For **Default action**, create a target group to receive traffic. This target group must use the GENEVE protocol.
    b. Choose **Create target group**, which opens a new tab in your browser.
    c. Choose the “Instances” target type.
    d. Enter a name for the target group.
    e. Keep the GENEVE protocol.

f. Select the VPC created earlier.

g. Modify the health check settings:

   i. Choose HTTPS

   ii. The health check path should be **/api/info/v1/health**

   iii. Under **Advanced health check settings**, the **Health check port** should be set to 443.

h. add any tags that you need. Choose **Next**.

i. Register the targets from the Auto Scaling group into the target group. Select **EC2 > Target Groups**, then select the target group created above.

   i. Under **Available Instances**, select the cVu-V instance from the Auto Scaling group.

   ii. Click **Include as pending**.

   iii. Click **Register pending targets**.

j. Choose **Create target group**.

8. Refresh the **Default Action** drop down list and select the target group created above.
9. (Optional) Expand **Listener tags** and add the tags that you need.
10. (Optional) Expand **Load balancer tags** and add the tags that you need.
11. Choose **Create load balancer**.

Create a Gateway Load Balancer endpoint service

Use the following procedure to create an endpoint service using your Gateway Load Balancer.

To create a Gateway Load Balancer endpoint service

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Endpoint services**.
3. Choose **Create endpoint service** and do the following:

   a. Fill in the **name tag** for the endpoint service.

   b. For **Load balancer type**, choose Gateway.

   c. For **Available load balancers,** select your Gateway Load Balancer.

   d. For **Additional Settings**,

   - Uncheck **Acceptance required** in order to ingest all traffic sent to the Gateway load balancer by default.
   - select **IPv4** – Enable the endpoint service to accept IPv4 requests.

e. (Optional) To add a tag, choose **Add new tag** and enter the tag key and tag value.

f. Choose **Create**.

> **Important:** Note the **Service name**; you'll need it when you create the endpoint

Create a client VPC (Optional)

You may monitor an existing VPC or create a client VPC. Use the following procedure to create a client network that will contain the traffic to be inspected by the services VPC. Traffic mirroring sessions will send traffic to a Gateway Load Balancer endpoint (GWLBe) (created in the next section) which will be automatically forwarded to the Gateway Load Balancer (GWLB) in the services VPC.

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. On the VPC dashboard, choose **Create VPC**.
3. For **Resources to create**, choose **VPC** and more.
4. Keep **Name tag auto-generation** selected to create Name tags for the VPC resources or clear it to provide your own Name tags for the VPC resources.
5. For **IPv4 CIDR block**, enter an IPv4 address range for the VPC. A VPC must have an IPv4 address range. While not strictly required, choose a CIDR block that does not overlap with the Service VPC.
6. Select the **Tenancy** option. Choose the tenancy of the VPC to be `Default`, EC2 instances launched into this VPC will use the tenancy attribute specified when you launch the instance.
7. For **Number of Availability Zones (AZs)**, choose 1 availability zone.
8. To configure your subnets, choose values for **Number of public subnets** and **Number of private subnets**. To choose the IP address ranges for your subnets, expand **Customize subnets CIDR blocks**. For this VPC, choose 0 public subnets and 2 private subnets. One private subnet will contain the virtual machines that generate traffic to be inspected. The other will contain the Gateway Load Balancer endpoint.
9. Choose **None** for NAT gateways.
10. Select None for VPC Endpoints. (We will create the Gateway Load Balancer in the next section.)
11. Enable DNS hostnames and DNS resolution.

12. (Optional) To add a tag to your VPC, expand **Additional tags**, choose **Add new tag**, and enter a tag key and a tag value.
13. When you are finished configuring your VPC, choose **Create** VPC.
14. Create two virtual machines in the private subnet. Traffic mirroring sessions will replicate the packets from these NICs to the Services VPC.

Create a Gateway Load Balancer endpoint

Use the following procedure to create a Gateway Load Balancer endpoint that connects to your Gateway Load Balancer endpoint service.

To create a Gateway Load Balancer endpoint

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Endpoints**.
3. Choose **Create endpoint** and do the following:
    a. Fill in the **name tag** for the endpoint.
    b. For the Service **category**, choose **Other endpoint services**.
    c. For **Service name**, enter the service name that you noted earlier, and then choose **Verify service**.
    d. For **VPC**, select the VPC you wish to monitor. If you created a client VPC in the previous step select it. (You should not select the Services VPC that contains the Gateway Load Balancer).
    e. For **Subnets**, select a subnet for the Gateway Load Balancer endpoint.
    f. For **IP address type**, choose **IPv4** – Assign IPv4 addresses to your endpoint network interfaces. This option is supported only if all selected subnets have IPv4 address ranges.
    g. Adding any tags required by your organization. To add a tag, choose Add new tag and enter the tag key and tag value.
    h. Choose **Create endpoint**. The initial status is **Pending**.
    i. Record the **Endpoint ID** of the endpoint for the traffic mirroring target. This will be used to create the traffic mirroring target.

After Installation and Launching

Please allow a few minutes for the cVu-V to become accessible. To reach the cVu-V, a jump box connection needs to be established. This connection will be created using the cClear-V. In case your cClear-V does not already have a public IP address, it is necessary to assign one. Associate an Elastic IP with an instance.

Log In and License

1. Create a tunnel using SSH local port forwarding. In your terminal, enter:

```
ssh -N -L 127.0.01:8443:<cvu_private_ip_address>:443 \
ubuntu@<cclear_public_ip_address> -i <privatekey>
```

   - 8443 is the local port that is forwarded to the VM instance via SSH.
   - <cvu_private_ip_address> is the cCur-V instance private IP address.
   - <cclear_public_ip_address> is the cClear-V instance public IP address.
   - <privatekey> is the SSH key pair used as a prerequisite.

2. Set a manual proxy in your web browser for a SOCKS host: **localhost** and Port: **8443**
3. Enter https://127.0.0.1:8443 in your web browser to reach the cVu login page.

NOTE: You may need to add the **8443** into your URL if it is removed.

cVu-V requires you to have a valid license to replicate traffic to downstream tools. You should connect this cVu-V to an existing cClear-V with an active cVu-V license. The cClear-V must have a network path to the cVu-V.

Traffic Mirroring

Once you have completed the installation and activated the license, your cVu-V is ready to receive mirrored traffic from EC2 Instances. The AWS guide for Traffic Mirroring can be found here (AWS Traffic Mirroring). This guide will walk you through the necessary steps to receive mirrored traffic from an EC2 instance.

There are three resources to create for Traffic Mirroring

1. Traffic Mirror Target – This is the Gateway Load Balancer endpoint
2. Traffic Mirror Filter – Rules to determine which traffic to mirror from the source(s)
3. Traffic Mirror Session – Session that identifies the mirrored source and target

Create a traffic mirror target

To set up a Traffic Mirror Target you will need the identifier for the Gateway Load Balancer endpoint.

1. In the AWS Management Console, in the top menu, select **Services**.

2. Select **VPC**.
3. In the left pane, under Traffic Mirroring, select **Mirror Targets**.
4. Select **Create Traffic Mirror Target**.
5. In the **Name** tag field, type a descriptive name for the target.
6. In the **Description** field, type a description for the target.
7. From the **Target** type drop-down list, select **Gateway Load Balancer endpoint**.
8. From the **Target** search box, enter the cVu-V gateway load balancer endpoint ID recorded after its creation earlier.
9. Select **Create**.

> **Note:** The Traffic Mirror Target ID that is created, it will look like tmt-01421338b23ede911 with unique digits for your ID.

Create a traffic mirror filter

You must create a filter to allow or restrict traffic sent from the mirroring sources to cVu-V. For the purposes of this guide we will mirror all inbound and outbound traffic from EC2 instances.

1. In the AWS Management Console, in the left pane under Traffic Mirroring, select **Mirror Filters**.
2. Select **Create traffic mirror filter**.
3. In the **Name tag** field, type a name for the filter.
4. In the **Description** field, type a description for the filter.
5. Under **Network services**, select the **amazon-dns** checkbox.
6. In the **Inbound rules** section, select **Add rule**.
7. Configure an inbound rule:
    1. In the **Number** field, type a number for the rule, such as 100.
    2. From the **Rule action** drop-down list, select **accept**.
    3. From the **Protocol** drop-down list, select **All protocols**.

4. In the **Source CIDR block** field, type 0.0.0.0/0.
5. In the **Destination CIDR block** field, type 0.0.0.0/0.
6. In the **Description** field, type a description for the rule.

8. In the **Outbound rules** section, select **Add rule**.

9. Configure an outbound rule:
    1. In the **Number** field, type a number for the rule, such as 100.
    2. From the **Rule action** drop-down list, select accept.
    3. From the **Protocol** drop-down list, select All protocols.
    4. In the **Source CIDR block** field, type 0.0.0.0/0.
    5. In the **Destination CIDR block** field, type 0.0.0.0/0.
    6. In the **Description** field, type a description for the rule.
10. Select **Create**.

For VPC and subnets that contain application instances that will be mirrored, we recommend the following filtering rules to help avoid mirroring duplicate frames from peer EC2 instances that are in a single VPC to the cVu-V.

- All outbound traffic is mirrored to the cVu-V, whether the traffic is sent from one peer device to another on the subnet or if the traffic is sent to a device outside of the subnet.
- Inbound traffic is only mirrored to the cVu-V when the traffic is from an external device. For example, this rule ensures that an app server request is not mirrored twice: once from the sending app server and once from the database that received the request.
- Rule numbers determine the order in which the filters are applied. Rules with lower numbers, such as 100, are applied first.

Important: These filters should only be applied when mirroring all the instances in a CIDR block including app servers.

Create a traffic mirror session

You must create a session for each AWS resource that you want to monitor.

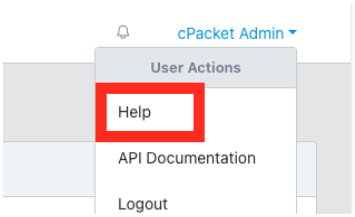
Note: To prevent mirror packets from being truncated, set the traffic mirror source interface MTU value to 54 bytes less than the traffic mirror target MTU value for IPv4 and 74 bytes less than the traffic mirror target MTU value for IPv6. For more information about configuring the network MTU value, see the following AWS documentation: Network Maximum Transmission Unit (MTU) for Your EC2 Instance.

1. In the AWS Management Console, in the left pane, under Traffic Mirroring, select **Mirror Sessions**.
2. Select **Create traffic mirror session**.
3. In the **Name** tag field type a descriptive name for the session.
4. In the **Description** field type a description for the session.
5. From the **Mirror source** drop-down list, select the source **ENI** to mirror, this is typically attached to the EC2 instance that you want to monitor. This EC2 instance should be in a private subnet of the VPC selected as the GWLBe VPC.
6. From the **Mirror target** drop-down list, select the traffic mirror target ID generated for the Gateway Load Balancer endpoint that was created earlier..
7. In the **Session number** field, type 1.
8. For the VNI field, leave this field empty and the system assigns a random unique VNI.
9. For the **Packet length** field, leave this field empty, this will mirror the entire packet.
10. From the **Filter** drop-down list select the ID for the traffic mirror filter you created in the previous step.
11. Select **Create.**

The traffic from the ENI you selected should now be mirrored to the cVu-V. You can verify operation by running a packet capture in the cVu-V and verifying the source address in the packet capture includes the IP address of the ENI being mirrored.

Verifying Operation

1. In cClear add your cVu. For detailed instructions, please refer to the cClear User Guide which can be accessed from the help menu.

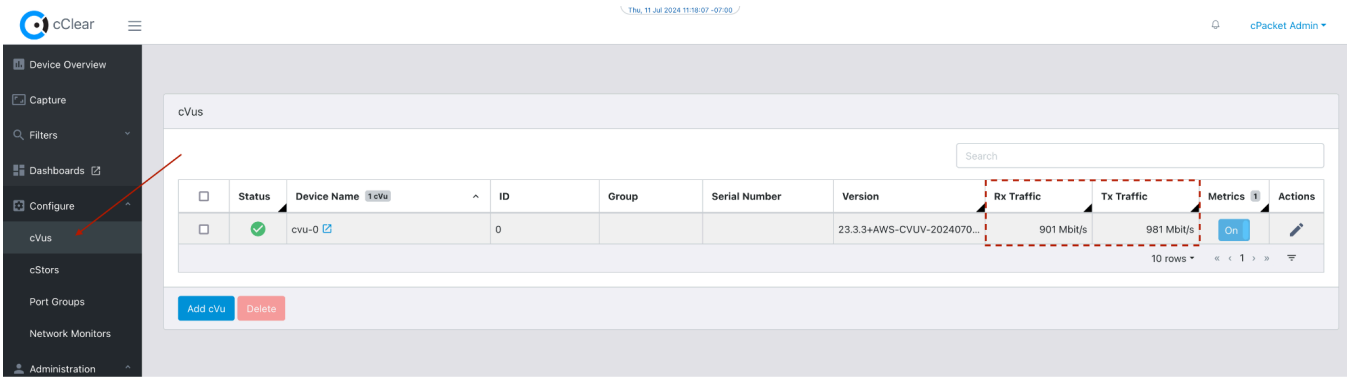


   a. Go to Configure > cVus.
   b. Click Add cVu.

      For Auth Type, select User Login.

      i. Enter the cVu name, IP address, and login credentials.
      ii. Click Save to add the cVu.

2. Verify that cClear-V is indicating that traffic is flowing to your cVu-V packet capture appliance.



IAM Policy to Install cVu-V

The following policy is used to install and operate the cVu-V appliance. It is a very restricted policy and defines the minimum permissions necessary. To install cVu-V from the AWS marketplace, the user or the role used will need to attach the AWSMarketplaceManageSubscriptions policy.

```
{
   "Version": "2012-10-17",
   "Statement": [
       {
```



```
            "Sid": "cStorMinInstall",
            "Effect": "Allow",
            "Action": [
                "iam:ListInstanceProfiles",
                "ec2:*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "cStorMinSSHKey",
            "Effect": "Allow",
            "Action": [
                "iam:GetSSHPublicKey",
                "iam:ListSSHPublicKeys",
                "iam:UpdateSSHPublicKey",
                "iam:UploadSSHPublicKey"
            ],
            "Resource": [
                "arn:aws:iam::*:user/${aws:username}"
            ]
        },
        {
            "Sid": "cStorMinCloudWatch",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricData",
                "cloudwatch:DescribeAlarms"
            ],
            "Resource": "*"
        }
    ]
}
```