Secure Kubernetes Services in an EKS Cluster

BETA

techDOCS paloalto NETWORKS

Contact Information

Corporate Headquarters:
Palo Alto Networks
3000 Tannery Way
Santa Clara, CA 95054
www.paloaltonetworks.com/company/contact-support

About the Documentation

- To ensure you are viewing the most current version of this document, or to access related documentation, visit the Technical Documentation portal: docs.paloaltonetworks.com.
- To search for a specific topic, go to our search page: docs.paloaltonetworks.com/search.html.
- Have feedback or questions for us? Leave a comment on any page in the portal, or write to us at documentation@paloaltonetworks.com.

Copyright

Last Revised

July 26, 2019

Table of Contents

Secure Kubernetes Services in an EKS Cluster

The Panorama plugin for EKS enables you to secure North/South traffic destined to and/or originating from container services and workloads in Amazon Elastic Kubernetes Service (EKS) environments in which you have deployed VM-Series firewalls.

After you configure the EKS plugin on Panorama to communicate with an EKS cluster, the plugin uses the Kubernetes SDK to retrieve information from each service that has an exposed IP or FQDN. With this information you can create security policy in Panorama. To secure inbound and outbound traffic to the cluster, push your configuration to managed VM-Series firewalls.

How Does the Panorama Plugin for Amazon EKS Secure Kubernetes Services?

You can use VM-Series firewalls to secure inbound traffic for Amazon Elastic Kubernetes Service (EKS) clusters. The Panorama plugin for Amazon EKS secures inbound traffic to Kubernetes clusters, and provides outbound monitoring for traffic exiting the cluster. Outbound traffic can return through the VM-Series firewall, but firewall rules applied to outbound traffic must have a default allow all policy to permit Kubernetes orchestration traffic to function.

- The minimum Panorama software version is 9.0.3.
- You must deploy your VM-Series firewall (or firewall set) in the same VPC as your EKS cluster. You can create up to 16 clusters in the same VPC and secure them with the same firewall or firewall set.

This chapter reviews different components that enable the EKS Plugin for Panorama to secure an EKS cluster.

- Requirements
- System Architecture
- EKS Plugin on Panorama

Requirements

This solution requires the following:

- Panorama—A licensed version of Panorama, version 9.0.3 and later.
  - Your Panorama version must be the same version or a later version than your VM-Series firewall PAN-OS version.
  - EKS Plugin on Panorama—version 1.0.0. See EKS Plugin on Panorama.
- VM-Series firewalls—PAN-OS version 8.1 and later, or version 9.0.3 and later.

  For each firewall you need a BYOL license and you must know the auth code so that you can use it to bootstrap the firewall.
- EKS template bundle, version 1.0, available. See Templates.
- AWS account—In addition to your user name and password you must know your AWS Access Key, which is comprised of the access key ID and the secret access key. If you have an account but do not know your secret access key, you can create an access key and save the .csv file in a secure place. The required policies and roles below will permit it.

  Be sure to follow the IAM best practices.
- AWS policies and roles—Your AWS account must be able to access to the following service policies:

  To manage the firewalls:

  - AmazonEC2ContainerServiceforEC2Role
  - AmazonEKS_CNI_Policy
  - AmazonEKSClusterPolicy
  - AmazonEKSServicePolicy
  - AmazonEKSWorkerNodePolicy

  To manage EKS:

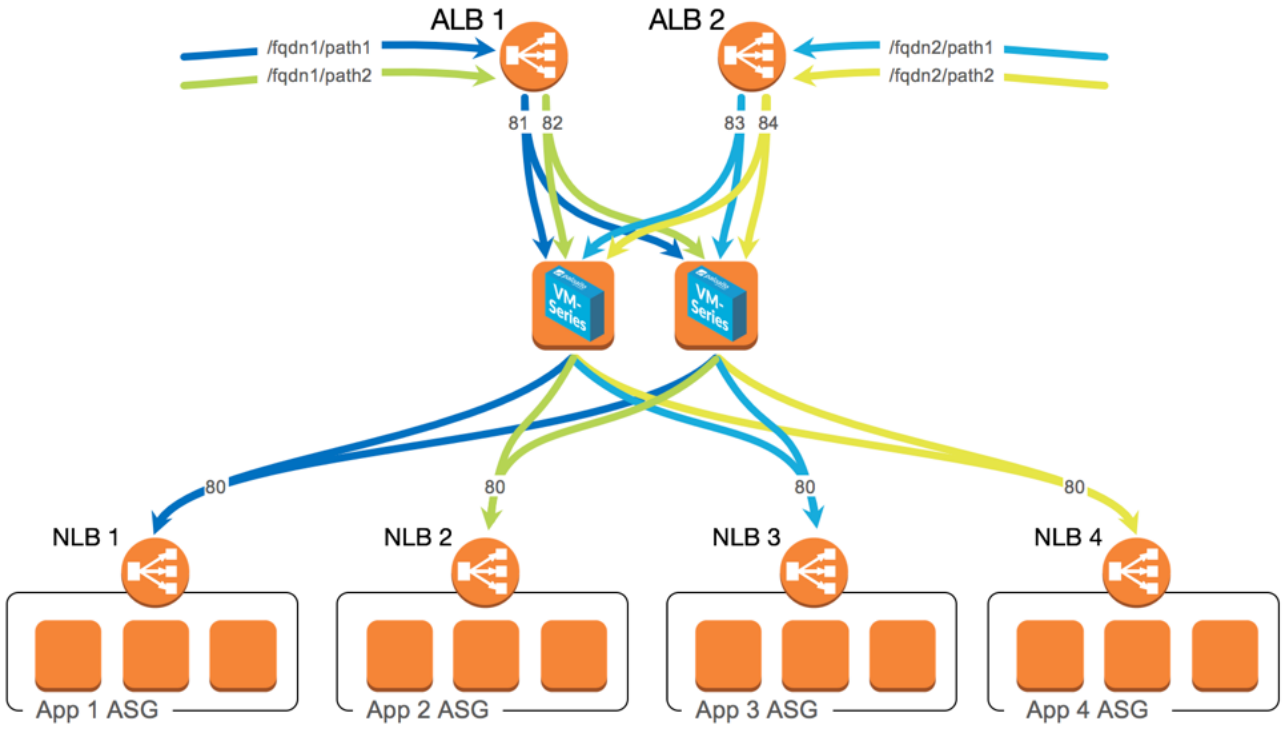
  - AmazonEC2FullAccess
  - AmazonLambdaFullAccess
  - AmazonS3FullAccess

- AmazonSQSFullAccess
- AmazonVPCFullAccess
- AWSCloudFormationFullAccess
- AWSMarketplaceFullAccess
- ElasticLoadBalancingFullAccess
- IAMFullAccess

- AWS CLI—Install or update the AWS CLI; note the dependency on a supported version of Python.
- Kubernetes and kubectl—View the available Amazon EKS versions and install kubectl for your local OS. The version you install must be within one minor version of the EKS version (you will choose the Kubernetes version when you create the cluster).

System Architecture

The following diagram illustrates a sample deployment that secures inbound traffic for Amazon EKS clusters —a load balancer sandwich.



The application load balancers (ALBs) face the internet. For EKS, the application ASGs in this diagram are Kubernetes pods in a service behind an network load balancer (NLB). The VM-Series firewall set sandwiched between the ALBs and the NLBs to provides inbound security to the cluster.

Inbound Security

To secure traffic without interrupting communication flows, the VM-Series firewall set is programmed with static routes that properly route traffic to the desired destination, and NAT rules to perform source and destination NATs on the inbound packets, ensuring that the initial traffic as well as the return traffic passes through the firewall set.

To register a service with the firewall, you must label your services with `panw-tg-port` and a port value. This label is applied when the service launches. You must also configure a target group with the destination

of the firewall set and a destination port matching the label on the service. When the traffic hits the firewall, the port that receives it tells the firewall which NAT rule to apply.

The applied NAT rule is a source and destination NAT. The source changes from the ALB to the firewall trust interface, ensuring that return traffic hits the firewall for inspection. The destination then changes from the firewall untrust interface to the service internal load balancer (ILB).

Outbound Security

To route the traffic from the trust to untrust interface, the template ensures the virtual router on the firewall has a default route pointed to untrust. Static routes are programmed for each cluster subnet so that traffic returning to the firewall is routed properly to its destination.

EKS Plugin on Panorama

The EKS Plugin on Panorama configures the firewall set according to the services deployed in a cluster. It creates inbound NAT rules for services, outbound NAT rules (one for each cluster subnet), and static routes for each cluster subnet.

The plugin uses the Kubernetes Python SDK to retrieve information related to services deployed in your cluster. The plugin queries for services that are labeled `panw-tg-port` and have been assigned a valid port value. The plugin uses the port to create an inbound NAT rule that is programmed on the VM-Series firewall. When traffic hits the firewall on that specified port, Panorama applies the inbound NAT rule for that port and routes the packet to its destination. For each service the plugin creates:

- An address object created with the FQDN of the service ILB.
- A service object created for the port specified in the label.
- An inbound NAT rule which creates source and destination NAT using the address object and service object just created.

The plugin is also responsible for adding configuration when a new cluster is added. The plugin uses the API to retrieve cluster information, such as subnets, and VM-Series firewall information, such as the instance ID. The plugin uses the information to create one router per firewall per cluster subnet. For example, if there are two firewalls and three cluster subnets, the plugin creates six static routes.

Additionally, for every cluster subnet, the plugin creates an outbound NAT rule. The NAT rule is applied to any traffic originating from these subnets and it does a source NAT to change the source from the Node IP to the firewall untrust interface.

In Panorama, the plugin provides visibility into discovered services, and services that are currently protected.

Templates

You can download the templates from https://github.com/PaloAltoNetworks/aws/eks. The template files are as follows:

- `firewall-vpc-v1.0.template`

  Creates a new VPC and deploys a VM-Series firewall or firewall set (greenfield deployment).
- `firewall-existing-vpc-v1.0.template`

  Deploys a VM-Series firewall in an existing VPC (brownfield deployment).
- `eks-cluster-v1.0.template`
  - Creates EKS Cluster.
  - Creates control plane security group.
  - Creates private cluster subnets.

    - Creates route table associated with cluster subnets. The default route points to the internet gateway (IGW).
- `eks-node-v1.0.template`
    - Adds node autoscaling group.
    - Adds node security group.
    - Adds entry to cluster control plane security group.

Securing an EKS Cluster with VM-Series Firewall and EKS Plugin on Panorama

To enable Panorama to connect to the load balancers in an AWS Kubernetes Services (EKS) cluster, you must enable the EKS plugin on Panorama to establish a connection with your EKS cluster. Then, you must configure the device groups and templates to which your VM-Series firewalls belong so that Panorama can push configuration objects and policy rules to your managed firewalls.

Set Up Your Panorama Configuration

Configure these Panorama elements before you use the templates to deploy firewalls.

STEP 1 | Add a template.

In Panorama, go to **Panorama** > **Templates** and **Add** a template.

STEP 2 | Add a stack.

Select **Panorama** > **Templates** and **Add Stack**. In the Templates pane, **Add** the template you created in Step 1.

STEP 3 | Add a device group.

Select **Panorama** > **Device Groups** and **Add** a device group. You don't need to enter anything yet.

STEP 4 | Configure the DNS server to point to the AWS DNS server.

1. On the **Device** tab, from the Template menu, select the template stack you created in Step 2.
2. Select **Device** > **Services** and click the Edit gear.
3. Under **Services** select **Servers** and add the IP address of the primary DNS server: 169.254.169.253
4. Click **OK**.

STEP 5 | Configure untrust and trust interfaces, virtual routers, and zones to push to your managed firewalls.

1. Select **Network** > **Interfaces**, and from the **Template** menu, select the template you created in Step 1.
2. Select **Ethernet** > **Add Interface** to configure the untrust interface.
    1. **Slot**: Select Slot 1.
    2. **Interface Name**: Select ethernet1/1.
    3. **Interface Type**: Select Layer3.

4. To create the virtual router, select **Config** and under **Assign Interface To** > **Virtual Router** choose **New Virtual Router**. Under Router Settings, name the router **`vr-default`**. The plugin searches for this specific router name.

   Select **ECMP** and select **Enable**, then click **OK** to return to the **Config** tab.
5. Go to **Assign Interface** > **Security Zone**, choose **New Zone**, name the zone untrust, and click **OK**.
6. Select **IPV4** > **DHCP Client**. Leave **Enable** and **Automatically create default route pointing to default gateway provided by server** checked. This sets the default route to point to the untrust interface.
7. Click **OK**.

3. Configure the trust interface.
   1. Select **Interfaces** > **Ethernet** > **Add Interface**
   2. **Slot**: Select Slot 1.
   3. **Interface Name**: Select ethernet1/2.
   4. **Interface Type**: Select Layer3.
   5. Select **Config** and under **Assign Interface** > **Virtual Router** choose the router you just created: **`vr-default`**.
   6. Select **Security Zone** > **New Zone**, name the zone trust, and click **OK**.
   7. Click **IPV4**, choose **DHCP Client**, and disable (uncheck) **Automatically create default route pointing to default gateway provided by server**.
   8. Click **OK**.
4. (Optional) Configure an allow all policy so you can test connectivity.
   1. Select **Policies** and from the **Device Group** menu, select the device group you made in step 3.
   2. Select **Security** > **Pre Rules** and **Add** a security policy rule.

   - **General**: Name the policy allow-all.
   - **Source**: Select **Any**.
   - **Destination**: Select **Any**.
   - **Service/URL Category**: Select **Any**.
   - Click **OK**.

STEP 6 | **Commit** your changes.

Set Up Your AWS Bootstrap Bucket

STEP 1 | Create an Amazon S3 bucket and bootstrap package as described in Bootstrap the VM-Series Firewall on AWS.

STEP 2 | Download eks.zip ??? from https://github.com/PaloAltoNetworks/aws/eks. In a local directory, extract the contents:

```
\cfg   init-config.txt\templates   panw-aws.zip
```

STEP 3 | Upload panw-aws.zip to your S3 bucket.

This file contains the AWS Lambda code for the templates.

STEP 4 | Edit the init-config.txt file to supply the values for vm-auth-key, panorama-server, tplname, and dgname. This sample configuration uses only one Panorama server, so Panorama-server-2 remains undefined.

- vm-auth-key
  - If you have an auth-key, log on to your Panorama CLI and type:

```
request bootstrap vm-auth-key show
```

- If you don't have an auth-key, to generate one from the CLI, type:

```
request bootstrap vm-auth-key generate lifetime <1-8768>
```

- panorama-server—The IP address of Panorama server.
- tplname—The name of the template stack you created.
- dgname—The name of the device group you created.

Save the file.

STEP 5 | In your Amazon S3 bucket, add files to your bootstrap package as follows:

1. Upload the edited init-config.txt file to \config.
2. Upload `authcodes` to \license.

   `authcodes` (no extension) is a text file you create that contains the VM auth code you received when you purchased your license.The authcodes file ensures bootstrapped firewalls are licensed.

Deploy the Firewall Template on AWS

This task uses the firewall-new-vpc-v1.0.template to configure a firewall stack.

STEP 1 | In AWS go to **AWS Services** > **Management & Governance** > **Cloud Formation** > **Stacks** > **Create stack**.

If you completed the steps in Set Up Your AWS Bootstrap Bucket, your template is ready.

STEP 2 | In **Specify a template**, select **Upload a template file** and upload **`firewall-new-vpc-v1.0.template`** from your local drive.

STEP 3 | Click **Next**.

STEP 4 | Name the stack.

STEP 5 | VPC configuration.

1. VPCName: Your choice.
2. Number of AZs: The number of available zones (AZs) in the region you chose for your S3 bucket (enter 2 or 3).
3. Select AZs: From the list, copy the available AZs for your region:
4. ELBType: Choose either application or network.

STEP 6 | VM-Series firewall instance configuration.

1. AMIID of PANFW image:

   Go to the AMI list, copy the 9.0.1 AMI for the BYOL license, and paste it here.
2. Key pair: Select an Amazon EC2 key pair.
3. SSH From: Enter your public IP address. This address is added to the security group to allow SSH access. To find it, type what's my IP in a browser. If you are specifying a new VPC you must enter a valid CIDR range. For example, x.x.x.x/x.
4. NumberofFWs: 2

STEP 7 | S3 Bucket details. Supply the name of your bucket from Set Up Your AWS Bootstrap Bucket, which contains both firewall and Lambda code.

1. Bootstrap bucket for VM-Series firewalls: Your bucket name.
2. S3 Bucket Name for Lambda Code: Your bucket name.

STEP 8 | ELBName: public-exlb. This name comes from the template.

STEP 9 | Click **Next**.

STEP 10 | Click **Next**. Skip configuring stack options.

STEP 11 | Click **Next**.

On the review page, scroll down and check **I acknowledge that AWS CloudFormation might create IAM resources** and click **Create stack**.

Creation can take up to ten minutes.

STEP 12 | In CloudFormation > Stacks, confirm that the stack is active and the status is CREATE_COMPLETE.

STEP 13 | In Panorama, confirm the firewalls are up and connected to Panorama.

1. Select Panorama > Device Groups, and choose the device group you created. In the Devices/Virtual System column, verify that you have two IP addresses.
2. Select Panorama > Templates, select the template stack you created earlier and you also see the two IP addresses.

Deploy the Cluster Stack

This task uses `eks-cluster-v1.0.template` to set up the cluster subnets and the control plane.

STEP 1 | In AWS go to **AWS Services** > **Management & Governance** > **Cloud Formation** > **Stacks** > **Create stack**.

If you completed the steps in Set Up Your AWS Bootstrap Bucket, your template is ready.

STEP 2 | In **Specify a template**, select **Upload a template file** and upload **`eks-cluster-v1.0.template`** from your local drive.

STEP 3 | Click **Next**.

STEP 4 | Name the cluster.

STEP 5 | Cluster configuration.

1. VPCID: Select the VPC you just deployed.
2. Number of Cluster Subnets: (the same as the number of AZs you specified for the stack).
3. AZs for cluster subnets: Choose two.
4. Private Subnet IP Blocks: Enter a CIDR for each cluster subnet. For example, 192.168.110.0/24, 192.168.111.0/24
5. Internet Gateway ID of VPC: Enter the internet ID for the stack you just created.

   To find the ID, go to **Services** > **Your VPCs** > **Internet Gateways**, and copy the ID (igw-*) corresponding to the stack you just created.

STEP 6 | Click **Next**.

STEP 7 | Click **Next**.

On the review page, scroll down and check **I acknowledge that AWS CloudFormation might create IAM resources** and click **Create**. Creation can take up to minutes.

STEP 8 | In **CloudFormation** > **Stacks** confirm that the stack is active and the status is CREATE_COMPLETE.

STEP 9 | In the cluster you just deployed, note the API server endpoint and your subnets.

Set Up Kubectl and Configure Your Cluster

Set up Kubectl config file so you can use Kubectl commands locally to configure your cluster (when you do not have the AWS CLI installed).

STEP 1 | Set up your Kubectl configuration.

1. Go to Create a kubeconfig for Amazon EKS and follow the directions in "To create your kubeconfig file manually."
   - Copy the sample .config file from "To use the AWS IAM Authenticator for Kubernetes."
   - On the command line, open a text file.

   ```
   vi  ~/.kube/config~<YourClusterName>
   ```
2. Paste in the sample configuration.
3. Edit the sample config file.
   - server—View your EKS Cluster and copy the API server endpoint (https://...) and paste it into your config file.
   - certificate-authority-data—View your EKS Cluster and copy the certificate authority and paste it into your config file.
   - args—Replace the cluster name variable with your cluster name.
   - Save.
4. Set an environment variable for AWS authentication.

   ```
   export AWS_ACCESS_KEY_ID=<your-access-key>
   export AWS_SECRET_ACCESS_KEY_ID= <your-secret-access-key>
   ```
5. Apply the configuration.

   ```
   export KUBECONFIG=$KUBECONFIG:~/<yourConfigFile>
   ```
6. Print the current service.

   ```
   kubectl get svc
   ```

STEP 2 | Create credentials.

- Create a service account for pan-user, and assign permissions.

  ```
  kubectl create serviceaccount pan-user
  ```
- Create the cluster role cluster-read-all.

  ```
  vi cluster_role.yaml
  ```

```
kubectl create -f cluster_role.yaml
```

- Associate the service account with pan-user.

```
vi cluster_role_binding.yaml
create -f cluster_role_binding.yaml
```

STEP 3 | Export service account credentials. The service account name <SA_NAME> is pan-user.

1. Get your service accounts:

```
MY_TOKEN=`kubectl get serviceaccounts <SA_NAME> -o
 jsonpath='{.secrets[0].name}''
```

   The SA_NAME is typically pan-user.
2. Get your secret token:

```
kubectl get secret $MY_TOKEN -o json > <file_name.json>
```

   Replace <file_name.json> the name of your credential file.

Configure AWS Roles and Accounts in Panorama

Add your configuration to the Panorama plugin for AWS. To perform this task you must know the access key ID and the secret access key.

STEP 1 | Select **Panorama** > **AWS** > **Setup** > **IAM Role**.

1. Create eks-role—Provide a role to make API calls to AWS (autoscaling groups and other resources).

   Supply values for Name, Access Key ID, Secret Access Key, and Confirm Secret Access Key.
2. Create assume-role—Supply values for Name, Access Key ID, Secret Access Key, and Confirm Secret Access Key.

STEP 2 | Select **Panorama** > **AWS** > **Setup** > **EKS Service Account**.

1.

STEP 3 | Verify plugin actions.

When you add a new cluster, the plugin creates a NAT rule for every cluster subnet that you created, and configures a static route for each firewall to tell it how to how to access each subnet and the cluster.

In this case there are two outbound NAT rules under in the device group.

Select **Policies** > **Device Group** > **<your Device Group>** > **NAT** and view two outbound NAT rules static route for each firewall. Given one firewall and two subnets, there are 4 static routes.

Configure Inbound Protection and Outbound Monitoring

Configure the ALB

Send traffic to ALB then forward it to firewalls and services deployed in the cluster.

STEP 1 | Create a target group for every service that you are securing with managed firewalls. Every service for which you create a NAT rule for must have its own target group.

1. Create a target group.

   Select **EC2** > **Load Balancing** > **Target Groups** > **Create target group**.

   Fill out the form as follows:

   - Target group name: Enter a name.
   - Target type: Instance
   - Protocol: HTTP
   - Port: Enter the port on the firewall that will receive traffic when this Target Group is applied.
   - VPC: Select the VPC you just created.
2. Click **Create**.

STEP 2 | Edit the firewall auto scaling group.

Select **EC2** > **Auto Scaling Group**.

- Select the auto scaling group you deployed previously.
- Under Target Groups, choose the target group you created in the previous step.
- Click **Save**. Wait a minute before continuing.

STEP 3 | Verify the targets are registered.

- Return to **Load Balancing** > **Target Groups**.
- Select your service, and on the **Targets** tab below, verify the targets are registered.

STEP 4 | Verify load balancing.

- Go to **EC2** > **Load Balancing** > **Load Balancers**.
- Choose your load balancer (check your Cloud Formation template for the name you supplied).
- (Optional) Click **+** to add a rule and click **Insert Rule**. Add a condition and an action (forward to).
- Click the pencil to edit the default rule (if no rules match, forward to). Requests otherwise not routed Forward to frontend-demo-service. click UPDATE. If traffic hitting the ALB on port 80 does not meet any rules, it forwards traffic to front end-demo-service which should forward traffic to firewall on port 82. From there, it should go to the service
- View the Load Balancer description to get the DNS name for the ALB.

  Issue a curl command to ping the DNS name.

```
curl http://rhVPC2-1219937001.us-west-2.elb-amazonws.com
```

  You receive a response from the guestbook demo application, meaning the traffic entered successfully.

STEP 5 | Log in to the firewall CLI to confirm traffic is directed to the correct port.

```
show sessionall
```

View web-browsing traffic originating rom the untrust network and directed to port 80 on the firewall.

Test the Outbound Workflow

STEP 1 | Log in to the outbound firewall, and from the CLI, show session all.

You should see SSL traffic originating from the cluster subnets. View the node IP, and notice that it sends outbound traffic to communicate with the master node.

STEP 2 | Deploy a pod that you can log into.

1. Deploy a pod.

```
kubectl create -f shell-demo.yaml
```

2. Log in to the demo.

```
kubectl exec -it shell-demo – /bin/bash
```

You are logged in.

STEP 3 | Use apt-get to test the session.

```
apt-get update
show session all
```

On the bash shell you can see the apt-get update goes to the firewall and apt-get requests are registered.

STEP 4 | You can also curl something from the internet to demonstrate traffic is going in and out.

```
apt-get install curl
curl http://www.google.com
show session all
```

You see a request to google-base originating from your node IP.

Known Issues

Review the following known issues with using the Panorama plugin for Amazon EKS to secure EKS clusters:

Issue ID	Description
PLUG-2246	When the following command is issued from the Panorama CLI, the output does not show plugin_aws.log: `tail follow yes mp-log plugin_api_server.log plugin_azure.log` To avoid this problem, ensure that your Panorama version is 9.0.3 or later.
PLUG-2253	Delete node stack fails due to dependency on network interfaces. You must delete the stack elements manually.

Getting Help

The following topics provide information on where to find more about this release and how to request support:

- Related Documentation
- Requesting Support

Related Documentation

Refer to the following documentation on https://docs.paloaltonetworks.com/ or search the documentation for more information on our products:

- **Panorama Administrator's Guide**—Provides the basic framework to quickly set up the Panorama™ virtual appliance or an M-Series appliance running version 8.1 or 9.0 for centralized administration of the Palo Alto Networks firewalls.
- **PAN-OS Administrator's Guide**—Provides the concepts and solutions to get the most out of your Palo Alto Networks next-generation firewalls. This includes taking you through the initial configuration and basic set up on your Palo Alto Networks firewalls for PAN-OS 8.1 or 9.0.

Requesting Support

For contacting support, for information on support programs, to manage your account or devices, or to open a support case, go to https://support.paloaltonetworks.com.

To provide feedback on the documentation, please write to us at: documentation@paloaltonetworks.com.

Contact Information

Corporate Headquarters:

Palo Alto Networks

3000 Tannery Way

Santa Clara, CA 95054

https://www.paloaltonetworks.com/company/contact-support

Palo Alto Networks, Inc.

www.paloaltonetworks.com