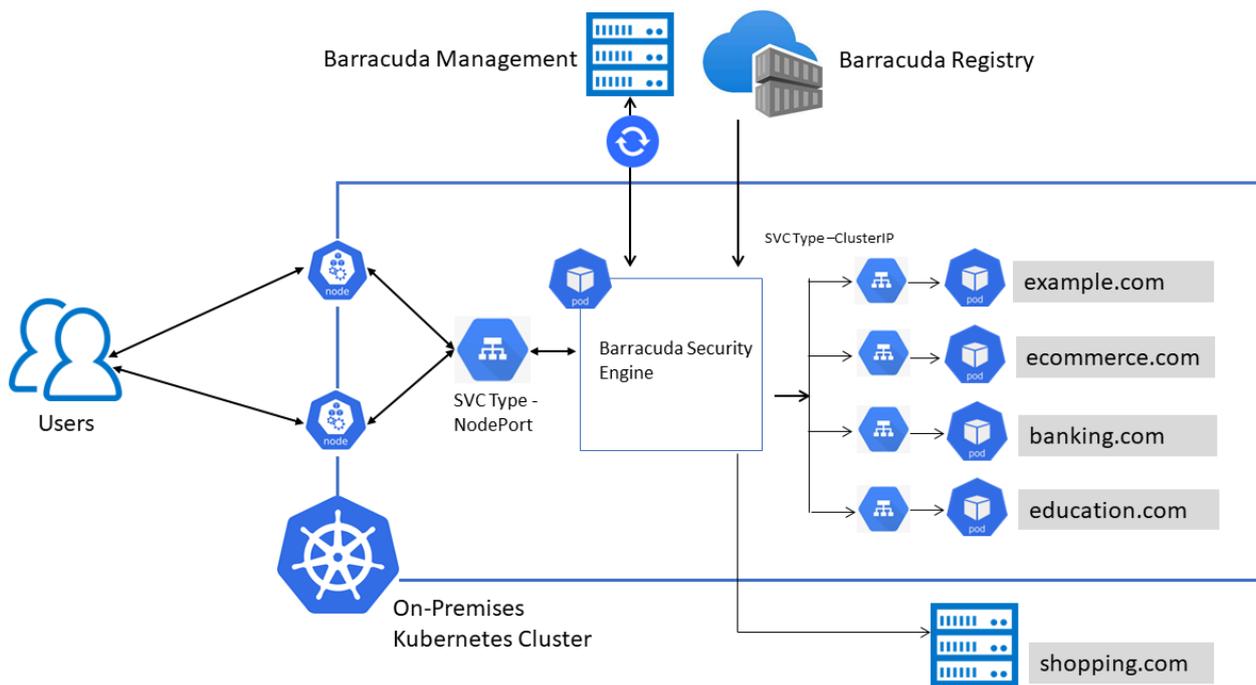


Deploying WAF-as-a-Service Security Module as a Container on On-Premises Kubernetes Cluster

<https://campus.barracuda.com/doc/99123909/>

The traffic processing engine of WAF-as-a-Service can be deployed as a container on a Kubernetes cluster that is hosted and managed by the customer.



Prerequisites

- You must have a WAF-as-a-Service account enabled for custom container deployment. This requires a signed NDA with Barracuda Networks. Contact your Barracuda sales representative for more information on signing an NDA.
- You must have a running [kubernetes cluster](#).
- You must have permission to create the required resources in your Kubernetes cluster.
- Kubernetes command line tool [kubect](#) should be installed on the workstation that is used to manage your kubernetes cluster.
- Allow access to the following domains from the Kubernetes cluster:

| Hostname | Port | TCP/UDP | Direction | Purpose |
|--|------|---------|-----------|-------------------------------|
| container-api.waas.barracudanetworks.com | 443 | TCP | Outbound | Update configuration settings |

| | | | | |
|--|------|------|------------------|--|
| waascontainerprod.blob.core.windows.net | 443 | TCP | Outbound | Storing troubleshooting information |
| wafaas-prod-eh.servicebus.windows.net | 443 | TCP | Outbound | Storing access and firewall logs |
| waas-iot-hub-proxy-func-prod.azurewebsites.net | 8883 | AMQP | Inbound/Outbound | Exchange of configuration and other statistics |

Configuring WAF-as-a-Service

Step 1. Create a Container Key

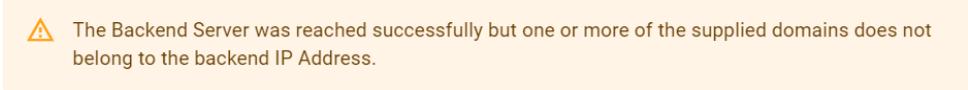
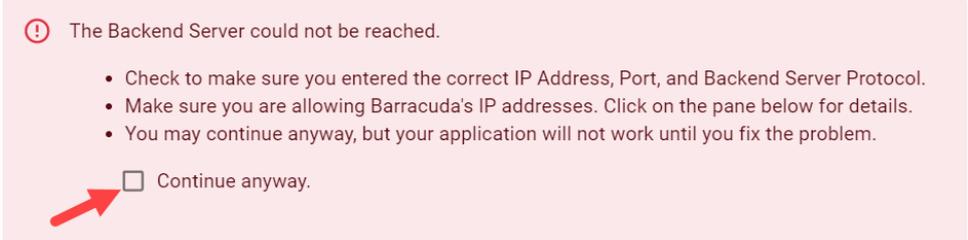
1. Navigate to <https://waas.barracudanetworks.com/> and log in with your Barracuda account credentials.
2. If you do not already have a Barracuda account, click Free 30-Day Trial to sign up for a trial of WAF-as-a-Service.
3. On the WAF-as-a-Service web interface, go to the **RESOURCES** tab, click **WAF CONTAINERS** in the left panel and click **Container Keys**.
4. On the **Container Keys** page, click **New Key**.
5. On the **Create new key** window:
 1. **Key Name** - Enter a name for the key.
 2. Select an option to create the key.
 1. If you select **I will generate my own key and provide the public portion**:
 1. Copy the UNIX command from the window and paste it into your UNIX-like system: `ssh-keygen -f barracuda-wafaas-container-key`
 2. Copy the contents of the `barracuda-wafaas-container-key.pub` file and paste them into the Public key box.
 3. Click **Create**.
 2. If you select **I would like WAF-as-a-Service to generate a key for me**:
 1. The Barracuda WAF-as-a-Service generates a key for the container.
 2. Click **Download** and download the key file.
 3. Click **Create**.

Step 2. Create a Container

1. On the WAF-as-a-Service web interface, go to the **RESOURCES** tab, click **WAF CONTAINERS** on the left panel, and click **Container Management**.
2. On the **Container Management** page, click **Add Container**.
3. On the **Add Container** window:

1. **Name** - Enter a name for the container.
2. **Encryption Key** - Select the key that you created in [Step 1. Create a Container Key](#).
3. Google reCAPTCHA is available for the applications in your container. An advanced risk analysis engine and adaptive CAPTCHAs are employed to challenge suspicious clients and protect against spam, BOTS, and other threats. Clients failing the challenge will not be able to further use your application. To enable this protection you must provide your own reCAPTCHA keys. Refer to the [Google documentation for creating reCAPTCHA keys](#).
4. If you leave these fields blank, or if reCAPTCHA is enabled, but the connection with Google is lost, WAF-as-a-Service's basic CAPTCHA will still challenge clients marked as suspicious.
5. Click **Add**.

Step 3. Add an Application

1. On the WAF-as-a-Service web interface, go to the **APPLICATIONS** tab and click **Add Application**.
2. On the **Add Application** window:
 1. **Websites**
 1. **Application Name** - Enter a name for the application.
 2. **Domain Name** - Enter the domain name of the application.
 3. Click **Continue**.
 2. **Backend Server reachable from public network (internet)**
 1. **Backend Server Protocol** - Select the protocol that needs to be used to access the server.
 2. **IP Address/Hostname** - Verify the IP address/hostname of the backend server.
 3. **Port** - Verify the port number on which the server is listening to.
 4. Click **Test Connection**.
 1. If the backend server is reachable, the following message is displayed:
 1. Click **Add**.
 2. If the backend server is **not reachable** from the public network, the following message is displayed:
 1. Click **Continue** anyway and then click **Add**.
5. Click **Close**.

Step 4. Update the SSL Certificate

1. On the WAF-as-a-Service web interface, go to the **APPLICATIONS** tab and select the application that you created in [Step 3. Add an Application.](#)
2. Click **Endpoints** in the left panel.
3. On the **Endpoint** page:
 1. Click on the three dots next to the endpoint with port 443 and select **Edit Endpoint.**

| DOMAIN | CNAME | PORT | STATUS | MORE |
|----------------------|-------|---|---|--|
| example.com (0 more) | | 80 |  DNS Update Pending |  |
| example.com (0 more) | | 443  |  Error due to Domain's AAAA record |  Edit Endpoint  Delete Endpoint |

4. On the **Edit Endpoint** page:
 1. Scroll down and disable **Automatic Certificate.**
 2. Paste the certificate private key in the **Private Key** text box.
 3. Paste the certificate in the **Certificate** text box.

Edit Endpoint ×

Exposed port on WAF container

Automatic Certificate

Private Key Paste from file

Paste the private key including the BEGIN and END lines shown below

```
-----BEGIN PRIVATE KEY-----  
MIID5TCCAs2gAwIBAgIJALCgZq0XI3xiMA0GCSqGSIb3DQEBBQUAMIGIMQswCQYD  
-----END PRIVATE KEY-----
```

Required

Certificate Paste from file

Paste the certificate and optional certificate chain including the BEGIN and END lines shown below

```
-----BEGIN CERTIFICATE-----  
MIID5TCCAs2gAwIBAgIJALCgZq0XI3xiMA0GCSqGSIb3DQEBBQUAMIGIMQswCQYD  
-----END CERTIFICATE-----
```

Cancel Save

4. Click **Save**.

Step 5. Associate Applications to the Container

1. On the WAF-as-a-Service web interface, go to the **APPLICATIONS** tab and select the application that you created in [Step 3. Add an Application](#).
2. Click **Endpoints** in the left panel.
3. On the **Endpoints** page, click **Edit** in the **Deployment Location** section.
4. On the **Edit Location** window:
 1. Select the **Deploy this application to my own container** checkbox.
 2. Select the container you created in [Step 2. Create a Container](#) from the **Container Name** drop-down list.
 3. Click **Save**.

Edit Location ×

Deploy this application as a service, hosted by Barracuda.

Your application's protection will be deployed in this location. Select the location closest to your application servers for the best performance.

Warning: Changing the location of your application will take up to 30 minutes. During this time, your application may experience downtime. Only change the location during a maintenance window.

Automatically select region

Location
Virginia, USA

This location requires a backup location for redundancy.

Backup Location
Washington, USA

Deploy this application in my own container

Container Name

Create and configure a container under **Resources > Container Management**

[Cancel](#) [Save](#)

Step 6. Download the YAML File

1. On the WAF-as-a-Service web interface, go to the **RESOURCES** tab, click **WAF CONTAINERS** in the left panel and click **Container Management**.
2. On the **Container Management** page:
3. Click the three dots under **ACTION** next to the container that you created in [Step 2. Create a Container](#) and select **Deploy**.
4. On the **Container Deployment** window:
 1. **Environment**
 1. Kubernetes is selected by default.
 2. Click **Continue**.
 2. **Container Key**
 1. Select the option to associate the container key that you created in [Step 1. Create a](#)

[Container Key.](#)

1. If you want to insert the container key manually, select **Insert the key into the deployment file manually (recommended)**.
 2. If you have downloaded the container key and want to upload it before the deployment, select **Upload the container-keys key now**.
 3. If you want to paste the key content instead of uploading the file, select **Paste the contents of the container-keys key now**.
2. Click **Continue**.
3. **Advanced Options** - This section is optional. If you want to configure advanced settings, see [Advanced Options Configuration](#).
 1. Click **Continue**.
4. **Download**
 1. Click **Download YAML** to download the deployment file that can be used to create the WAF container into your Kubernetes cluster.

Advanced Options Configuration

Following are the advanced settings for the container deployment:

- **DATAPATH VERSION**
 - Select a Datapath Version. See [Managing the datapath](#) for more information.
- **CONFIG TOKEN**
 - Select the Config Token to be used with this deployment. If you are not sure which token to use, select Token 1. See [Rotating Container Config Keys](#) for more information about using and rotating Config Token.
- **VIRUS SCANNING**
 - Choose whether you want to include Virus Scanning with this deployment.

App Profiles under application shows the option to enable virus scanning even if the virus scanning is not selected in the **Container Deployment > Advanced Options** wizard. However, virus scanning will not take effect until the ClamAV is deployed. Virus scanning can be added later by modifying and redeploying the container.
- **TROUBLESHOOTING**
 - **Disable diagnostic log collection** - When selected, the diagnostic and deployment data is not sent to the Barracuda diagnostic center. If the option is not selected, the data is encrypted and stored in a secure location. It is only made available to support engineers for troubleshooting issues.
 - **Do not send Access Logs and Firewall Logs to WAF-as-a-Service** - When selected, Access Logs and Firewall Logs are not sent to Barracuda WAF-as-a-Service.
 - **Disable handling of Scheduled Events when deployed on Microsoft Azure** - When selected, the scheduled events are not handled when the container is deployed on Microsoft Azure.
- **Support Tunnel**

- **Disable** – When selected, the support tunnel access will not be available for the Barracuda support to troubleshoot issues.
- **Enable** – When selected, the support tunnel access is made available to troubleshoot issues.
- **Allow tunnel to be opened remotely through the WAF-as-a-Service UI or API** – When selected, the support tunnel is allowed to be accessed remotely through the WAF-as-a-Service web interface or API.
- **Allow tunnel to be opened only through a local kubectl command** – When selected, the support tunnel access is allowed ONLY through the local kubectl command when the container is deployed in Kubernetes cluster.
- **Core Dump Collection**
 - **Disable** – When selected, the state of the container (call stack and debug information) is not collected in case of a crash.
 - **Enable** – When selected, the state of the container (call stack and debug information) is collected in case of a crash. The collected information can be used when troubleshooting issues.
 - **Allow collection to be opened remotely through the WAF-as-a-Service UI or API** – When selected, the core dump collection can be enabled through the WAF-as-a-Service web interface or API.
 - **Allow collection to be opened only through a local kubectl command** – When selected, the core dump collection is allowed ONLY through the local kubectl command when the container is deployed in the Kubernetes cluster.

Kubernetes Resources in the Deployment File

The Kubernetes YAML file includes the following objects:

- A **Secret** that communicates with the Barracuda container registry and pulls the WaaS container image from the registry.
- A **Service** that routes the traffic from the public network to the container.
 - 443 and 80 are external ports exposed on public IP
 - 9000 and 9001 are internal ports used by containers to serve the traffic.

Ports 9000 and 9001 are configurable. You can edit the container ports or add a new endpoint with port details on the application's Endpoints page. If the ports are modified, the container should be redeployed.
- A **Deployment** downloads the container image from the Barracuda container repository and deploys 1 replica to your Kubernetes cluster. This container secures the deployed applications by inspecting the incoming traffic for attacks and anomalies.
- A **DaemonSet** downloads the container image from the Barracuda repository and deploys 1 container on every worker node in the Kubernetes cluster. This container collects all information required for troubleshooting and the information is shared with the Barracuda diagnostic center for advanced analysis.

Deploy and Check the Application

On the management workstation where the kubectl is running, execute the following command to create the resources defined in the “waf-container.yaml” definition file:

Open the “waf-container.yaml” file and change the service type from **LoadBalancer** to **NodePort**.

```
kubectl apply -f <path>/waf-container.yaml
```

Here, the <path> indicates the location where the YAML file is saved on your system.

you should see output that looks like the following, indicating your Kubernetes objects were created successfully:

```
secret/acr-secret created
service/svc-{Application Name}-9000 created
deployment.apps/cuda-tm created
daemonset.apps/ds-node-sys created
```

Run the following command to check the WaaS container deployment:

```
kubectl get deployment cuda-tm
```

The output should look like the following:

| NAME | READY | UP-TO-DATE | AVAILABLE | AGE |
|---------|-------|------------|-----------|-----|
| cuda-tm | 1/1 | 1 | 1 | 5m |

Run the following command to verify if the troubleshooting container is up and running:

```
kubectl get daemonset ds-node-sys
```

The output should look like the following:

| NAME | DESIRED | CURRENT | READY | UP-TO-DATE | AVAILABLE | NODE |
|-------------|---------|---------|-------|------------|-----------|------|
| ds-node-sys | 1 | 1 | 1 | 1 | 1 | |
| SELECTOR | AGE | | | | | |
| <none> | 5m | | | | | |

Run the following command to know the public IP of the application:

```
kubectl get services svc-{Application Name}-9000
```

The output should look like the following:

| NAME | AGE | TYPE | CLUSTER-IP | EXTERNAL-IP | PORT(S) |
|--------------------------------------|------------|-------------|---------------------|--------------------|----------------|
| svc- {Application Name} -9000 | | NodePort | {Private IP} | <none> | |
| 443:31000/TCP,80:30347/TCP | 5m | | | | |

Verify if the application is accessible.

Open a browser and access your application using the node IP address:

- For HTTP request: **http://{NodeIP}:30347/**
- For HTTPS request: **https://{NodeIP}:31000/**

For example, if you have 2 nodes with public IP addresses 123.4.5.6 and 12.3.4.5, and Kubernetes assigns your service port 31000, then the service will be accessible on both 123.4.5.6:31000 & 12.3.4.5:31000.

The services should now be accessible through the node IP as shown above. Use the load balancer IP address and port if you have a load balancer setup ahead of your worker nodes.

Figures

1. Deploying_WAAS_Security_Module_on_On_Premises_Kubernetes_Cluster.png
2. Server_Reachable.png
3. Backend_Server_Not_Reachable.png
4. Endpoints.png
5. Edit_Endpoint.png
6. Edit_Location.png

© Barracuda Networks Inc., 2024 The information contained within this document is confidential and proprietary to Barracuda Networks Inc. No portion of this document may be copied, distributed, publicized or used for other than internal documentary purposes without the written consent of an official representative of Barracuda Networks Inc. All specifications are subject to change without notice. Barracuda Networks Inc. assumes no responsibility for any inaccuracies in this document. Barracuda Networks Inc. reserves the right to change, modify, transfer, or otherwise revise this publication without notice.