ALTEON KUBERNETES CONNECTOR USER GUIDE

Document ID: RDWR-AKO_UG2412

December 2024

Copyright Notices

The programs included in this product are subject to a restricted use license and can only be used in conjunction with this application.

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL, please contact openssl-core@openssl.org.

OpenSSL License

Copyright (c) 1998-2011 The OpenSSL Project. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement:

   This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (http://www.openssl.org/)
4. The names “OpenSSL Toolkit” and “OpenSSL Project” must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org.
5. Products derived from this software may not be called “OpenSSL” nor may “OpenSSL” appear in their names without prior written permission of the OpenSSL Project.
6. Redistributions of any form whatsoever must retain the following acknowledgment:

   “This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/)”

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT “AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Original SSLeay License

Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)

All rights reserved.

This package is an SSL implementation written by Eric Young (eay@cryptsoft.com).

The implementation was written so as to conform with Netscapes SSL.

This library is free for commercial and non-commercial use as long as the following conditions are aheared to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com).

TABLE OF CONTENTS

CHAPTER 1 – OVERVIEW

This document details the Alteon Kubernetes Connector operation.

The Alteon Kubernetes Connector

When running an application inside a Kubernetes cluster, you need to provide a way for external users to access the applications from outside the cluster.

You can use an ingress controller to route traffic from outside to your Kubernetes cluster through a load balancer. It simplifies the network traffic management for your Kubernetes cluster and directs the requests to the right services, in other words, it connects the incoming requests for your applications to the specific server that handles them.

The Alteon Kubernetes Connector (AKC) is a solution that integrates Alteon ADC with Kubernetes/ OpenShift orchestration, allowing you to automatically load balance traffic for your Kubernetes workloads.

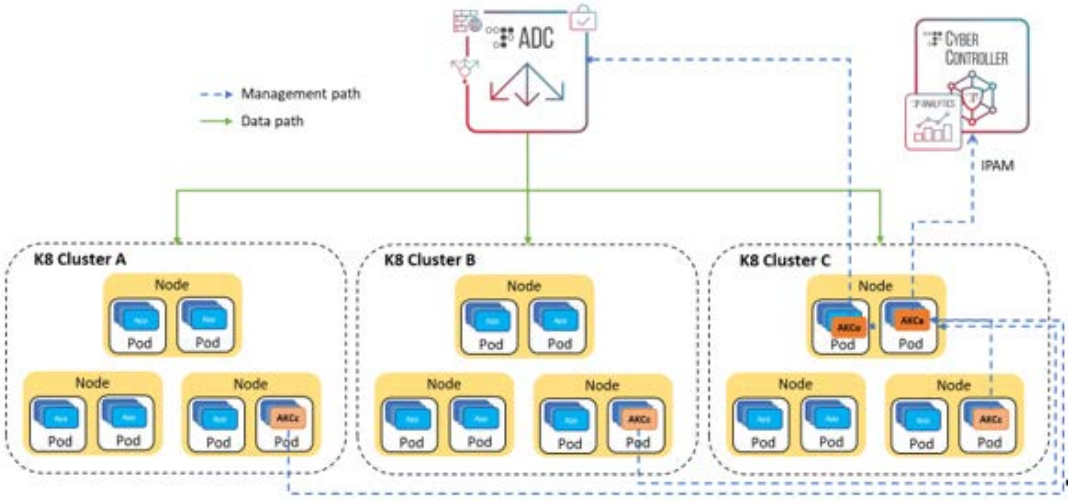
The Alteon Kubernetes Connector (AKC) discovers services in the Kubernetes cluster and translates them to Alteon ADC configuration. It also looks for the addition or removal of nodes/pods from the Kubernetes clusters and communicates the changes to Alteon to keep the Alteon configuration in sync.

Using AKC ability to discover and aggregate service that runs in multiple clusters, Alteon can provide load balancing and high availability between clusters.

The AKC has three components:

- The AKC Controller - discovers the service objects in the Kubernetes clusters. Such a component is required in each of the participating Kubernetes clusters.
- The AKC Aggregator - aggregates inputs from all the controllers and communicates the necessary configuration changes to AKC Configurator.
- The AKC Configurator - prepares Alteon configuration file and pushes it to the Alteon device.

In addition, the solution uses the vDirect module in Cyber Controller (or APSolute Vision), which handles IPAM service, required to allocate service IP (VIP) from an IP pool (currently a single pool is supported).



AKC Deployment Architecture

AKC Controller

The AKC controller fulfills the following tasks:

- It scans the cluster services and nodes to detect new services deployed in the cluster as well as changes in existing services, such as nodes being added or removed.
- When a service is being deployed, the controller also validates that the required parameters are valid - for example the specified SLB metric is available in Alteon, or the specified health check is configured in Alteon. If the validation fails, the operation on the service will not be permitted by Kubernetes and hence that service won't be opened in Alteon.
- It sends to the aggregator the map of services and nodes/pods relevant for Alteon.

Controller Reconciliation

To detect and correct any discrepancies that may arise due to failures or changes in the cluster's resources, the controller performs periodic cluster resync process. The controller retrieves the current state of the resources from the Kubernetes API server and compares it to its internal state. If there are any differences, the controller takes the necessary actions to bring its internal state in line with the current state of the resources. The interval at which AKC Controller performs reconciliation is defined during the controller deployment.

AKC Aggregator

The AKC Aggregator is deployed in one of the clusters and when it receives new configuration maps from the AKC controllers it performs the following tasks:

- If the namespace and service name combination is new, and if static IP was not mentioned in the service manifest, it contacts Cyber Controller vDirect module IPAM service to allocate an IP address for this new service (VIP).
- It prepares Alteon configuration updates (either new virtual service, group and servers, or just new servers for existing service, when the K8s service is already deployed in additional clusters) and sends it to the AKC Configurator.

  When the same service LoadBalancer resource, with the same service name and same namespace is deployed in multiple clusters, this component aggregates the nodes/pods relevant to this service from the different clusters as real servers belonging to the same service. This enables Alteon to distribute the service traffic between nodes/pods in different clusters. In such scenario it is important that the AKC annotations used for a service have the same value in all clusters deployment, otherwise there will be a conflict and service deployment on Alteon will fail.

  Note that same service aggregation over multiple clusters is only supported for service LoadBalancer resource deployment and not for ingress resource deployment.
- Once Alteon configuration is completed and applied, the Aggregator updates AKC Controller with the new service IP.

Aggregator Reconciliation

To ensure the aggregator recovers from unknown errors and missed events, it performs periodic reconciliation process with the AKC Controllers.

The aggregator clears its internal service tables and sends a request for configuration maps to all the controllers. It then aggregates all answers, rebuilds updated service tables and sends latest configuration to AKC Configurator.

AKC Configurator

The AKC Configurator is deployed in the same cluster as the AKC Aggregator. When it receives from the aggregator the necessary configuration changes:

- It translates them to Alteon configuration and prepares updated configuration file
- Pushes the new configuration file to the Alteon device
- Applies and saves the new configuration

The AKC creates and updates the following type of Alteon objects: virtual server, virtual service, group and server.

- The id of all the objects created by AKC for a service starts with AKC_<K8s service name>, for easy identification.
- Configuration of such objects should not be manually edited on Alteon - they will be overwritten on the next configuration update performed by the configurator.

Important! Only one Aggregator and one Configurator can operate vis-a-vis an Alteon device/pair. When an Alteon device/pair manages traffic to multiple Kubernetes clusters the Aggregator and Configurator components must be installed in only one of the clusters, and only the Controller component must be installed in all the clusters.

Related Alteon Documentation

Alteon Application Switches have the following related documentation, which is required to regularly manage the Alteon platform.

- Alteon Installation and Maintenance Guide
- Alteon Command Line Interface Reference Guide
- Alteon Command Line Interface Application Guide
- Alteon Web-Based Management Application Guide
- Alteon Web-Based Management Getting Started Guide
- Alteon OS Release Notes

CHAPTER 2 – AKC DEPLOYMENT

This section describes the procedure required for deploying the AKC components, Aggregator, Configurator, and Controller, in a Kubernetes or OpenShift environment. It includes the following topics:

Note: Only one cluster should have all the AKC components deployed (Aggregator, Configurator, and Controller). All the other clusters should have only a Controller installed.

Deployment Prerequisites

Before deploying the Alteon Kubernetes Controller the following tasks are required:

1. The Alteon device that will be load balancing the K8s workloads is installed and its management interface accessible from the Kubernetes clusters.
2. If you wish to dynamically allocate service IPs (VIPs) make sure that:
   - Cyber Controller (or legacy APSolute Vision) is installed and is accessible from the Kubernetes clusters.
   - The IPAM service is up and running, an IPAM pool is configured, and an IPAM workflow created. See IPAM Service Configuration, page 12.
3. Ensure that you have the following versions installed:
   - OpenShift version 4.10 or later
   - Kubernetes version 1.24.2 or later
4. Prepare AKC files for installation:
   - Download the AKC package from the Radware Customer portal.
   - From the AKC package file (AKC_<version>.tgz) extract the AKC_<version>.tar file.
   - Extract the content of the AKC_<version>.tar.
5. Generate SSL key and certificate for AKC Controller use and create YAML secret for it. This is not required if you are deploying AKC in default namespace. See Non-Default Controller Key and Certificate Pair, page 16.

Deploying Aggregator, Configurator, and Controller on a Cluster

The following procedure is required for deploying the Aggregator and Controller on the same cluster.

1. Add the **akc_aggregator_<version>.tar, akc_configurator_<version>.tar** and **akc_controller_<version>.tar** container images provided in the AKC package (found under the **services** directory) to the Kubernetes/OpenShift environment image repository.
2. Install the Helm application on your cluster.

3. Copy the AKC Helm file (**akc-<version>.tgz** file found under Helm directory) to your local registry.
4. Connect to the cluster where you want to install the Aggregator and the Controller.
5. Install the helm chart. For an explanation on the helm parameters see Chapter 4. (An explanation is also available in the README file provided with the release.)

Below is an example for AKC version 1.4.0:

```
$ helm install radware-akc akc-1.4.0.tgz \
--set global.namespace=default \
--set controller.enabled=true \
--set controller.uid=akc_local \
--set controller.aggregator_ip=akc-aggregator.default.svc.cluster.local \
--set controller.image.repository=myrepository:8081/dev/akc-controller \
--set controller.image.tag=8 \
--set controller.controller_reconcile_period=3600 \
--set controller.webhook.enabled=true \
--set controller.secret.crt=base64 encoded certificate \
--set controller.secret.key=base64 encoded key
--set aggregator.enabled=true \
--set aggregator.configurator.ip=akc-configurator.default.svc.cluster
.local\
--set aggregator.configurator.port=30188 \
--set aggregator.configurator.protocol=http \
--set aggregator.ipam.workflow=ipam_workflow \
--set aggregator.ipam.pool=ipam_pool \
--set aggregator.ipam.ip=10.175.150.228 \
--set aggregator.ipam.port=2189 \
--set aggregator.ipam.protocol=https \
--set aggregator.ipam.user=admin \
--set aggregator.ipam.pass=radware \
--set aggregator.image.repository=myrepository:8081/dev/akc-aggregator \
--set aggregator.image.tag=latest \
--set configurator.enabled=true \
--set configurator.alteon.master_ip=10.175.101.203 \
--set configurator.alteon.backup_ip=10.175.101.205 \
--set configurator.alteon.user=admin \
--set configurator.alteon.pass=admin1 \
--set configurator.image.repository=myrepository:8081/dev/akc-configurator
--set configurator.image.tag=latest
```

Note: If service IPs will be manually allocated there is no need for the aggregator.ipam parameters.

Deploying the Controller

The following procedure is required for deploying the Controller only.

1. Add the **akc_controller_<version>.tar** files provided in the AKC package (found under the **services** directory) to the Kubernetes/OpenShift environment image repository.
2. Install the Helm application on your cluster.
3. Copy the AKC Helm file (**akc-<version>.tgz** file found under Helm directory) to your local registry.
4. Install the helm chart. For an explanation on the helm parameters please see Chapter 4. (An explanation is also available in the README file provided with the release.)

Below is an example for AKC version 1.4.0:

```
$ helm install radware-akc akc-1.4.0.tgz \
--set global.namespace=default \
--set controller.enabled=true \
--set controller.uid=akc_remote \
--set controller.aggregator_ip=10.2.2.2 \
--set controller.aggregator_port=30051 \
--set controller.image.repository=myrepository:8081/dev/akc-controller \
--set controller.image.tag=8 \
--set controller.controller_reconcile_period=3600 \
--set controller.webhook.enable=true \
--set controller.secret.crt=base64 encoded certificate \
--set controller.secret.key=base64 encoded key \
--set aggregator.enabled=false
```

Replacing a Cluster with Aggregator Deployment

The following procedure is required for replacing a faulty cluster that was deployed with an Aggregator, Configurator, and Controller.

1. If possible, uninstall the Aggregator, Configurator, and Controller on the cluster you want to replace, by entering the following command:

   ```
   helm uninstall radware-akc
   ```

2. If there is no communication to the cluster you want to replace, such that you cannot uninstall the helm, in order to avoid collisions, redeploy the Aggregator, Configurator, and Controller on the new cluster by using the procedure detailed in the section Deploying Aggregator, Configurator, and Controller on a Cluster, page 9 above.
3. Check the new Aggregator IP and redeploy the Controller with the new Aggregator IP on all other clusters that should have Controller only deployment by using the procedure detailed in the section Deploying the Controller, page 11 above.

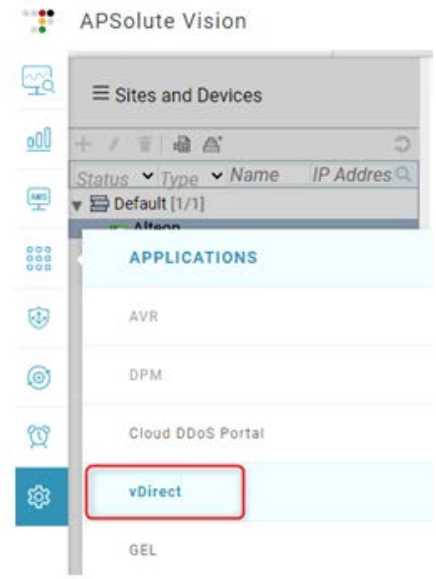
IPAM Service Configuration

The following procedure is required to allow AKC to allocate service IPs via IPAM (IP address management). If the service VIP is allocated manually upon service deployment this section is not required.

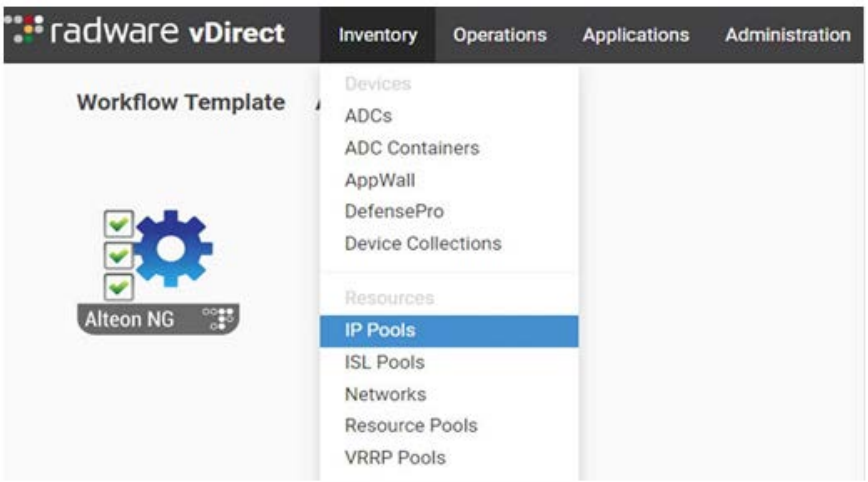
- Configure IP Pool
- Upload and activated the vDirect workflow that handles the IPAM.

To configure IPAM services

1. Login into Cyber Controller.
2. Open **vDirect** from the *Applications* menu.

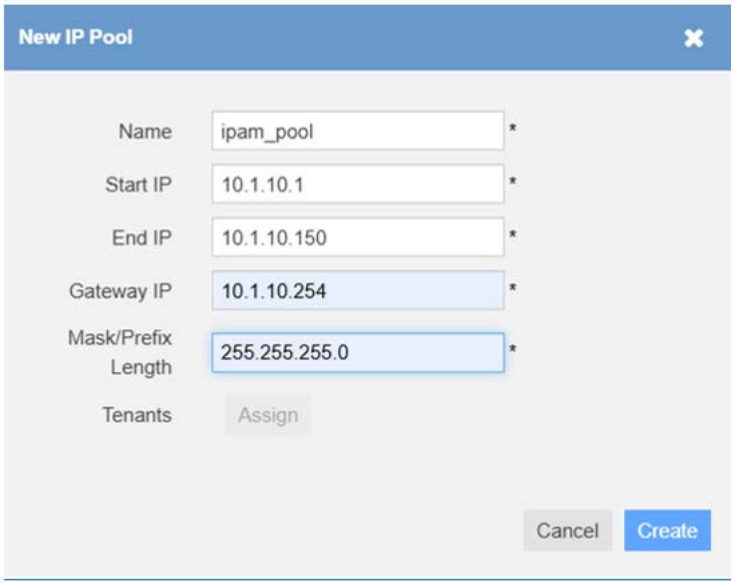


3. Go to **Inventory -> IP Pools.**



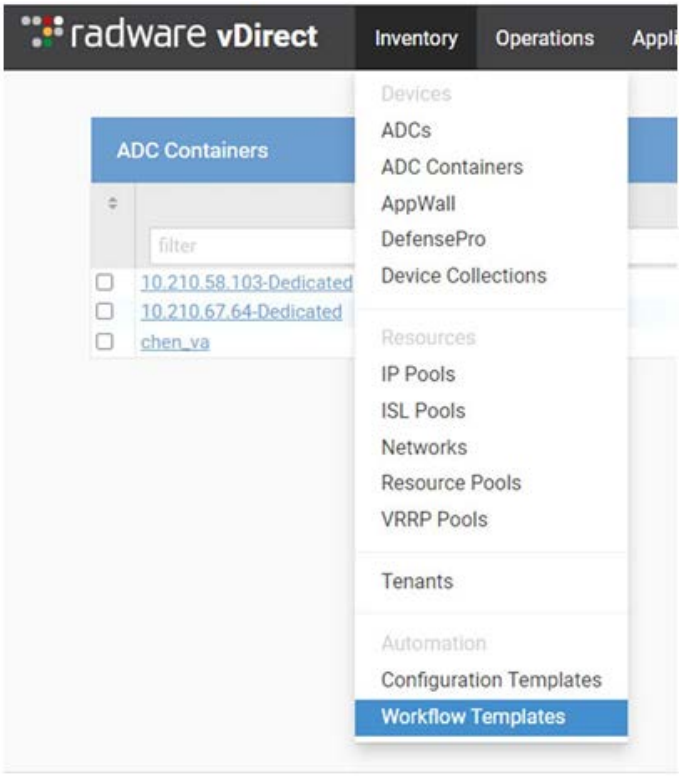
4. Click on the **+** icon to create a new IP Pool.

5. Provide a valid range of IP addresses.



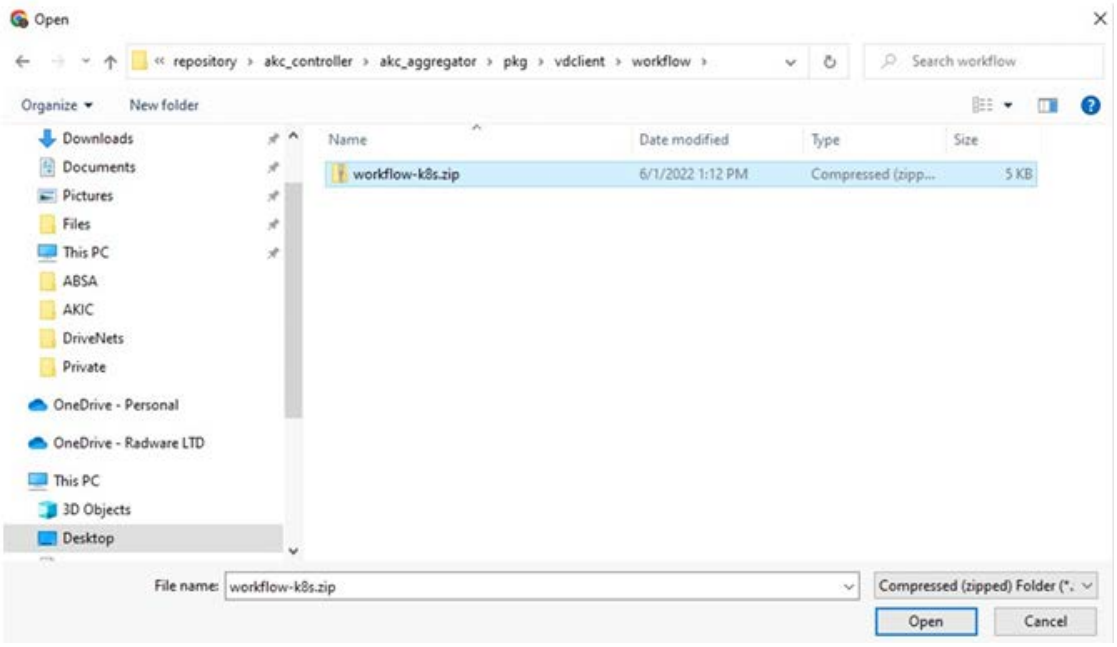
6. In OpenShift an External IP subnet must be added for this IP pool (see Appendix A. page 27).

7. Upload to vDirect the workflow file received with the installation package.
   a. Select **Inventory -> Workflow Templates**.

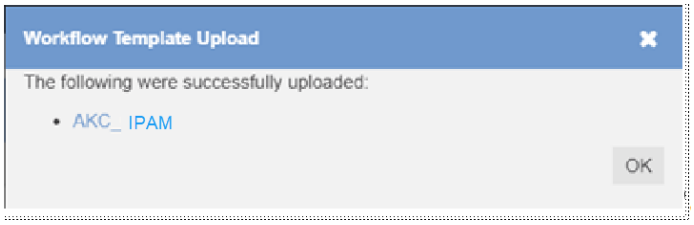


   b. On the right side of the screen click on the arrow button and select **Upload workflow template ZIP archive(s)**.

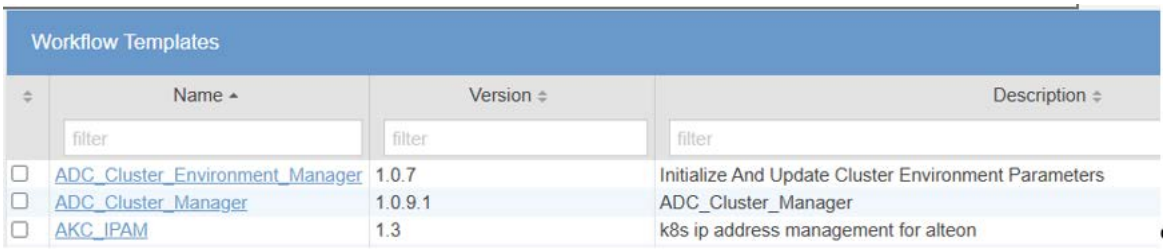
c. Select the **akc_vd_files.zip** workflow file to upload and click **Open**.



d. On the *Workflow* Template Upload screen, click **OK**.



e. Now you should be able to see the uploaded workflow in the list of workflow templates.

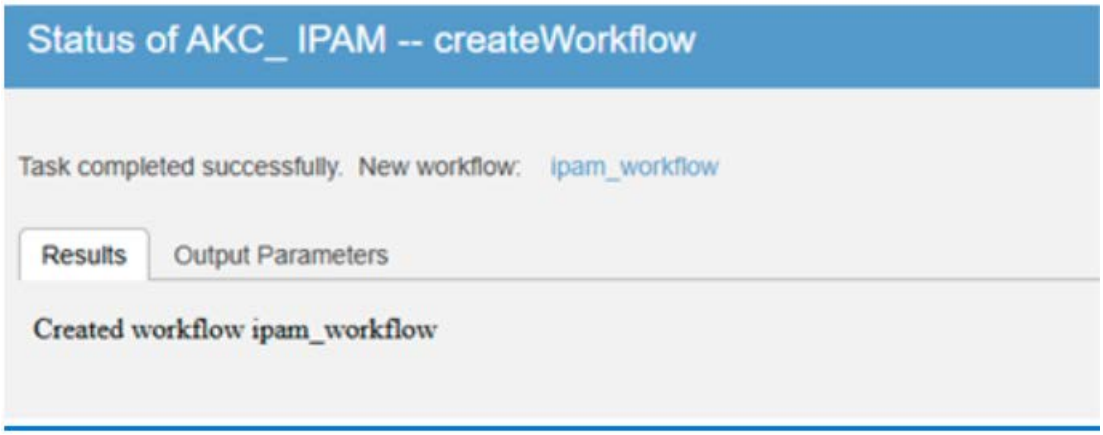


8. Activate the vDirect workflow:
   a. From the list of workflow templates select **AKC_IPAM**.
   b. Click on the **Create Workflow** button.

c. Select the Alteon device (or pair of Alteon devices for HA), select the IP Pool defined for IPAM, and click Run.

Note: Currently activation of this workflow requires to select an Alteon device from the Cyber Controller inventory. Please ensure that the Alteon device is in the Cyber Controller inventory.

d. A workflow instance will be created.



Non-Default Controller Key and Certificate Pair

The AKC Controller validation webhook acts as an HTTPS server and receives validation requests from Kubernetes api-server. As an HTTPS server it is required to associate an SSL certificate and private key to it.

By default, the AKC controller will include an SSL key/cert pair generated with 2K RSA key and 10 years validity. This default key/certificate pair is relevant only if the controller is deployed in default namespace. If you want to deploy the controller on different namespace, you have to provide your own key & certificate.

To configure user-defined key and certificate pair

1. Generate key and certificate pair using any method or tool. The common name and SAN used in the certificate should be **akc-controller.<namespace>.svc** (for default namespace it would be **akc-controller.default.svc**)

   OpenSSL example: **openssl req -new -x509 -subj "/CN=akc-controller.default.svc" -addext "subjectAltName=DNS: akc-controller.default.svc " -nodes -newkey rsa:2096 -keyout tls.key -out tls.crt -days 3650**

2. Generate the "secret" YAML using the certificate and key that was generated in the step 1.

   kubectl create secret generic alteon-svc-validator-cert -n default --from-file tls.crt --from-file tls.key --dry-run=client -oyaml > AlteonSvcValidator_secret.yaml

   The secret YAML will be generated in the below format.

   ```
   apiVersion: v1
   data:
     tls.crt: "base64 encoded certificate"
     tls.key: "base64 encoded key"
   kind: Secret
   metadata:
     creationTimestamp: null
     name: alteon-svc-validator-cert
     namespace: default
   ```

3. Use the certificate and key during AKC Controller deployment or update existing deployment. Copy the base64 encoded certificate & base64 encoded key into the controller secret HELM variables **controller.secret.crt** and **controller.secret.key**.
   (The HELM will update the webhook manifest internally).

CHAPTER 3 – SERVICE LOAD BALANCER DEPLOYMENT

A Service is an abstract way to expose an application running on a set of Pods as a network service.

When there is a need to expose a Service to an external IP address that is outside of your cluster, Kubernetes ServiceTypes allow you to specify what kind of Service you want (default is ClusterIP). On cloud providers which support external load balancers, ServiceType LoadBalancer is used to provision the load balancer.

When AKC (Alteon Kubernetes Connector) is deployed in your Kubernetes clusters, it allows to provision Alteon as an external load balancer and provide load balancing between multiple clusters and cluster nodes.

In addition, Alteon can provide SSL offload as well as application security (WAF, API protection and BotM) for the deployed Kubernetes services.

AKC supports the following connectivity options between Alteon and the service pods:

- **NodePort:** By default, when a service of type LoadBalancer is deployed, a NodePort is created for that service. Therefore, the default AKC mode connects to the NodePort on cluster nodes. This is done by configuring the cluster nodes as real servers on Alteon, with the NodePort as the server port. Currently, AKC configures real servers on Alteon for each node in the cluster, which can result in a large number of real servers per service, particularly in multi-cluster environments. This could limit the number of services Alteon can handle. To manage this, the number of nodes for which real servers will be created can be limited by configuring the aggregator.max_nodes_per_cluster parameter during AKC Aggregator installation.
- **Direct Pod Access:** In this mode, AKC configures the IPs of the relevant service pods as real servers, using the target port as the server port. To enable this mode, the controller.direct_access parameter must be set during AKC Controller deployment. Additionally, to allow Alteon to reach the pod IPs, a static route must be configured on Alteon for each node's pod network, with the node's IP serving as the gateway.

  This mode is supported from AKC version 1.5.0 and up.

When a new service of type LoadBalancer is deployed in the cluster, the following occurs:

- If the service has the label **AlteonDevice: "true"**, the AKC Controller communicates with the AKC aggregator and provides the service parameters.
- If this namespace and service name combination is new, and a static IP is not mentioned in the service manifest, the Aggregator contacts Cyber Controller module IPAM service to allocate an IP address for service (VIP).
- The AKC Aggregator prepares Alteon configuration updates and sends it to the AKC Configurator
- The AKC Configurator prepares the configuration file and applies it to Alteon.
- Once Alteon configuration is completed and applied, the Aggregator updates AKC Controller with the new service IP.
- The AKC Controller updates the service IP address in the Kubernetes cluster. The kubectl get service -o yaml command can be used to see the allocated external IP address, if it was allocated from IPAM.

AKC Controller discovers resources that have the following parameters:

- The resource type is Service (kind: Service)
- The Service type is LoadBlancer (type:LoadBalancer)
- Alteon label (AlteonDevice): "true"

In addition, the following annotations are supported:

Annotation	Description	Example	Default Behavior	Required/ Version
akc.radware.com/ static-ip: <virtual-ip>	Allows to specify the IP to be used as service Virtual IP (VIP).	akc.radware.com/ static-ip: 192.1681.10	VIP is allocated from the IPAM pool.	Optional Version 1.4.0
akc.radware.com/lb-algo: <metric>	Allows to specify load balancing algorithm other than default.	akc.radware.com/ lb-algo: lestconns		Required Version 1.0.0
akc.radware.com/lb-health-check: <hc id>	Allows to specify health check other than default. The health check id must exist on Alteon (either pre-defined or user-defined health checks).	akc.radware.com/ lb-health-check: http	TCP health check is used	Optional Version 1.0.0
akc.radware.com/ sslpol: <ssl policy id>	Allows to specify an SSL Policy ID on Alteon device that should be attached to this service. The SSL Policy can be configured directly on Alteon or via CRD (see SSL Policy Custom Resource, page ??). Note: An SSL policy configured via CRD can be specified in one of two ways: • Using the full Alteon name, AKC_<namespace>_ <CR name> • Using the custom resource name (CR name). In this case AKC will first search Alteon configuration for this name, since SSL policy can also be directly configured on Alteon. If not found, AKC will add the "AKC_<namespace>_ _" prefix and search again. Valid only for service port 443.	akc.radware.com/ sslpol: akc-default		Required if **cert** annotation is present Version 1.3.0

Annotation	Description	Example	Default Behavior	Required/ Version
akc.radware.com/ cert: <cert id>	Allows to specify a certificate and key pair on Alteon device to be attached to this service. The SSL certificate and key can be generated or imported directly on Alteon or via K8 TLS secret. When using K8 TLS secret, the secret must be pre-defined in the K8 cluster with the AlteonDevice: "true" label, for the configuration import to succeed. Only after a certificate and key are created in Alteon, user can provide the secret name, as it appears in Alteon and not K8 (AKC_<namespace>__<K8 secret name>,) to the service configuration. Valid only for service port 443.	akc.radware.com/ cert: WebManagement Cert		Required if **sslpol** annotation is present Version 1.3.0
akc.radware.com/ secpath: <secpath policy id>	Allows to specify a SecurePath Policy ID on Alteon device that should be attached to this service. Valid only for service port 80 or 443. SecurePath connector provides application security (Cloud WAF, API protection & BoTM) for the deployed service. This parameter must come together with Sideband policy.	akc.radware.com/ secpath: secpath1		Mandatory only if **sideband** annotation is present Version 1.3.0
akc.radware.com/ sideband: <sideband policy id>	Allows to specify a Sideband Policy ID on Alteon device that should be attached to this service. This parameter is required for SecurePath connector. Valid only for service port 80 or 443.	akc.radware.com/ sideband: sideband1		Mandatory only if **secpath** annotation is present Version 1.3.0

Annotation	Description	Example	Default Behavior	Required/ Version
akc.radware.com/ secwa: <inline|outofpath>	Allows to enable integrated WAAP application protection and specify the required operation mode – inline or out-of-path. Notes: -The AppWall module must be enabled on the Alteon device before deploying Kubernetes services with this capability. - The name of the Secure Web Application (secwa) created on Alteon is AKC_<first 11 namespace chars>_<hash (16chars>. - The detailed WAAP security policies for this service must be configured directly on Alteon device.c	akc.radware.com/ secwa:outofpath		Optional Version 1.6.0
akc.radware.com/ http2https-redirect: '[<http port>]:<https port>'	Allows to specify for an HTTPS service that is deployed, that traffic that arrives as HTTP must be redirected to HTTPS. The HTTPS port must be specified and must appear as service port in the manifest. The default for HTTP port is 80.	akc.radware.com/ http2https-redirect:'80:443' Note: Default source port (80) can also be presented without a value: http2https-redirect: :443		Optional Version 1.5.0
akc.radware.com/ appshape-scripts:<script1, script2, …>	Allows to attach AppShape++ scripts available in Alteon to the service. The scripts priority in the service is determined by their order in the annotation. It supports to attach up to 16 scripts per service (up to 15 if SecurePath is enbled on the service).	akc.radware.com/ appshape-scripts: redirecttogoogle, addheaders		Optional Version 1.6.0

Annotation	Description	Example	Default Behavior	Required/ Version
akc.radware.com/ appshape-alwayson: <enble|disable>	Specifies whether a virtual service should always be available, even if all servers are down, when an AppShape++ script is attached to the service. This parameter must only be enabled when one of the attached AppShape++ scripts contains treatment for the “no server available” state (such as returning the Sorry page or redirecting to a special URL).	akc.radware.com/ appshape-alwayson:enable	disable	Optional Version 1.6.0

Following is an example of a deployment of a service with SSL offload and SecurePath connector:

```
apiVersion: v1
kind: Service
metadata:
  annotations:
    akc.radware.com/lb-algo: leastconns
    akc.radware.com/lb-health-check: http
    akc.radware.com/sslpol: AKC_default_policy
    akc.radware.com/cert: cert1
    akc.radware.com/sideband: sideband1
    akc.radware.com/secpath: secpath1

  labels:
    AlteonDevice: “true”
  name: test-service-http
  namespace: default
spec:
  ports:
    - name: https
      port: 443
      targetPort: 8080
  selector:
    app: lab
  type: LoadBalancer
```

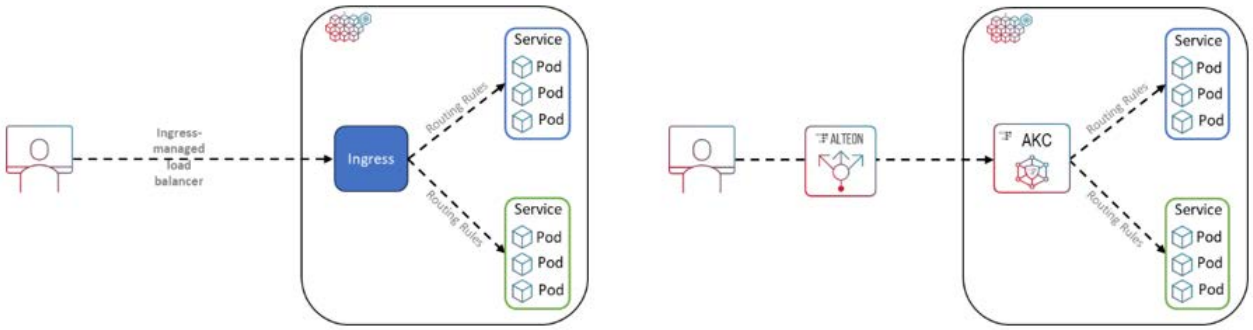
Note: The port number that appears in the deployment yaml in parameter "port" is configured as the virtual service port in Alteon.

Note: When a service is deployed in multiple clusters, ensure that the service is defined identically in all the clusters. When AKC identifies a multi-cluster mismatch, such as multi cluster service parameters (metric, health check, etc.), new configuration will not be sent to Alteon until this mismatch is resolved

CHAPTER 4 – INGRESS SERVICE DEPLOYMENT

The Ingress resource in Kubernetes allows external access to services within the cluster by exposing HTTP and HTTPS routes. The routing of traffic is governed by rules defined on the Ingress resource.

An Ingress may be configured to provide services with externally-reachable URLs, load balance traffic, terminate SSL/TLS, and offer name-based virtual hosting. An Ingress controller is responsible for fulfilling the Ingress, usually with a load balancer, for example Alteon.



To create an Ingress object, you need to specify the apiVersion, kind, metadata and spec fields. The name of the Ingress object should follow the rules of a valid DNS subdomain name. The spec field contains all the details for configuring a load balancer or proxy server. The most important part is a list of HTTP rules that match the incoming requests and direct them to the appropriate services.

An HTTP rule contains the following information:

- A host (optional). If a host is provided (for example, foo.bar.com), the rules apply to that host. If no host is specified, the rule applies to all inbound HTTP traffic through the IP address specified.
- A list of paths (for example, /foo). Each path has an associated backend defined, that includes service.name and a service.port.name or service.port.number. Both the host and path must match the content of an incoming request before the load balancer directs traffic to the referenced Service.

A defaultBackend is often configured in an Ingress controller to service any requests that do not match a path in the spec.

When AKC (version 1.4.0 or higher) is deployed in your Kubernetes clusters, it allows you to provision Alteon as an external load balancer fulfilling the ingress. A Content Rule is automatically configured on Alteon for each HTTP rule.

To make an Ingress secure, you need to provide a Secret that has a TLS private key and certificate. The Ingress resource can only use one TLS port, 443, and it assumes that the TLS connection ends at the ingress point (Alteon in our case), the traffic to the Service and its Pods is not encrypted. The TLS secret must have keys named tls.crt and tls.key that have the certificate and private key for TLS.

A secret must be pre-defined in the Kubernetes cluster with the AlteonDevice: "true" label, for the configuration import to succeed. Only after a cert and a key are created in Alteon, user can provide the secret name (as it appears in the Kubernetes cluster) to the ingress tls configuration.

If you have different hosts in the TLS configuration section of an Ingress, they share the same port based on the hostname given by the SNI TLS extension (AKC uploads all the private keys and certificates specified via TLS secret to Alteon and aggregates them in a certificate group attached to the HTTPS virtual service).

When fulfilling ingress Alteon provides load balancing at pod level. Alteon sends traffic directly to the Pods, bypassing the internal load balancing mechanism of the Kubernetes cluster.

This can be achieved in one of two ways:

- **BGP** - Relevant for Kubernetes cluster only (and not Openshift), requires the Calico Container Network Interface (CNI) plugin.
- **Direct Pod Access** - Relevant for both Kubernetes and Openshift (OVN) clusters, requires configuring a static route on Alteon for each node's pod network, with the node's IP serving as the gateway.

When a new Ingress service is deployed in the cluster , with ingressClasName set to "akc", the following occurs:

- The AKC Controller communicates with the AKC aggregator and provides the service parameters.
- If the namespace and service name combination is new, and a static IP is not mentioned in the service manifest, the Aggregator contacts Cyber Controller or vDirect module IPAM service to allocate an IP address for service (VIP).
- The AKC Aggregator prepares Alteon configuration updates and sends it to the AKC Configurator
- The AKC Configurator prepares the configuration file and applies it on the Alteon.
- Once Alteon configuration is completed and applied, the Aggregator updates AKC Controller with the new service IP.
- The AKC Controller updates the service IP address in the Kubernetes cluster.
  The **`kubectl get service -o yaml`** command can be used to see the allocated external IP address.

AKC Controller discovers resources that have the following parameters:

- The resource type is Ingress (kind: Ingress)
- The ingress class is akc (ingressClassName: akc)

In addition, the following annotations are supported with Ingress resource:

- Service IP (VIP on Alteon): akc.radware.com/static-ip: <virtual-ip>. If this annotation is not present, AKC allocates IP using the IPAM service.

Following is an example of a deployment of a simple ingress service secured with TLS:

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: simple-fanout-example-1
spec:
  ingressClassName: akc
  tls:
  - hosts:
    - foo.bar.com
    secretName: my-secret
  defaultBackend:
    service:
      name: test-ingress
      port:
        number: 80
```

```
rules:
- host: foo.bar.com
  http:
    paths:
    - path: /foo
      pathType: Prefix
      backend:
        service:
          name: test-ingress
          port:
            number: 80
```

AKC supports ingress resource per cluster, multi-cluster ingress resource is not currently supported.

CHAPTER 5 – CUSTOM RESOURCE DEFINITIONS

Custom Resource Definitions (CRDs) are a powerful feature in Kubernetes that allow users to extend the Kubernetes API by defining their own custom resource types. This capability enables the management of application-specific objects as if they were native Kubernetes resources. By creating a CRD, you can introduce new resource types that suit your application's needs, such as a custom database or a specific configuration object. Once a CRD is defined and applied to the cluster, Kubernetes will recognize and manage instances of the custom resource. CRDs are fully integrated with Kubernetes tools like kubectl, allowing for declarative management of these resources.

Additionally, custom controllers can be developed to automate the desired state management of custom resources, ensuring that the actual state of the cluster matches the specified state. This extensibility makes CRDs a versatile tool for developers looking to tailor Kubernetes to their specific use cases, providing a seamless way to incorporate custom logic and workflows into the Kubernetes ecosystem.

SSL Policy Custom Resource

When deploying a service of type Load Balancer you can request that Alteon perform SSL offload by specifying an SSL policy and server certificate.

The SSL Policy can be configured directly on the Alteon device, or it can be configured via AKC using a CRD.

Radware provides a Kubernetes CustomResourceDefinitions (CRDs) called AlteonSSLPolicy that you can use with the Alteon Kubernetes Controller to configure an SSL policy on the Alteon devices.

The Alteon SSL Policy CRD provides attributes for the various options that are required to define the SSL policies on Alteon.

Table 1: AlteonSSLPolicy Components

Parameter	Type	Default	Description
secRenegotiation	Integer	5	Specifies the maximum number of allowed secure renegotiations per SSL session. Values: 0 - 1024.
frontEndSSL	Object	N/A	Parameters for the client side SSL connection
backEndSSL	Object	N/A	Parameters for the server side SSL connection

Table 2: frontEndSSL Components

Parameter	Type	Default	Description
encryption	String	Enable	Specifies whether to establish an SSL connection with the client and allow decryption/encryption of client traffic. Allowed options are Enable and Disable.

Table 2: frontEndSSL Components

Parameter	Type	Default	Description
supportedCiphers	String	Main	Allows to specify the cipher suite to propose during SSL handshake, as a list of specific ciphers, in OpenSSL format, delimited by ":". Example: "TLS_AES_256_GCM_SHA384:TLS_AES_128_GCM_SHA256". The default is Alteon pre-defined list of ciphers called Main.
allowedSSLVersions	Object	N/A	Allows to specify the SSL/TLS versions acceptable.
sessionReuse	String	Inherit	Specifies whether to enable or disable front-end session reuse for this policy, or to inherit the globally defined setting. Allowed options are Enable, Disable and Inherit.

Table 3: backEndSSL Components

Parameter	Type	Default	Description
encryption	String	Disable	Specifies whether to establish an SSL connection towards the server and allow decryption/encryption of server traffic. Allowed options are Enable and Disable.
includeSNI	String	Enable	Specifies whether to include or not SNI in ClientHello message to the backend server. Allowed options are Enable and Disable.
supportedCiphers	String	Main	Allows to specify the cipher suite to support during SSL handshake, as a list of specific ciphers, in OpenSSL format, delimited by ":". Example: "TLS_AES_256_GCM_SHA384:TLS_AES_128_GCM_SHA256". The default is Alteon pre-defined list of ciphers called Main.
allowedSSLVersions	Object	N/A	Allows to specify the SSL/TLS versions to be used on the backend SSL connection.
sessionReuse	String	Inherit	Specifies whether to enable or disable back-end session reuse for this policy, or to inherit the globally defined setting. Allowed options are Enable, Disable and Inherit.

Table 4: allowedSSLVersions Parameters

Parameter	Type	Default	Description
TLS1_0	String	Disable	Specifies whether to allow TLS 1.0 SSL handshakes. Allowed options are Enable and Disable.

Table 4: allowedSSLVersions Parameters

Parameter	Type	Default	Description
TLS1_1	String	Disable	Specifies whether to allow TLS 1.1 SSL handshakes. Allowed options are Enable and Disable.
TLS1_2	String	Enable	Specifies whether to allow TLS 1.2 SSL handshakes. Allowed options are Enable and Disable.
TLS1_3	String	Enable	Specifies whether to allow TLS 1.3 SSL handshakes. Allowed options are Enable and Disable.

In order for AKC to recognize an SSL policy CRD, the following values are required for apiVersion and kind parameters:

apiVersion:alteon.radware.com/v1

kind:alteonsslpolicy

Example:

```
apiVersion:alteon.radware.com/v1
kind:alteonsslpolicy
metadata:
  name: example_ssl_policy
  namespace:alteonexamples
spec:
  frontEndSSL:
    supportedCiphers:
"TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
"
    allowedVersions:
      TLS1_2: "Disable"
  backEndSSL:

    encryption: "Enable"
```

The above example will result on Alteon in an SSL policy with ID AKC_alteonexamples__example_ssl_policy, that enables SSL for both client side and server side and allows only TLS 1.3 version and ciphers on the client side.

Important: The ID of the SSL Policy configured in Alteon is "AKC_<namespace>__<custom resource name>" and it allows for the maximum namespace length (63 chars) and custom resource names of up to 59 characters (maximum Alteon SSL Policy ID is 128 chars). During service LoadBalancer deployment if the SSL policy specified in annotations is one configured via CRD, the SSL policy can be specified in one of two ways:

- The full Alteon name, for example *AKC_alteonexamples__example_ssl_policy*
- The custom resource name, for example *example_ssl_policy*. In this case AKC will first search Alteon configuration for this name, since SSL policy can also be directly configured on Alteon and referenced in Kubernetes. If not found, AKC will add the "*AKC_<namespace>__*" prefix and search again.

APPENDIX A – HELM CHARTS PARAMETERS

Option Name	Description
Global Options – valid for all components	
global.namespace	The namespace in which to install the AKC components.
Aggregator Options	
aggregator.enable	Specifies whether to install the aggregator. Values are true and false. Note: When only installing the controller, this parameter must also be present, with value false.
aggregator.image.repository	The name of the docker image.
aggregator.image.pullPolicy	The imagePullPolicy for a container and the tag of the image affect when the kubelet attempts to pull (download) the specified image.
aggregator.image.tag	The tag of the docker image (e.g., latest).
aggregator.service.nodePort	The port via which the aggregator service can be accessed from outside the cluster. The same port is used for the connection between controllers installed in different clusters than the aggregator, and the aggregator. Should have the same value as the "controller.aggregator_port" parameter for those controllers. Default value: 30051
aggregator.env.max_inactivity	The time the aggregator waits from the last update from any controller until it applies the configuration on Alteon. Default: 30 sec.
aggregator.env.max_delay	The time the aggregator waits from receiving the first controller update until it applies the configuration on Alteon. Default: 120 sec.
aggregator.env.apply_retry	The time the aggregator waits from the last Alteon Apply failure until it retries to Apply the configuration on Alteon. Default: 600 sec.
aggregator.env.reconciliation_to	The interval between reconciliation processes, in seconds. Default: 43,200 seconds (12 hours).
aggregator.env.ageout_to	The time, in seconds, that the aggregator waits for keep-alive message from a controller before considering the controller cluster as down and removing it from its database. Default: 120 sec
aggregator.env.assets_sync_period	The time interval, in minutes, at which the aggregator will synchronize assets with Alteon configuration. Default: 2 min.
aggregator.grpc_conn.is_grpc_encrypted	Defines whether the GRPC connection between controller and aggregator is encrypted. Default: false.

Option Name	Description
aggregator.rest_conn.is_rest_encrypted	Defines whether the REST connection between controller and aggregator is encrypted. Default: false.
aggregator.configurator.client	Define type of configurator client. The options are: -configurator module within the aggregator (cfgclient) -vDirect module in Vision/Cyber Controller as configurator (vdcclient) - legacy option. Default: cfgclient
aggregator.configurator.ip	The IP/hostname of the AKC configurator service. Because the configurator is running on the same cluster as the aggregator, this parameter should be set to the configurator's default service name - "akc-configurator.[namespace].svc.cluster.local". This is mandatory when is_rest_encrypted=true
aggregator.configurator.port	The port on which AKC configurator service listens for aggregator messages. Default: 30188
aggregator.configurator.protocol	The protocol (HTTP or HTTPS) the aggregator uses to communicate with the configurator service.
aggregator.ipam.workflow	The name of the workflow in Cyber Controller/Vision used to manage IPAM.
aggregator.ipam.pool	The name of the IP pool configured for IPAM.
aggregator.ipam.ip	The IP/hostname of the Cyber Controller/Vision device where the IPAM workflow resides.
aggregator.ipam.user	The user the configurator uses to log in to the Cyber Controller/Vision device used to manage IPAM.
aggregator.ipam.password	The password the configurator uses to log in to the Cyber Controller/Vision device used to manage IPAM. **Important!** If the password is changed for this user, it requires to update the configurator deployment using the helm upgrade -install command with the user and password options.
aggregator.ipam.port	The port on which Cyber Controller/Vision listens for aggregator messages to the IPAM workflow.
aggregator.ipam.protocol	The protocol (HTTP or HTTPS) the aggregator uses to communicate with Cyber Controller
aggregator.args.verbose	Set the verbosity level. Values: 1 to 3 (1 for least verbose)
aggregator.max_nodes_per_cluster	Set the node limitation. The maximum nodes (reals per cluster) that will be configured in Alteon. Values: >= 1 (int), 0 (unlimited) **Note:** Supported in versions 1.5.0 and up.
Configurator Options	
configurator.enable	Specifies whether to install configurator. Values are true and false.
configurator.image.repository	The name of the docker image.

Option Name	Description
configurator.image.pullPolicy	The imagePullPolicy for a container and the tag of the image affect when the kubelet attempts to pull (download) the specified image.
configurator.loglevel	Set the verbosity level ("INFO" or "DEBUG")
configurator.image.tag	The tag of the docker image (e.g., latest).
configurator.alteon.master.ip	The IP or host name of the primary Alteon device.
configurator.alteon.backup_ip	The IP or host name of the secondary (backup) Alteon device.
configurator.alteon.adcSecretCredential	The name of the user-defined secret that contains Alteon device credentials (username and password). See how to create the secret at **Alteon Credentials Secret** below. **Note:** Supported in versions 1.5.0 and up.
configurator.alteon.user	The user the configurator uses to log in to the Alteon device. **Note:** To specify the Alteon device credentials you can either use secret (configurator.adcSecredCredential) or configurator.alteon.user and configurator.alteon.password parameters
configurator.alteon.pass	The password the configurator uses to log in to the Alteon device. Important! If the password is changed for this user, it requires to update the configurator deployment using the helm upgrade -install command with the user and password options. Note: When using HA, the credentials for the backup ADC should be identical to the master ADC.
configurator.alteon.pip	Specifies the IP address/subnet that should be used by Alteon to perform Client NAT on traffic towards the real servers (K8 nodes).
configurator.alteon.pip_mask	Specifies the mask for the PIP address/subnet
Controller Options	
controller.enable	Specifies whether to install controller. Values are true and false.
controller.image.repository	The name of the docker image.
controller.image.pullPolicy	The imagePullPolicy for a container and the tag of the image affect when the kubelet attempts to pull (download) the specified image.
controller.image.tag	The tag of the docker image (e.g., latest).
controller.uid	The identifier the controller will use when communicating with the aggregator.
controller.aggregator_ip	The IP or host name of the aggregator. When running on the same cluster as the aggregator, use aggregator's default service name - "akc-aggregator.[namespace].svc.cluster.local".

Option Name	Description
controller.aggregator_port	The port of the aggregator. This parameter is only needed when the AKC controller and the AKC aggregator are on different clusters. In case a hostname is specified for aggregator_ip the port range must be in the range 30000-32767. Default value: 30051.
controller.controller_reconcile_period	Controller reconciliation period. Default 3600 seconds (1 hour).
controller.webhook.enabled	Enables/disables service parameters validation using admission control webhook. Default: true.
controller.direct_access	Enables/disables the direct pod access. When enabled, the real servers configured are pods (with their configured target port) and not nodes (node port). **Note:** Supported in versions 1.5.0 and up.
controller.secret.crt	User-defined SSL certificate used by webhook in base64 encoded format.
controller.secret.key	User defined SSL key used by webhook in base64 encoded format.
controller.args.verbose	Set the verbosity level. Values: 1 to 3 (1 for least verbose)
controller.grpc_conn.is_grpc_encrypted	Defines whether the GRPC connection between controller and aggregator is encrypted. Default: false.

Alteon Credentials Secret

This section describes how to create an Alteon credentials secret. The Alteon credentials secret is the user-defined secret that contains Alteon device credentials (username and password).

To create a secret for Alteon credentials

1. Create two environment variables for username and password:

```
U=$(echo -n "username" | base64)
P=$(echo -n "password" | base64)
```

2. Create the secret where the data section contains two fields with required names: 'u' - <username> and 'p'-<password>

```
echo "
apiVersion: v1
kind: Secret
metadata:
```

```
  name: customer-adc-secret-name
data:
  u: ${U}
  p: ${P}
" | kubectl apply --filename=- --namespace="configurator name space"
```

APPENDIX B – ALLOCATING EXTERNAL IPS IN OPENSHIFT SETUP

In order to allocated external IPs in OpenShift setup for any Alteon load balancing service, for each OpenShift cluster type the **oc edit network cluster** under Spec.externalIP instead of it being set to an empty value of {}.

```
spec:
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  externalIP: {}
  networkType: OpenShiftSDN
  serviceNetwork:
  - 172.30.0.0/16
```

Add the following (for example in case the needed external IP subnet is 10.175.130.0):

```
spec:
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  externalIP:
    policy:
networkType: OpenShiftSDN
  serviceNetwork:
  - 172.30.0.0/16
  - 10.175.130.0/24
```

RADWARE LTD. END USER LICENSE AGREEMENT

By accepting this End User License Agreement (this "License Agreement") you agree to be contacted by Radware Ltd.'s ("Radware") sales personnel.

If you would like to receive license rights different from the rights granted below or if you wish to acquire warranty or support services beyond the scope provided herein (if any), please contact Radware's sales team.

THIS LICENSE AGREEMENT GOVERNS YOUR USE OF ANY SOFTWARE DEVELOPED AND/OR DISTRIBUTED BY RADWARE AND ANY UPGRADES, MODIFIED VERSIONS, UPDATES, ADDITIONS, AND COPIES OF THE SOFTWARE FURNISHED TO YOU DURING THE TERM OF THE LICENSE GRANTED HEREIN (THE "SOFTWARE"). THIS LICENSE AGREEMENT APPLIES REGARDLESS OF WHETHER THE SOFTWARE IS DELIVERED TO YOU AS AN EMBEDDED COMPONENT OF A RADWARE PRODUCT ("PRODUCT"), OR WHETHER IT IS DELIVERED AS A STANDALONE SOFTWARE PRODUCT. FOR THE AVOIDANCE OF DOUBT IT IS HEREBY CLARIFIED THAT THIS LICENSE AGREEMENT APPLIES TO PLUG-INS, CONNECTORS, EXTENSIONS AND SIMILAR SOFTWARE COMPONENTS DEVELOPED BY RADWARE THAT CONNECT OR INTEGRATE A RADWARE PRODUCT WITH THE PRODUCT OF A THIRD PARTY (COLLECTIVELY, "CONNECTORS") FOR PROVISIONING, DECOMMISSIONING, MANAGING, CONFIGURING OR MONITORING RADWARE PRODUCTS. THE APPLICABILITY OF THIS LICENSE AGREEMENT TO CONNECTORS IS REGARDLESS OF WHETHER SUCH CONNECTORS ARE DISTRIBUTED TO YOU BY RADWARE OR BY A THIRD PARTY PRODUCT VENDOR. IN CASE A CONNECTOR IS DISTRIBUTED TO YOU BY A THIRD PARTY PRODUCT VENDOR PURSUANT TO THE TERMS OF AN AGREEMENT BETWEEN YOU AND THE THIRD PARTY PRODUCT VENDOR, THEN, AS BETWEEN RADWARE AND YOURSELF, TO THE EXTENT THERE IS ANY DISCREPANCY OR INCONSISTENCY BETWEEN THE TERMS OF THIS LICENSE AGREEMENT AND THE TERMS OF THE AGREEMENT BETWEEN YOU AND THE THIRD PARTY PRODUCT VENDOR, THE TERMS OF THIS LICENSE AGREEMENT WILL GOVERN AND PREVAIL. PLEASE READ THE TERMS AND CONDITIONS OF THIS LICENSE AGREEMENT CAREFULLY BEFORE OPENING THE PACKAGE CONTAINING RADWARE'S PRODUCT, OR BEFORE DOWNLOADING, INSTALLING, COPYING OR OTHERWISE USING RADWARE'S STANDALONE SOFTWARE (AS APPLICABLE). THE SOFTWARE IS LICENSED (NOT SOLD). BY OPENING THE PACKAGE CONTAINING RADWARE'S PRODUCT, OR BY DOWNLOADING, INSTALLING, COPYING OR USING THE SOFTWARE (AS APPLICABLE), YOU CONFIRM THAT YOU HAVE READ AND UNDERSTAND THIS LICENSE AGREEMENT AND YOU AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE AGREEMENT. FURTHERMORE, YOU HEREBY WAIVE ANY CLAIM OR RIGHT THAT YOU MAY HAVE TO ASSERT THAT YOUR ACCEPTANCE AS STATED HEREINABOVE IS NOT THE EQUIVALENT OF, OR DEEMED AS, A VALID SIGNATURE TO THIS LICENSE AGREEMENT. IF YOU ARE NOT WILLING TO BE BOUND BY THE TERMS OF THIS LICENSE AGREEMENT, YOU SHOULD PROMPTLY RETURN THE UNOPENED PRODUCT PACKAGE OR YOU SHOULD NOT DOWNLOAD, INSTALL, COPY OR OTHERWISE USE THE SOFTWARE (AS APPLICABLE). THIS LICENSE AGREEMENT REPRESENTS THE ENTIRE AGREEMENT CONCERNING THE SOFTWARE BETWEEN YOU AND RADWARE, AND SUPERSEDES ANY AND ALL PRIOR PROPOSALS, REPRESENTATIONS, OR UNDERSTANDINGS BETWEEN THE PARTIES. "YOU" MEANS THE NATURAL PERSON OR THE ENTITY THAT IS AGREEING TO BE BOUND BY THIS LICENSE AGREEMENT, THEIR EMPLOYEES AND THIRD PARTY CONTRACTORS. YOU SHALL BE LIABLE FOR ANY FAILURE BY SUCH EMPLOYEES AND THIRD PARTY CONTRACTORS TO COMPLY WITH THE TERMS OF THIS LICENSE AGREEMENT.

1. **License Grant**. Subject to the terms of this Agreement, Radware hereby grants to you, and you accept, a limited, nonexclusive, nontransferable license to install and use the Software in machine-readable, object code form only and solely for your internal business purposes ("Commercial License"). If the Software is distributed to you with a software development kit (the "SDK"), then, solely with regard to the SDK, the Commercial License above also includes a limited, nonexclusive, nontransferable license to install and use the SDK solely on computers within your organization, and solely for your internal development of an integration or interoperation of the Software and/or other Radware Products with software or hardware

products owned, licensed and/or controlled by you (the "SDK Purpose"). To the extent an SDK is distributed to you together with code samples in source code format (the "Code Samples") that are meant to illustrate and teach you how to configure, monitor and/or control the Software and/or any other Radware Products, the Commercial License above further includes a limited, nonexclusive, nontransferable license to copy and modify the Code Samples and create derivative works based thereon solely for the SDK Purpose and solely on computers within your organization. The SDK shall be considered part of the term "Software" for all purposes of this License Agreement. You agree that you will not sell, assign, license, sublicense, transfer, pledge, lease, rent or share your rights under this License Agreement nor will you distribute copies of the Software or any parts thereof. Rights not specifically granted herein, are specifically prohibited.

2. **Evaluation Use**. Notwithstanding anything to the contrary in this License Agreement, if the Software is provided to you for evaluation purposes, as indicated in your purchase order or sales receipt, on the website from which you download the Software, as inferred from any time-limited evaluation license keys that you are provided with to activate the Software, or otherwise, then You may use the Software only for internal evaluation purposes ("Evaluation Use") for a maximum of 30 days or such other duration as may specified by Radware in writing at its sole discretion (the "Evaluation Period"). The evaluation copy of the Software contains a feature that will automatically disable it after expiration of the Evaluation Period. You agree not to disable, destroy, or remove this feature of the Software, and any attempt to do so will be a material breach of this License Agreement. During or at the end of the evaluation period, you may contact Radware sales team to purchase a Commercial License to continue using the Software pursuant to the terms of this License Agreement. If you elect not to purchase a Commercial License, you agree to stop using the Software and to delete the evaluation copy received hereunder from all computers under your possession or control at the end of the Evaluation Period. In any event, your continued use of the Software beyond the Evaluation Period (if possible) shall be deemed your acceptance of a Commercial License to the Software pursuant to the terms of this License Agreement, and you agree to pay Radware any amounts due for any applicable license fees at Radware's then-current list prices.
3. **Lab License**. Notwithstanding anything to the contrary in this License Agreement, if the Software is provided to you for use in your lab or for development purposes, as indicated in your purchase order, sales receipt, the part number description for the Software, the webpage from which you download the Software, or otherwise, then You may use the Software only for internal testing and development purposes in your lab but not for any production use purposes.
4. **Subscription Software**. If you licensed the Software on a subscription basis, your rights to use the Software are limited to the subscription period. You have the option to extend your subscription. If you extend your subscription, you may continue using the Software until the end of your extended subscription period. If you do not extend your subscription, after the expiration of your subscription, you are legally obligated to discontinue your use of the Software and completely remove the Software from your system.
5. **Feedback**. Any feedback concerning the Software including, without limitation, identifying potential errors and improvements, recommended changes or suggestions ("Feedback"), provided by you to Radware will be owned exclusively by Radware and considered Radware's confidential information. By providing Feedback to Radware, you hereby assign to Radware all of your right, title and interest in any such Feedback, including all intellectual property rights therein. With regard to any rights in such Feedback that cannot, under applicable law, be assigned to Radware, you hereby irrevocably waives such rights in favor of Radware and grants Radware under such rights in the Feedback, a worldwide, perpetual royalty-free, irrevocable, sub-licensable and non-exclusive license, to use, reproduce, disclose, sublicense, modify, make, have made, distribute, sell, offer for sale, display, perform, create derivative works of and otherwise exploit the Feedback without restriction. The provisions of this Section 5 will survive the termination or expiration of this Agreement.
6. **Limitations on Use**. You agree that you will not: (a) copy, modify, translate, adapt or create any derivative works based on the Software; or (b) sublicense or transfer the Software, or include the Software or any portion thereof in any product; or (b) reverse assemble, disassemble, decompile, reverse engineer or otherwise attempt to derive source code (or the underlying ideas, algorithms, structure or organization) from the Software, in whole or in part,

except and only to the extent: (i) applicable law expressly permits any such action, despite this limitation, in which case you agree to provide Radware at least ninety (90) days advance written notice of your belief that such action is warranted and permitted and to provide Radware with an opportunity to evaluate if the law's requirements necessitate such action, or (ii) required to debug changes to any third party LGPL-libraries linked to by the Software; or (c) create, develop, license, install, use, or deploy any software or services to circumvent, enable, modify or provide access, permissions or rights which violate the technical restrictions of the Software; (d) in the event the Software is provided as an embedded or bundled component of another Radware Product, you shall not use the Software other than as part of the combined Product and for the purposes for which the combined Product is intended; (e) remove any copyright notices, identification or any other proprietary notices from the Software (including any notices of Third Party Software (as defined below); or (f) copy the Software onto any public or distributed network or use the Software to operate in or as a time-sharing, outsourcing, service bureau, application service provider, or managed service provider environment. Notwithstanding Section 5(d), if you provide hosting or cloud computing services to your customers, you are entitled to use and include the Software in your IT infrastructure on which you provide your services. It is hereby clarified that the prohibitions on modifying, or creating derivative works based on, any Software provided by Radware, apply whether the Software is provided in a machine or in a human readable form. Human readable Software to which this prohibition applies includes (without limitation) to "Radware AppShape++ Script Files" that contain "Special License Terms". It is acknowledged that examples provided in a human readable form may be modified by a user.

7. **Intellectual Property Rights**. You acknowledge and agree that this License Agreement does not convey to you any interest in the Software except for the limited right to use the Software, and that all right, title, and interest in and to the Software, including any and all associated intellectual property rights, are and shall remain with Radware or its third party licensors. You further acknowledge and agree that the Software is a proprietary product of Radware and/or its licensors and is protected under applicable copyright law.
8. **No Warranty**. The Software, and any and all accompanying software, files, libraries, data and materials, are distributed and provided "AS IS" by Radware or by its third party licensors (as applicable) and with no warranty of any kind, whether express or implied, including, without limitation, any non-infringement warranty or warranty of merchantability or fitness for a particular purpose. Neither Radware nor any of its affiliates or licensors warrants, guarantees, or makes any representation regarding the title in the Software, the use of, or the results of the use of the Software. Neither Radware nor any of its affiliates or licensors warrants that the operation of the Software will be uninterrupted or error-free, or that the use of any passwords, license keys and/or encryption features will be effective in preventing the unintentional disclosure of information contained in any file. You acknowledge that good data processing procedure dictates that any program, including the Software, must be thoroughly tested with non-critical data before there is any reliance on it, and you hereby assume the entire risk of all use of the copies of the Software covered by this License. Radware does not make any representation or warranty, nor does Radware assume any responsibility or liability or provide any license or technical maintenance and support for any operating systems, databases, migration tools or any other software component provided by a third party supplier and with which the Software is meant to interoperate.

   This disclaimer of warranty constitutes an essential and material part of this License.

   In the event that, notwithstanding the disclaimer of warranty above, Radware is held liable under any warranty provision, Radware shall be released from all such obligations in the event that the Software shall have been subject to misuse, neglect, accident or improper installation, or if repairs or modifications were made by persons other than by Radware's authorized service personnel.
9. **Limitation of Liability**. Except to the extent expressly prohibited by applicable statutes, in no event shall Radware, or its principals, shareholders, officers, employees, affiliates, licensors, contractors, subsidiaries, or parent organizations (together, the "Radware Parties"), be liable for any direct, indirect, incidental, consequential, special, or punitive damages whatsoever relating to the use of, or the inability to use, the Software, or to your relationship with, Radware or any of the Radware Parties (including, without limitation, loss or disclosure of data or information,

and/or loss of profit, revenue, business opportunity or business advantage, and/or business interruption), whether based upon a claim or action of contract, warranty, negligence, strict liability, contribution, indemnity, or any other legal theory or cause of action, even if advised of the possibility of such damages. If any Radware Party is found to be liable to You or to any third-party under any applicable law despite the explicit disclaimers and limitations under these terms, then any liability of such Radware Party, will be limited exclusively to refund of any license or registration or subscription fees paid by you to Radware.

10. **Third Party Software**. The Software includes software portions developed and owned by third parties (the "Third Party Software"). Third Party Software shall be deemed part of the Software for all intents and purposes of this License Agreement; provided, however, that in the event that a Third Party Software is a software for which the source code is made available under an open source software license agreement, then, to the extent there is any discrepancy or inconsistency between the terms of this License Agreement and the terms of any such open source license agreement (including, for example, license rights in the open source license agreement that are broader than the license rights set forth in Section 1 above and/or no limitation in the open source license agreement on the actions set forth in Section 6 above), the terms of any such open source license agreement will govern and prevail. The terms of open source license agreements and copyright notices under which Third Party Software is being licensed to Radware or a link thereto, are included with the Software documentation or in the header or readme files of the Software. Third Party licensors and suppliers retain all right, title and interest in and to the Third Party Software and all copies thereof, including all copyright and other intellectual property associated therewith. In addition to the use limitations applicable to Third Party Software pursuant to Section 6 above, you agree and undertake not to use the Third Party Software as a general SQL server, as a stand-alone application or with applications other than the Software under this License Agreement.
11. **Term and Termination**. This License Agreement is effective upon the first to occur of your opening the package of the Product, purchasing, downloading, installing, copying or using the Software or any portion thereof, and shall continue until terminated. However, sections 5-15 shall survive any termination of this License Agreement. The Licenses granted under this License Agreement are not transferable and will terminate upon: (i) termination of this License Agreement, or (ii) transfer of the Software, or (iii) in the event the Software is provided as an embedded or bundled component of another Radware Product, when the Software is un-bundled from such Product or otherwise used other than as part of such Product. If the Software is licensed on subscription basis, this Agreement will automatically terminate upon the termination of your subscription period if it is not extended.
12. **Export**. The Software or any part thereof may be subject to export or import controls under applicable export/import control laws and regulations including such laws and regulations of the United States and/or Israel. You agree to comply with such laws and regulations, and, agree not to knowingly export, re-export, import or re-import, or transfer products without first obtaining all required Government authorizations or licenses therefor. Furthermore, You hereby covenant and agree to ensure that your use of the Software is in compliance with all other foreign, federal, state, and local laws and regulations, including without limitation all laws and regulations relating to privacy rights, and data protection. You shall have in place a privacy policy and obtain all of the permissions, authorizations and consents required by applicable law for use of cookies and processing of users' data (including without limitation pursuant to Directives 95/46/EC, 2002/58/EC and 2009/136/EC of the EU if applicable) for the purpose of provision of any services.
13. **US Government**. To the extent you are the U.S. government or any agency or instrumentality thereof, you acknowledge and agree that the Software is a "commercial computer software" and "commercial computer software documentation" pursuant to applicable regulations and your use of the is subject to the terms of this License Agreement.
14. **Governing Law**. This License Agreement shall be construed and governed in accordance with the laws of the State of Israel.
15. **Miscellaneous**. If a judicial determination is made that any of the provisions contained in this License Agreement is unreasonable, illegal or otherwise unenforceable, such provision or provisions shall be rendered void or invalid only to the extent that such judicial determination finds such provisions to be unreasonable, illegal or otherwise unenforceable, and the remainder

of this License Agreement shall remain operative and in full force and effect. In any event a party breaches or threatens to commit a breach of this License Agreement, the other party will, in addition to any other remedies available to, be entitled to injunction relief. This License Agreement constitutes the entire agreement between the parties hereto and supersedes all prior agreements between the parties hereto with respect to the subject matter hereof. The failure of any party hereto to require the performance of any provisions of this License Agreement shall in no manner affect the right to enforce the same. No waiver by any party hereto of any provisions or of any breach of any provisions of this License Agreement shall be deemed or construed either as a further or continuing waiver of any such provisions or breach waiver or as a waiver of any other provision or breach of any other provision of this License Agreement.

IF YOU DO NOT AGREE WITH THE TERMS OF THIS LICENSE YOU MUST REMOVE THE SOFTWARE FROM ANY DEVICE OWNED BY YOU AND IMMIDIATELY CEASE USING THE SOFTWARE.