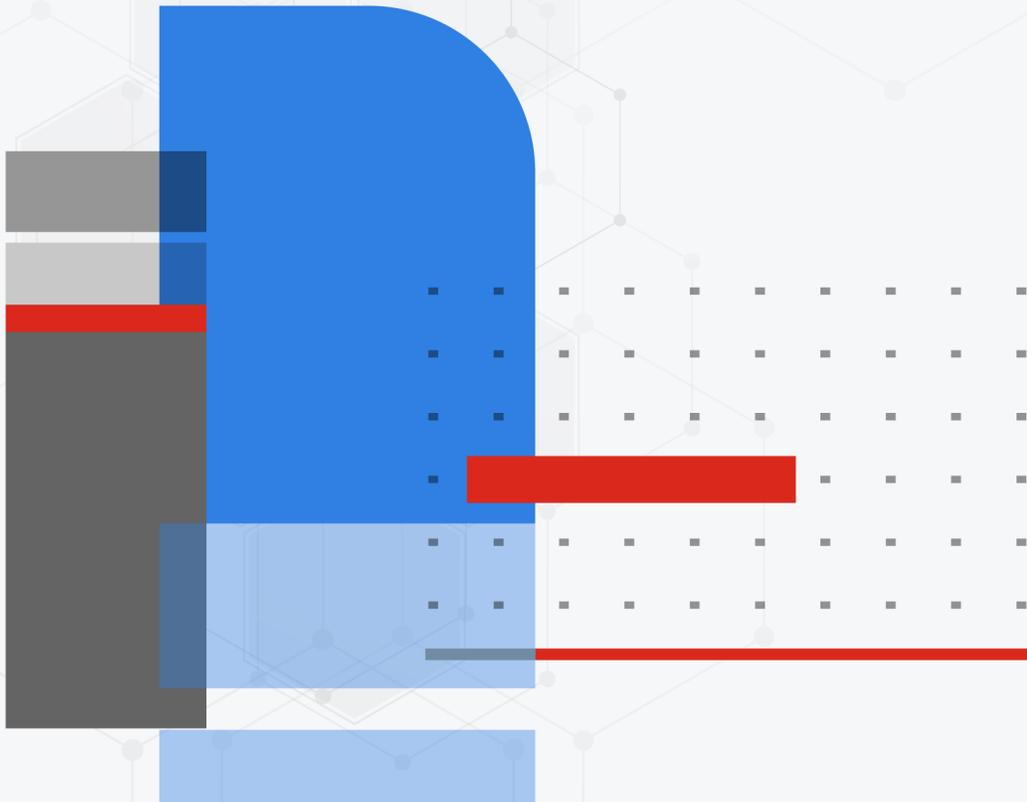




FortiADC Kubernetes Controller in OpenShift 3.1.0



FORTINET DOCUMENT LIBRARY

<https://docs.fortinet.com>

FORTINET VIDEO LIBRARY

<https://video.fortinet.com>

FORTINET BLOG

<https://blog.fortinet.com>

CUSTOMER SERVICE & SUPPORT

<https://support.fortinet.com>

FORTINET TRAINING & CERTIFICATION PROGRAM

<https://www.fortinet.com/training-certification>

FORTINET TRAINING INSTITUTE

<https://training.fortinet.com>

FORTIGUARD LABS

<https://www.fortiguard.com>

END USER LICENSE AGREEMENT

<https://www.fortinet.com/doc/legal/EULA.pdf>

FEEDBACK

Email: techdoc@fortinet.com



December 31, 2025

FortiADC Kubernetes Controller in OpenShift 3.1.0

TABLE OF CONTENTS

| | |
|---|-----------|
| Change Log | 4 |
| About FortiADC Kubernetes Controller in OpenShift | 5 |
| Architecture and Concepts | 6 |
| Architecture Overview | 6 |
| High-Level Workflow | 6 |
| Key Components | 6 |
| Resource Mapping | 7 |
| Static Route and VxLAN Models | 7 |
| Prerequisite Knowledge | 9 |
| Supported Environments | 10 |
| Deploying FortiADC Kubernetes Controller in OpenShift | 11 |
| OpenShift Routes | 11 |
| Mapping of the Routes related resources with the FortiADC objects | 12 |
| Supported Routes types | 12 |
| Installation | 15 |
| Installation in VxLAN Mode | 16 |
| Installation in Static Route Mode | 22 |
| Deployment | 28 |
| Deployment in VxLAN Mode | 29 |
| Deployment in Static Route Mode | 34 |
| Configuration parameters | 38 |
| Debug | 42 |

Change Log

| Date | Change Description |
|-------------|--------------------|
| December 31 | Initial release |

About FortiADC Kubernetes Controller in OpenShift

The **FortiADC Kubernetes Controller in OpenShift** extends the functionality of the FortiADC Kubernetes Controller to operate within **Red Hat OpenShift** environments.

It allows FortiADC to synchronize configuration objects such as virtual servers, real server pools, and SSL profiles with OpenShift cluster resources through API-based integration.

In OpenShift deployments, the controller interacts with the OpenShift API to monitor resources including **Route**, **Service**, **Pod**, and **Secret**. It translates these resources into FortiADC configuration updates through REST API calls, ensuring that applications exposed through OpenShift *Routes* or Fortinet's custom resources such as *VirtualServer*, *RemoteServer*, and *Host* are delivered securely and efficiently.

Key capabilities include:

- **Route-based resource mapping** – Uses OpenShift Route resources to manage external access to application services, mapping OpenShift routing definitions to FortiADC virtual servers and real server pools.
- **Support for VirtualServer CRDs** – Enables advanced Layer 7 and Layer 4 configuration of FortiADC features directly within OpenShift, including traffic control and security parameters.
- **Distributed global traffic** – Deploys FortiADC's Global Load Balance to publish services and direct traffic to suitable servers across different geographical locations.
- **Integrated application security** – Applies FortiADC's Web Application Firewall (WAF), antivirus scanning, and Denial of Service (DoS) protection to OpenShift-managed applications.
- **Real-time synchronization** – Automatically updates FortiADC configurations when OpenShift resources are created, modified, or removed.

This guide focuses on deploying and configuring the FortiADC Kubernetes Controller in OpenShift, including installation requirements, supported resources, configuration parameters, and deployment examples using Route objects.



For more information on Fortinet's custom resources such as *VirtualServer*, *RemoteServer*, and *Host*, including definitions and deployment examples, refer to [FortiADC Kubernetes Controller Deployment Guide](#).

Architecture and Concepts

The FortiADC Kubernetes Controller in OpenShift operates as an intermediary between OpenShift cluster resources and FortiADC configuration objects.

It runs as a controller Pod that subscribes to OpenShift API events, interprets *Route*, *VirtualServer*, *RemoteServer*, and *Host* resource definitions, and reconciles the corresponding FortiADC configurations through REST API transactions.

Architecture Overview

The controller is deployed as a containerized application within an OpenShift project namespace.

It establishes a persistent connection to the OpenShift API server using service account permissions to watch resource events in real time. When a resource change is detected, the controller evaluates the new cluster state and determines whether a configuration update is required on FortiADC.

Configuration synchronization is performed through authenticated REST API calls to FortiADC. Administrative credentials are stored in a Kubernetes Secret referenced by the controller.

Each update is executed as a discrete transaction that ensures configuration integrity and prevents drift, allowing consistent operation even in large-scale or dynamic OpenShift environments.

High-Level Workflow

1. **Watch and detect** – Subscribes to the OpenShift API to monitor Add, Update, and Delete events for supported resources such as Routes, Services, Pods, and Fortinet’s custom resources.
2. **Translate** – Converts each detected resource definition into FortiADC configuration data, including virtual servers, GLB service, content-routing rules, and real-server pools.
3. **Apply** – Sends the translated configuration to FortiADC through authenticated REST API calls, creating or updating the corresponding configuration objects.
4. **Reconcile** – Continuously compares OpenShift resource definitions with FortiADC’s active configuration and updates FortiADC whenever differences are detected.

This architecture allows FortiADC to operate as a dynamic, state-aware load balancer and application security gateway that automatically reflects changes in OpenShift application topology.

Key Components

| Component | Description |
|----------------|---|
| Controller Pod | Runs the FortiADC Kubernetes Controller within the OpenShift cluster and manages communication with both the OpenShift API server and FortiADC. |

| Component | Description |
|---|---|
| FortiADC Instance | The external FortiADC device that receives REST API calls from the controller and applies the resulting configuration. |
| Service Account and RBAC | Provides the controller with permissions to read and watch OpenShift objects such as Route, Service, Pod, Secret, and VirtualServer. |
| Secret | Stores FortiADC login credentials securely within the OpenShift cluster. |
| Helm Chart | Simplifies installation, upgrade, and removal of the controller and its supporting OpenShift resources. |
| Custom Resource Definition (CRD) | Extends OpenShift with Fortinet's VirtualServer resource type for defining advanced Layer 7 and layer 4 load-balancing and security parameters. RemoteServer and Host resources are included to cover GLB-related services. |

Resource Mapping

The following table summarizes how OpenShift resources correspond to FortiADC objects.

| OpenShift Resource | FortiADC Object |
|--------------------------|----------------------------------|
| Route | Virtual Server / Content Routing |
| Service | Real Server Pool |
| Endpoint / EndpointSlice | Real Server |



For more information on Fortinet's custom resources such as *VirtualServer*, *RemoteServer*, and *Host*, including definitions and deployment examples, refer to [FortiADC Kubernetes Controller Deployment Guide](#).

Static Route and VxLAN Models

The FortiADC Kubernetes Controller in OpenShift supports two network integration models for connecting FortiADC to OpenShift cluster endpoints.

- **Static Route model**

FortiADC reaches OpenShift Pods or services through statically defined routes. This model requires manual route configuration on FortiADC and is typically used in environments without an overlay network. It provides predictable network paths and is suitable for deployments where network topology is fixed.

- **VxLAN model**

FortiADC connects to Pods through the OpenShift VxLAN overlay network. In this model, FortiADC participates in the VxLAN fabric, allowing dynamic discovery and direct Layer 2 connectivity with Pods across nodes. This simplifies routing and improves scalability in distributed OpenShift clusters.

Both models are supported through the same controller synchronization mechanism. The controller automatically translates OpenShift resource events into FortiADC configuration updates, regardless of the underlying connectivity model.



Before deploying the FortiADC Kubernetes Controller in OpenShift, review the OpenShift, Kubernetes, and Helm concepts described in [Prerequisite Knowledge on page 9](#).

Prerequisite Knowledge

OpenShift

Before using the Kubernetes Controller on the OpenShift Container Platform, you should have a basic understanding of OpenShift clusters and Route configuration. For more information, refer to the documents below:

- OpenShift Container Platform Concepts: https://docs.redhat.com/en/documentation/openshift_container_platform/latest/html/getting_started/openshift-overview
- OpenShift Container Platform network plugin - OVN-Kubernetes: https://docs.redhat.com/en/documentation/openshift_container_platform/
- OpenShift Routes: https://docs.redhat.com/en/documentation/openshift_container_platform/4.12/html/networking/configuring-routes#route-configuration

Helm Charts

As Helm Charts are used in FortiADC Kubernetes Controller installation, you will also need to have understanding of how Helm Charts work. For more information, please refer to the documents listed below:

- Helm Charts values files: https://helm.sh/docs/chart_template_guide/values_files/
- Helm Charts Installation and upgrade from the Helm repository: https://helm.sh/docs/helm/helm_install/

Kubernetes

Prerequisite knowledge of the Kubernetes cluster will help you understand the terms and concepts discussed in this document. For more information, please refer to the documents listed below:

- Kubernetes Concepts: <https://kubernetes.io/docs/concepts/>
- Kubernetes Ingress: <https://kubernetes.io/docs/concepts/services-networking/ingress/>
- Kubernetes Service: <https://kubernetes.io/docs/concepts/services-networking/service/>

Cert-Manager

FortiADC Kubernetes Controller 3.1, introduces **cert-manager** to automate the issuance, renewal, and management of TLS certificates for the kubernetes webhook server. For more usage information, see [cert-manager Documentation](#).

Supported Environments

Supported Release and Version

| Product | Version | | |
|--------------------------------|----------------|---------------------------------|---------------|
| FortiADC Kubernetes Controller | 2.0.0 | 2.0.2 | 3.0.0 - 3.1.0 |
| OpenShift Container Platform | 4.7 - 4.12.x | 4.7 - 4.12.x 4.13 - 4.15.x** | 4.13 - 4.19 |
| FortiADC Version | 7.4.0 - 8.x.x* | | |

*Some features from FortiADC Kubernetes Controller version 2.0.0 or later require FortiADC version 7.4.0 or later to support.

**FortiADC Kubernetes Controller version 2.0.2 introduces support for Static Route Mode, which is supported in the OpenShift Container Platform versions 4.13 through 4.19.

Deploying FortiADC Kubernetes Controller in OpenShift

The **OpenShift Container Platform (version 4.x)** supports the FortiADC Kubernetes Controller for both **Kubernetes Ingress** and **OpenShift Route** objects.

When the service type is **NodePort**, the controller operates in the same way as it does in a standard Kubernetes environment, managing Ingress resources and synchronizing FortiADC configurations accordingly.

When the service type is **ClusterIP**, the controller monitors **OpenShift Route** objects and automatically applies the corresponding configuration updates on FortiADC.

OpenShift Routes

An OpenShift **Route** exposes an application at a public URL through a **ClusterIP** service.

The URL can be secured or unsecured depending on the TLS configuration defined in the Route specification.

The FortiADC Kubernetes Controller monitors Route creation, modification, and deletion events and translates these events into load-balancer configurations on FortiADC.

FortiADC Kubernetes Controller supports OpenShift Routes in two operational modes:

- **VxLAN model** - Supported in OpenShift versions **4.7 through 4.12**.
Uses Virtual Extensible LAN (VxLAN) encapsulation for overlay networking and dynamic Pod connectivity.
- **Static Route model** - Supported in OpenShift versions **4.13 through 4.19**.
Uses predefined static routes on FortiADC to reach Pod networks without overlay encapsulation.

Kubernetes Ingress can also be deployed in the OpenShift environment via FortiADC Kubernetes Controller. Please refer to FortiADC Kubernetes Controller document for more details.



There are some limitations to what FortiADC Kubernetes Controller supports for OpenShift Routes:

- Hostname wildcard is not supported with OpenShift Routes since OpenShift Routes is exposing applications with a URL.
 - Only the ClusterIP service type is supported for OpenShift Routes.
 - Services with multiple ports exposed is not supported.
-

Mapping of the Routes related resources with the FortiADC objects

| OpenShift Objects | FortiADC Objects |
|-------------------------|--|
| Routes | Virtual server Content Routing Scripting |
| Service | Real Server Pool |
| Endpoint/ EndpointSlice | Real Server |

Supported Routes types

FortiADC supports both insecure and secured Routes.

Currently, FortiADC Kubernetes Controller only supports secured Routes that uses **edge** TLS termination with a custom certificate. The **insecureEdgeTerminationPolicy** is also not supported.

Insecure Routes with path



Due to PDF formatting limitations, the code example below would not retain indentations if copy and pasted directly into a YAML file. Without the proper indentations, the YAML will be invalid.

Please follow this link to copy and modify the route YAML example:

https://github.com/fortinet/fortiadc-kubernetes-controller/blob/main/route_examples/basic_route_with_path.yaml

```
apiVersion: route.openshift.io/v1
kind: Route
metadata:
  labels:
    name: fad-app
  name: fad-demo-app
  namespace: fadgw
  annotations: {
    "ingress-controller" : "fadc-ingress-controller",
    "fortiadc-ip" : "172.23.133.171",
    "fortiadc-login" : "fad-login",
    "fortiadc-vdom" : "root",
    "fortiadc-ctrl-log" : "enable",
    "virtual-server-ip" : "10.0.0.104",
    "virtual-server-interface" : "port2",
    "virtual-server-port" : "443",
    "load-balance-method" : "LB_METHOD_LEAST_CONNECTION",
```

```

    "load-balance-profile" : "LB_PROF_HTTPS"
  }
spec:
  host: testfad.com
  path: "/info"
  port:
    targetPort: 1234-tcp
  to:
    kind: Service
    name: fad-app

```

Insecure Routes without path



Due to PDF formatting limitations, the code example below would not retain indentations if copy and pasted directly into a YAML file. Without the proper indentations, the YAML will be invalid.

Please follow this link to copy and modify the route YAML example:

https://github.com/fortinet/fortiadc-kubernetes-controller/blob/main/route_examples/basic_route_no_path.yaml

```

apiVersion: route.openshift.io/v1
kind: Route
metadata:
  labels:
    name: fad-app
  name: fad-demo-app
  namespace: fadgw
  annotations: {
    "ingress-controller" : "fadc-ingress-controller",
    "fortiadc-ip" : "172.23.133.171",
    "fortiadc-login" : "fad-login",
    "fortiadc-vdom" : "root",
    "fortiadc-ctrl-log" : "enable",
    "virtual-server-ip" : "10.0.0.104",
    "virtual-server-interface" : "port2",
    "virtual-server-port" : "443",
    "load-balance-method" : "LB_METHOD_LEAST_CONNECTION",
    "load-balance-profile" : "LB_PROF_HTTPS"
  }
spec:
  host: testfad.com
  port:
    targetPort: 1234-tcp
  to:
    kind: Service
    name: fad-app

```

Secured Routes



Due to PDF formatting limitations, the code example below would not retain indentations if copy and pasted directly into a YAML file. Without the proper indentations, the YAML will be invalid.

Please follow this link to copy and modify the route YAML example:

https://github.com/fortinet/fortiadc-kubernetes-controller/blob/main/route_examples/secure_route_example.yaml

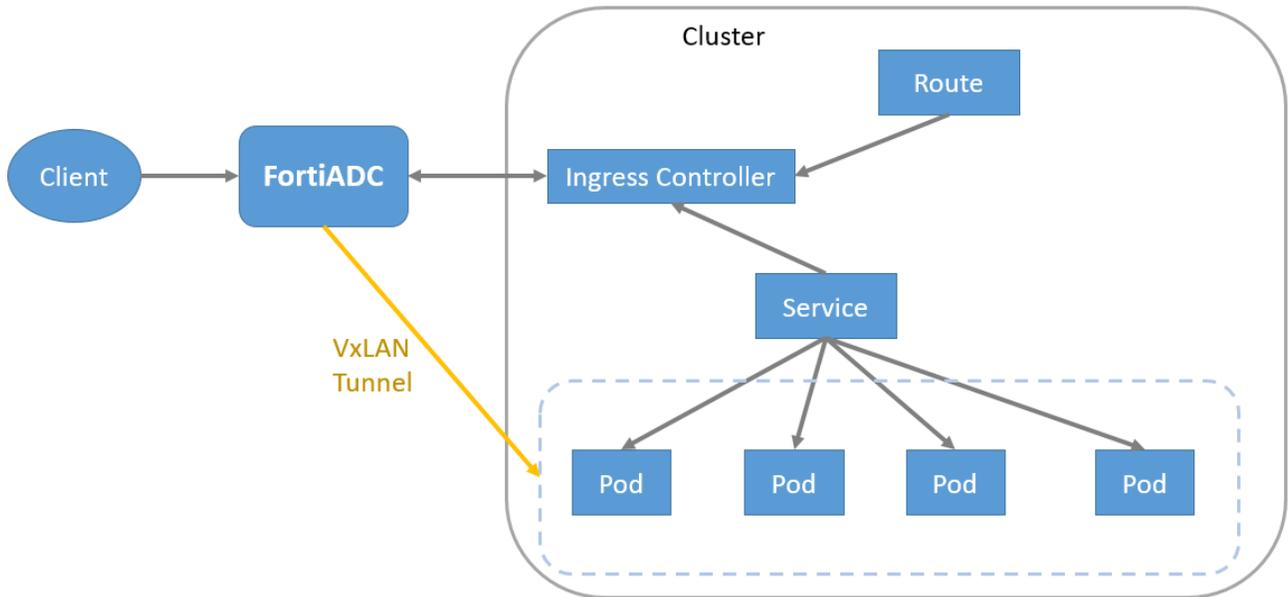
```
apiVersion: route.openshift.io/v1
kind: Route
metadata:
  annotations:
    fortiadc-ctrl-log: enable
    fortiadc-ip: 172.23.133.171
    fortiadc-login: fad-login
    fortiadc-vdom: root
    ingress-controller: fadc-ingress-controller
    load-balance-method: LB_METHOD_LEAST_CONNECTION
    load-balance-profile: LB_PROF_HTTPS
    virtual-server-interface: port1
    virtual-server-ip: 172.23.133.182
    virtual-server-port: "443"
  labels:
    app: fad-app
    name: fad-demo-app
    namespace: fadgw
spec:
  host: testtlsroute.com
  port:
    targetPort: 1234-tcp
  tls:
    certificate: |
      -----BEGIN CERTIFICATE-----
      -----END CERTIFICATE-----
    key: |
      -----BEGIN PRIVATE KEY-----
      -----END PRIVATE KEY-----
  termination: edge
to:
  kind: Service
  name: fad-app
```

Installation

The installation instructions for the FortiADC Kubernetes Controller in OpenShift are organized according to the two OpenShift Routes operational modes:

- [Installation in VxLAN Mode on page 16](#)
- [Installation in Static Route Mode on page 22](#)

Installation in VxLAN Mode



FortiADC Kubernetes Controller with OpenShift on VxLAN mode requires integration with an OpenShift Cluster utilizing the OVN-Kubernetes CNI, and it must have hybrid networking enabled.

Supported OpenShift version

The FortiADC OpenShift connector with VxLAN supports OpenShift Container Platform versions 4.7 to 4.12.x. FortiADC Kubernetes Controller needs to work with OpenShift OVN-Kubernetes CNI and hybrid networking enabled.

Installing FortiADC Kubernetes Controller in OpenShift with VxLAN

Install FortiADC Kubernetes Controller on OpenShift Container Platform 4 using Helm Charts.



Currently, only Helm 3 (version 3.6.3 or later) is supported.

Helm Charts ease the installation of FortiADC Kubernetes Controller in the OpenShift cluster. By using the Helm 3 installation tool, most of the OpenShift objects required for FortiADC Kubernetes Controller can be deployed in one simple command.

The OpenShift objects required for FortiADC Kubernetes Controller are listed below:

| OpenShift object | Description |
|----------------------|--|
| Deployment | By configuring the replica and pod template in the OpenShift deployment, the deployment ensures FortiADC Kubernetes Controller provides a non-terminated service. |
| Service Account | The service account is used in FortiADC Kubernetes Controller. |
| Cluster Role | A cluster role defines the permission on the OpenShift cluster-scoped Routes-related objects. |
| Cluster Role Binding | The cluster role is bound to the service account used for FortiADC Kubernetes Controller, allowing FortiADC Kubernetes Controller to access and operate the OpenShift cluster-scoped Routes-related objects. |
| Ingress Class | The IngressClass "fadc-ingress-controller" is created for FortiADC Kubernetes Controller to identify the Ingress resource. If the Ingress is defined with the IngressClass "fadc-ingress-controller", FortiADC Kubernetes Controller will manage this Ingress resource as FortiADC Kubernetes Controller also supports Ingress in the OpenShift cluster. |

The Helm Chart is composed of a collection of files that describe the related set of OpenShift files required by FortiADC Kubernetes Controller; one of which is the `values.yaml` file that provides the default configuration for deploying the OpenShift objects listed above.

Below lists parts of the values in the `values.yaml` file.

```
# Default values for fadc-k8s-ctrl.
# This is a YAML-formatted file.
# Declare variables to be passed into your templates.
# FortiADC Kubernetes Controller image from Dockerhub.com
image:
  repository: fortinet/fortiadc-ingress
  pullPolicy: IfNotPresent
  tag: "3.1.0"

serviceAccount:
  create: true
  annotations: {}
  name: "fortiadc-ingress"
tolerations:
  - effect: "NoExecute"
    key: "node.kubernetes.io/not-ready"
    operator: "Exists"
    tolerationSeconds: 30
  - effect: "NoExecute"
    key: "node.kubernetes.io/unreachable"
    operator: "Exists"
    tolerationSeconds: 30
# Define Ingress Class for FortiADC Kubernetes Controller
controller:
  ingressClassResource:
    name: "fadc-ingress-controller"
    enabled: true
```

```

    default: true
    controllerValue: "fortinet.com/fadc-ingress-controller"
# You can decide parameters defined in annotation of Ingress to be optional or mandatory.
# FortiADC Kubernetes Controller will check the parameter if it marks mandatory.
parameters:
  virtualServerNatSrcPool : "optional"
  virtualServerWafProfile : "optional"
  virtualServerAvProfile : "optional"
  virtualServerDosProfile : "optional"
  virtualServerCaptchaProfile : "optional"
  virtualServerPersistence : "optional"
  virtualServerFortiGSLB : "optional"
  openshiftRouteSupport: "no"
  enableStaticRouteSupport: "no"
webhook:
  useCertManager: true
  service:
    name: fad-webhook
    port: 443
    targetPort: 8443
  tlsSecretName: webhook-tls
  validatingWebhookName: validator.fadk8sctrl.fortinet.com
  mutatingWebhookName: mutator.fadk8sctrl.fortinet.com
  rules:
  validating:
    - name: validate-vs.fadk8sctrl.fortinet.com
      group: fadk8sctrl.fortinet.com
      version: v1alpha2
      resources:
        - virtualservers
      scope: "Namespaced"
      path: /validate-vs
    - name: validate-ingress.fadk8sctrl.fortinet.com
      group: networking.k8s.io
      version: v1
      resources:
        - ingresses
      scope: "Namespaced"
      path: /validate-ingress
  mutating:
    - name: mutate-vs.fadk8sctrl.fortinet.com
      group: fadk8sctrl.fortinet.com
      version: v1alpha2
      resources:
        - virtualservers
      scope: "Namespaced"
      path: /mutate-vs

```



In some scenarios, you may want to override some of the values included in the values .yaml, such as for the toleration seconds or parameter properties. As the values .yaml file is packed in the Helm Chart package, you can override the values when installing or upgrading the Helm Chart (see [Install the Helm Chart on page 19](#) and [Upgrade the Helm Chart on page 20](#)). For more details on the parameters, see [Configuration parameters on page 38](#).

Install the Helm Chart

```
curl -L https://mirror.openshift.com/pub/openshift-v4/clients/helm/latest/helm-linux-amd64 -o /usr/local/bin/helm
chmod +x /usr/local/bin/helm
helm version
```

For more details, see the OpenShift documentation on Helm Chart installation:

https://docs.openshift.com/container-platform/4.9/applications/working_with_helm_charts/installing-helm.html

Install cert-manager.io



Starting with FortiADC Kubernetes Controller version 3.1, a webhook server is introduced. A cert-manager installation is required to generate the self-signed certificate used for the TLS connection between the Kubernetes API server and the webhook server.

Before installing FortiADC Kubernetes Controller 3.1 or later, or upgrading to version 3.1 or later, install cert-manager by following the official installation guide at <https://cert-manager.io/docs/installation/>.

Compatibility has been verified with cert-manager v1.19.1.

```
helm repo add jetstack https://charts.jetstack.io
helm repo update
helm install --debug cert-manager jetstack/cert-manager \
  --namespace cert-manager \
  --create-namespace \
  --version v1.19.1 \
  --set crds.enabled=true
```

Get Repo Information

To get the repository information:

```
helm repo add fortiadc-kubernetes-controller https://fortinet.github.io/fortiadc-kubernetes-controller/
helm repo update
```

Installation parameters

To support Routes in OpenShift Container Platform, the **openshiftRouteSupport** parameter is added for installation. The default value of **openshiftRouteSupport** is **no**.

Starting from version **3.0.0**, the security level of the FortiADC Kubernetes Controller has been increased. When deploying in OpenShift environments, you must set **openshiftRouteSupport** to **yes** to ensure a secure installation and prevent installation failure.

Install and Update the Helm Chart

You can specify a particular OpenShift Project in which FortiADC Kubernetes Controller will be deployed. Note that OpenShift Project is equal to Kubernetes Namespace.

By default, if no OpenShift Project is specified, the default project would be "default". The `RELEASE_NAME` is the name you give to this chart installation.

Try with the following command to enable OpenShift Routes support in FortiADC Kubernetes Controller.

```
helm install --set parameters.openshiftRouteSupport="yes" --namespace [OpenShift Project]
[RELEASE_NAME] fortiadc-kubernetes-controller/fadc-k8s-ctrl
```

In the example below, the Helm chart is installed with the release name "first-release" in the OpenShift project "fortiadc-ingress".

```
helm install --set parameters.openshiftRouteSupport="yes" \
--namespace fortiadc-ingress first-release fortiadc-kubernetes-controller/fadc-k8s-ctrl
```

If you want to override values in the Helm Chart, you can add `--set` flags in the command. In the example below, you can set the **virtualServerWafProfile** parameter as mandatory.

```
helm install --set parameters.openshiftRouteSupport="yes" \
--set parameters.virtualServerWafProfile="mandatory" \
--namespace fortiadc-ingress first-release fortiadc-kubernetes-controller/fadc-k8s-ctrl
```

Moreover, you can create a new project and deploy FortiADC Kubernetes Controller within the project at the same time.

```
helm install --set parameters.openshiftRouteSupport="yes" --namespace fortiadc-ingress --create-
namespace first-release fortiadc-kubernetes-controller/fadc-k8s-ctrl
```

Upgrade the Helm Chart

Starting with version **3.0.0**, the Helm chart repository was renamed to **fortiadc-kubernetes-controller**.

When upgrading from version **2.x** to **3.0.0** or later, remove the old Helm repository and add the new repository before continuing with the upgrade.

```
helm repo remove fortiadc-ingress
helm repo add fortiadc-kubernetes-controller \
https://fortinet.github.io/fortiadc-kubernetes-controller/
helm repo update
```

To upgrade the Helm chart, add the required parameter.

```
helm upgrade --set parameters.openshiftRouteSupport="yes" [RELEASE_NAME] fortiadc-kubernetes-
controller/fadc-k8s-ctrl
```

In the example below, you can upgrade the helm chart with the release name "first-release".

```
helm upgrade --set parameters.openshiftRouteSupport="yes" \
first-release fortiadc-kubernetes-controller/fadc-k8s-ctrl
```

Using the `--debug` option, you can check the Helm debug information “USER-SUPPLIED VALUES” to check if you have all the value set as you need.

```
Release "first-release" has been upgraded. Happy Helming!
NAME: first-release
LAST DEPLOYED: Fri Aug 4 13:32:10 2023
NAMESPACE: fortiadc-ingress
STATUS: deployed
REVISION: 2
TEST SUITE: None
USER-SUPPLIED VALUES:
parameters:
  virtualServerWafProfile: mandatory
  openshiftRouteSupport: yes
```

Check the Installation

The `helm history` command shows the installation information.

```
[root@bastion ~]# helm history first-release
REVISION      UPDATED              STATUS      CHART
APP VERSION   DESCRIPTION
1             Fri Aug 4 13:31:20 2023    superseded  fadc-k8s-ctrl-2.0.0
2.0.0         Install complete
```

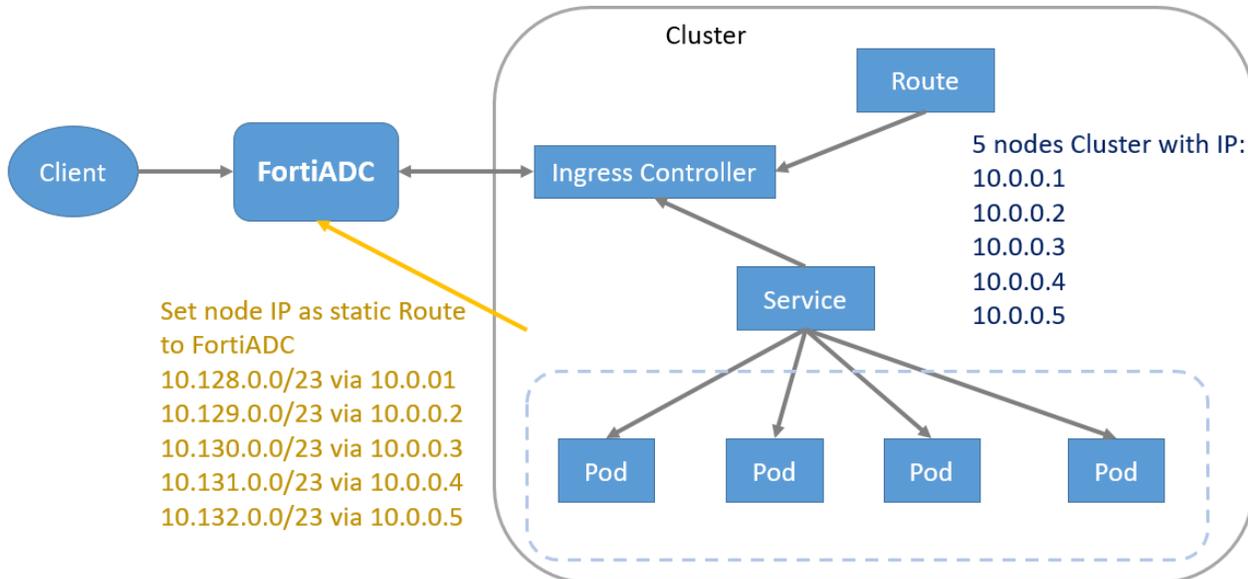
Check if FortiADC Kubernetes Controller is installed correctly.

```
[root@ocpexp openshift]# oc get deployment
NAME                                READY  UP-TO-DATE  AVAILABLE  AGE
first-release-fadc-k8s-ctrl        1/1    1            1          4d17h
```

Check the FortiADC Kubernetes Controller log.

```
[root@ocpexp openshift]# oc logs first-release-fadc-k8s-ctrl-7cdcfd6df6-fds6h
Stopping fortiadc Kubernetes controller
Starting fortiadc Kubernetes controller
time="2024-06-05T05:42:38Z" level=info msg="Starting FortiADC Kubernetes controller"
time="2024-06-05T05:42:39Z" level=info msg="Routes Monitor Enabled"
```

Installation in Static Route Mode



In Static Route mode, the FortiADC Kubernetes Controller configures static route entries for each node's pod network subnet and node IP on FortiADC when the first OpenShift Route is deployed to the OpenShift Container Platform cluster. These static routes remain in place until the last OpenShift Route is removed.

Supported OpenShift version

The FortiADC Kubernetes connector with Static Routes supports OpenShift Container Platform versions 4.13 to 4.19.x.

Installing FortiADC Kubernetes Controller in OpenShift with Static Route

Install FortiADC Kubernetes Controller on OpenShift Container Platform 4 using Helm Charts.



Currently, only Helm 3 (version 3.6.3 or later) is supported.

Helm Charts ease the installation of FortiADC Kubernetes Controller in the OpenShift cluster. By using the Helm 3 installation tool, most of the OpenShift objects required for FortiADC Kubernetes Controller can be deployed in one simple command.

The OpenShift objects required for FortiADC Kubernetes Controller are listed below:

| OpenShift object | Description |
|----------------------|--|
| Deployment | By configuring the replica and pod template in the OpenShift deployment, the deployment ensures FortiADC Kubernetes Controller provides a non-terminated service. |
| Service Account | The service account is used in FortiADC Kubernetes Controller. |
| Cluster Role | A cluster role defines the permission on the OpenShift cluster-scoped Routes-related objects. |
| Cluster Role Binding | The cluster role is bound to the service account used for FortiADC Kubernetes Controller, allowing FortiADC Kubernetes Controller to access and operate the OpenShift cluster-scoped Routes-related objects. |
| Ingress Class | The IngressClass "fadc-ingress-controller" is created for FortiADC Kubernetes Controller to identify the Ingress resource. If the Ingress is defined with the IngressClass "fadc-ingress-controller", FortiADC Kubernetes Controller will manage this Ingress resource as FortiADC Kubernetes Controller also supports Ingress in the OpenShift cluster. |

The Helm Chart is composed of a collection of files that describe the related set of OpenShift files required by FortiADC Kubernetes Controller; one of which is the `values.yaml` file that provides the default configuration for deploying the OpenShift objects listed above.

Below lists parts of the values in the `values.yaml` file.

```
# Default values for fadc-k8s-ctrl.
# This is a YAML-formatted file.
# Declare variables to be passed into your templates.
# FortiADC Kubernetes Controller image from Dockerhub.com
image:
  repository: fortinet/fortiadc-ingress
  pullPolicy: IfNotPresent
  tag: "3.1.0"

serviceAccount:
  create: true
  annotations: {}
  name: "fortiadc-ingress"
tolerations:
- effect: "NoExecute"
  key: "node.kubernetes.io/not-ready"
  operator: "Exists"
  tolerationSeconds: 30
- effect: "NoExecute"
  key: "node.kubernetes.io/unreachable"
  operator: "Exists"
  tolerationSeconds: 30
# Define Ingress Class for FortiADC Kubernetes Controller
controller:
  ingressClassResource:
    name: "fadc-ingress-controller"
    enabled: true
```

```
    default: true
    controllerValue: "fortinet.com/fadc-ingress-controller"
# You can decide parameters defined in annotation of Ingress to be optional or mandatory.
# FortiADC Kubernetes Controller will check the parameter if it marks mandatory.
parameters:
  virtualServerNatSrcPool : "optional"
  virtualServerWafProfile : "optional"
  virtualServerAvProfile : "optional"
  virtualServerDosProfile : "optional"
  virtualServerCaptchaProfile : "optional"
  virtualServerPersistence : "optional"
  virtualServerFortiGSLB : "optional"
  openshiftRouteSupport: "no"
  enableStaticRouteSupport: "no"
webhook:
  useCertManager: true
  service:
    name: fad-webhook
    port: 443
    targetPort: 8443
  tlsSecretName: webhook-tls
  validatingWebhookName: validator.fadk8sctrl.fortinet.com
  mutatingWebhookName: mutator.fadk8sctrl.fortinet.com
rules:
  validating:
    - name: validate-vs.fadk8sctrl.fortinet.com
      group: fadk8sctrl.fortinet.com
      version: v1alpha2
      resources:
        - virtualservers
      scope: "Namespaced"
      path: /validate-vs
    - name: validate-ingress.fadk8sctrl.fortinet.com
      group: networking.k8s.io
      version: v1
      resources:
        - ingresses
      scope: "Namespaced"
      path: /validate-ingress
  mutating:
    - name: mutate-vs.fadk8sctrl.fortinet.com
      group: fadk8sctrl.fortinet.com
      version: v1alpha2
      resources:
        - virtualservers
      scope: "Namespaced"
      path: /mutate-vs
```



In some scenarios, you may want to override some of the values included in the values .yaml, such as for the toleration seconds or parameter properties. As the values .yaml file is packed in the Helm Chart package, you can override the values when installing or upgrading the Helm Chart (see [Install the Helm Chart on page 25](#) and [Upgrade the Helm Chart on page 26](#)). For more details on the parameters, see [Configuration parameters on page 38](#).

Install the Helm Chart

```
curl -L https://mirror.openshift.com/pub/openshift-v4/clients/helm/latest/helm-linux-amd64 -o /usr/local/bin/helm
chmod +x /usr/local/bin/helm
helm version
```

For more details, see the OpenShift documentation on Helm Chart installation:

https://docs.openshift.com/container-platform/4.9/applications/working_with_helm_charts/installing-helm.html

Install cert-manager.io



Starting with FortiADC Kubernetes Controller version 3.1, a webhook server is introduced. A cert-manager installation is required to generate the self-signed certificate used for the TLS connection between the Kubernetes API server and the webhook server.

Before installing FortiADC Kubernetes Controller 3.1 or later, or upgrading to version 3.1 or later, install cert-manager by following the official installation guide at <https://cert-manager.io/docs/installation/>.

Compatibility has been verified with cert-manager v1.19.1.

```
helm repo add jetstack https://charts.jetstack.io
helm repo update
helm install --debug cert-manager jetstack/cert-manager \
  --namespace cert-manager \
  --create-namespace \
  --version v1.19.1 \
  --set crds.enabled=true
```

Get Repo Information

To get the repository information:

```
helm repo add fortiadc-kubernetes-controller https://fortinet.github.io/fortiadc-kubernetes-controller/
helm repo update
```

Installation parameters

To support Routes in OpenShift Container Platform, the **openshiftRouteSupport** parameter is added for installation. The default value of **openshiftRouteSupport** is **no**.

Starting from version **3.0.0**, the security level of the FortiADC Kubernetes Controller has been increased. When deploying in OpenShift environments, you must set **openshiftRouteSupport** to **yes** to ensure a secure installation and prevent installation failure.

To enable FortiADC Kubernetes Controller with static route, the **enableStaticRouteSupport** parameter is added for installation. The default value of **enableStaticRouteSupport** is **no**.

To enable static route in OpenShift Container Platform 4, both **openshiftRouteSupport** and **enableStaticRouteSupport** must be set to **yes**.

Install and Update the Helm Chart

You can specify a particular OpenShift Project in which FortiADC Kubernetes Controller will be deployed. Note that OpenShift Project is equal to Kubernetes Namespace.

By default, if no OpenShift Project is specified, the default project would be "default". The **RELEASE_NAME** is the name you give to this chart installation.

Try with the following command to enable OpenShift Routes support in FortiADC Kubernetes Controller.

```
helm install --set parameters.openshiftRouteSupport="yes" \  
--set parameters.enableStaticRouteSupport="yes" \  
--namespace [OpenShift Project] [RELEASE_NAME] fortiadc-kubernetes-controller/fadc-k8s-ctrl
```

In the example below, the Helm chart is installed with the release name "first-release" in the OpenShift project "fortiadc-ingress".

```
helm install --set parameters.openshiftRouteSupport="yes" \  
--set parameters.enableStaticRouteSupport="yes" \  
--namespace fortiadc-ingress first-release fortiadc-kubernetes-controller/fadc-k8s-ctrl
```

If you want to override values in the Helm Chart, you can add **--set** flags in the command. In the example below, you can set the **virtualServerWafProfile** parameter as mandatory.

```
helm install --set parameters.openshiftRouteSupport="yes" \  
--set parameters.enableStaticRouteSupport="yes" \  
--set parameters.virtualServerWafProfile="mandatory" \  
--namespace fortiadc-ingress first-release fortiadc-kubernetes-controller/fadc-k8s-ctrl
```

Moreover, you can create a new project and deploy FortiADC Kubernetes Controller within the project at the same time.

```
helm install --set parameters.openshiftRouteSupport="yes" \  
--set parameters.enableStaticRouteSupport="yes" \  
--namespace fortiadc-ingress --create-namespace \  
first-release fortiadc-kubernetes-controller/fadc-k8s-ctrl
```

Upgrade the Helm Chart

To upgrade the Helm chart, you need to add a parameter.

```
helm upgrade --set parameters.openshiftRouteSupport="yes" \  
--set parameters.enableStaticRouteSupport="yes" \  
[RELEASE_NAME] fortiadc-kubernetes-controller/fadc-k8s-ctrl
```

In the example below, you can upgrade the helm chart with the release name "first-release".

```
helm upgrade --set parameters.openshiftRouteSupport="yes" \  
--set parameters.enableStaticRouteSupport="yes" \  
first-release fortiadc-kubernetes-controller/fadc-k8s-ctrl
```

Using the `--debug` option, you can check the Helm debug information “USER-SUPPLIED VALUES” to check if you have all the value set as you need.

```
Release "first-release" has been upgraded. Happy Helming!  
NAME: first-release  
LAST DEPLOYED: Fri Aug 4 13:32:10 2023  
NAMESPACE: fortiadc-ingress  
STATUS: deployed  
REVISION: 2  
TEST SUITE: None  
USER-SUPPLIED VALUES:  
parameters:  
  virtualServerWafProfile: mandatory  
  openshiftRouteSupport: yes  
  enableStaticRouteSupport: yes
```

Check the Installation

The `helm history` command shows the installation information.

```
[root@bastion ~]# helm history first-release  
REVISION  UPDATED                   STATUS  CHART              APP VERSION  
DESCRIPTION  
1         Wed Jun 12 13:06:27 2024  deployed  fadc-k8s-ctrl-2.0.2-1  2.0.2-1  Install  
complete
```

Check if FortiADC Kubernetes Controller is installed correctly.

```
[root@ocpexp openshift]# oc get deployment  
NAME                                READY  UP-TO-DATE  AVAILABLE  AGE  
first-release-fadc-k8s-ctrl        1/1    1            1           4d17h
```

Check the FortiADC Kubernetes Controller log.

```
[root@ocpexp openshift]# oc logs first-release-fadc-k8s-ctrl-7cdcfdbdf6-fds6h  
Stopping FortiADC Kubernetes controller  
Starting FortiADC Kubernetes controller  
time="2025-08-25T18:46:09Z" level=info msg="Starting FortiADC Kubernetes controller"  
time="2025-08-25T18:46:09Z" level=info msg="Routes Monitor Enabled"  
time="2025-08-25T18:46:09Z" level=info msg="Static Routes Enabled"
```

Deployment

The deployment instructions for the FortiADC Kubernetes Controller in OpenShift are organized according to the two OpenShift Routes operational modes:

- [Deployment in VxLAN Mode on page 29](#)
- [Deployment in Static Route Mode on page 34](#)

Deployment in VxLAN Mode

The following example shows how the FortiADC connects with the Pod via the pod network when the service type is ClusterIP. To achieve the result, hybrid networking needs to be configured when creating the project.

OpenShift Project is the same as Kubernetes Namespace but with additional features.

Create the Project

Create the new project with the following YAML, and the FortiADC IP 172.23.133.171 and the remote gateway 10.132.100.10.



Due to PDF formatting limitations, the code example below would not retain indentations if copy and pasted directly into a YAML file. Without the proper indentations, the YAML will be invalid.

Please follow this link to copy and modify the namespace YAML example:

https://github.com/fortinet/fortiadc-kubernetes-controller/blob/main/route_examples/hybrid_networking_namespace_example.yaml

```
apiVersion: v1
kind: Namespace
metadata:
  name: fadgw
  annotations:
    k8s.ovn.org/hybrid-overlay-external-gw: 10.132.100.10
    k8s.ovn.org/hybrid-overlay-vtep: 172.23.133.171
```

Configure the Overlay Tunnel in FortiADC

Use the following CLI command to configure the overlay tunnel in FortiADC.

In the example below, note that the OpenShift Node IP is 172.23.133.173. The overlay tunnel interface IP should be the same as what is set in the namespace. The subnet should be 12 as the Pod IP would be under 10.128.0.0/14 subnet.

```
FortiADC-VM # show system overlay-tunnel openshift
config system overlay-tunnel
  edit "openshift"
    set interface port1
    set destination-ip-addresses 172.23.133.173
    set vni 4097
    config remote-host
  end
next
end

FortiADC-VM # show system interface openshift
config system interface
  edit "openshift"
```

```
set type vxlan
set vdom root
set ip 10.132.100.10/12
set allowaccess https ping ssh snmp http telnet
set mtu 1450
config ha-node-ip-list
end
next
end
```

Create the Secret

Most of the resources are isolated between different projects in OpenShift Container Platform 4. So, you will need to create a new secret to log into FortiADC when creating a new project.

```
oc create secret generic fad-login --from-literal=username=<FortiADC-username> --from-literal=password=<FortiADC-password> -n <project-name>
```

Deploy the Pod and Service

Deploy the deployment fad_app/.

The YAML file can be downloaded from here: https://github.com/fortinet/fortiadc-kubernetes-controller/tree/main/route_examples/fad_app/

```
oc apply -f fad_app
```

Check the result:

```
[root@ocpsandy openshift]# oc status
In project fadgw on server https://api.ocp4.ocpsandy.com:6443

svc/fad-app - 172.30.248.97:1234
deployment/fad-app deploys hsandy123/simpleserver:1.0.0-002
deployment #1 running for 23 hours - 3 pods
```

Note that the service can only expose one port. FortiADC Kubernetes Controller in OpenShift does not support services with multiple ports exposed.

Deploy the Routes

Deploy an unsecured Route with path basic_route_with_path.yaml.



Due to PDF formatting limitations, the code example below would not retain indentations if copy and pasted directly into a YAML file. Without the proper indentations, the YAML will be invalid.

Please follow this link to copy and modify the route YAML example:

https://github.com/fortinet/fortiadc-kubernetes-controller/blob/main/route_examples/basic_route_with_path.yaml

```

apiVersion: route.openshift.io/v1
kind: Route
metadata:
  labels:
    name: fad-app
  name: fad-demo-app
  namespace: fadgw
  annotations: {
    "ingress-controller" : "fadc-ingress-controller",
    "fortiadc-ip" : "172.23.133.171",
    "fortiadc-login" : "fad-login",
    "fortiadc-vdom" : "root",
    "fortiadc-ctrl-log" : "enable",
    "virtual-server-ip" : "10.0.0.104",
    "virtual-server-interface" : "port2",
    "virtual-server-port" : "443",
    "load-balance-method" : "LB_METHOD_LEAST_CONNECTION",
    "load-balance-profile" : "LB_PROF_HTTPS"
  }
spec:
  host: testfad.com
  path: "/info"
  port:
    targetPort: 1234-tcp
  to:
    kind: Service
    name: fad-app

```

Deploy the Routes with the following command.

```
oc apply -f fad_route.yaml
```

Describe Routes.

```

[root@ocpexp openshift]# oc describe route fad-demo-app
Name:          fad-demo-app
Namespace:    fadgw
Created:       54 seconds ago
Labels:       name=fad-app
Annotations:  fortiadc-ctrl-log=enable
              fortiadc-ip=172.23.133.171
              fortiadc-login=fad-login
              fortiadc-vdom=root
              ingress-controller=fadc-ingress-controller
              load-balance-method=LB_METHOD_LEAST_CONNECTION
              load-balance-profile=LB_PROF_HTTPS
              virtual-server-interface=port2
              virtual-server-ip=10.0.0.104
              virtual-server-port=443
Requested Host: testfad.com
               exposed on router default (host router-default.apps.ocp411.ocpexp.com)
54 seconds ago
Path:         /info

```

```

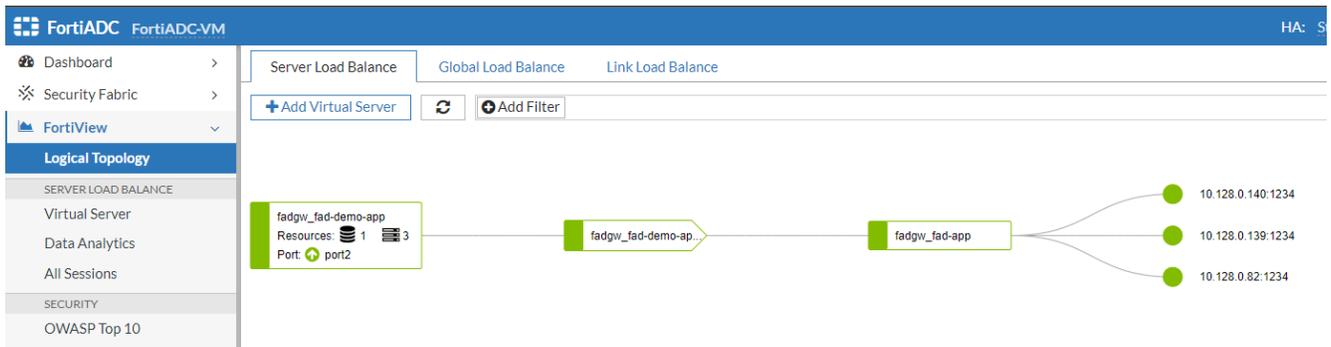
TLS Termination:      <none>
Insecure Policy:     <none>
Endpoint Port:       1234-tcp

Service:             fad-app
Weight:              100 (100%)
Endpoints:           10.128.0.139:1234, 10.128.0.140:1234, 10.128.0.82:1234

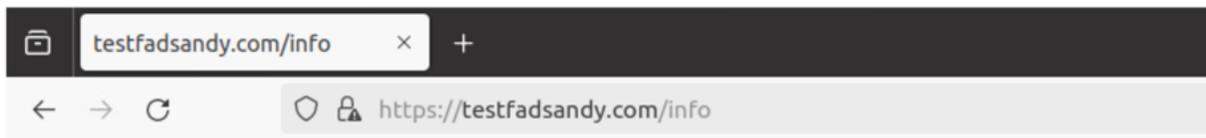
```

FortiView

Check the deployed Routes with FortiView.



Try to access testfad.com with the browser.



HELLO app 1!!

Update or delete the Routes

To update Routes, you can edit the YAML file and apply the file again. This is similar to the process for updating and deleting Kubernetes Ingress. Alternatively, you can just edit the Routes.

```
oc edit route fad-demo-app
```

Delete the Routes with the following command.

```
oc delete route fad-demo-app
```

Update Endpoints

The Endpoints resources are automatically generated by OpenShift, which will automatically store the Pod's IP address and port. You can view the Endpoints resource using the following command.

```
[root@ocpexp openshift]# oc get endpoints
```

```
NAME          ENDPOINTS                                     AGE
fad-app       10.128.0.139:1234,10.128.0.140:1234,10.128.0.150:1234 + 1 more... 5d23h
```

Since FortiADC accesses the Pod via the pod network, FortiADC Kubernetes Controller monitors the Endpoints with the ClusterIP service instead of Nodes.

To update the Endpoints, you need to modify the **replicas** in the deployment.

You can either modify the deployment file and apply again, or just edit the deployment.

```
oc edit deployment fad-app
```

Deployment in Static Route Mode

The following example shows how the FortiADC connects with the Pod via the pod network when the service type is ClusterIP.

OpenShift Project is the same as Kubernetes Namespace but with additional features.

Create the Secret

In OpenShift Container Platform 4, resources are isolated between projects. Each new project requires a new secret to log into FortiADC.

```
oc create secret generic fad-login --from-literal=username=<FortiADC-username> --from-literal=password=<FortiADC-password> -n <project-name>
```

Deploy the Pod and Service

Deploy the deployment fad_app/.

The YAML file can be downloaded from here: https://github.com/fortinet/fortiadc-kubernetes-controller/tree/main/route_examples/fad_app/

```
oc apply -f fad_app
```

Check the result:

```
[root@ocpsandy openshift]# oc status
In project fadgw on server https://api.ocp4.ocpsandy.com:6443

svc/fad-app - 172.30.248.97:1234
  deployment/fad-app deploys hsandy123/simpleserver:1.0.0-002
  deployment #1 running for 23 hours - 3 pods
```

Note that the service can only expose one port. FortiADC Kubernetes Controller in OpenShift does not support services with multiple ports exposed.

Deploy the Routes

Deploy an unsecured Route with path basic_route_with_path.yaml.



Due to PDF formatting limitations, the code example below would not retain indentations if copy and pasted directly into a YAML file. Without the proper indentations, the YAML will be invalid.

Please follow this link to copy and modify the route YAML example:

https://github.com/fortinet/fortiadc-kubernetes-controller/blob/main/route_examples/basic_route_with_path.yaml

```

apiVersion: route.openshift.io/v1
kind: Route
metadata:
  labels:
    name: fad-app
  name: fad-demo-app
  namespace: fadgw
  annotations: {
    "ingress-controller" : "fadc-ingress-controller",
    "fortiadc-ip" : "172.23.133.171",
    "fortiadc-login" : "fad-login",
    "fortiadc-vdom" : "root",
    "fortiadc-ctrl-log" : "enable",
    "virtual-server-ip" : "10.0.0.104",
    "virtual-server-interface" : "port2",
    "virtual-server-port" : "443",
    "load-balance-method" : "LB_METHOD_LEAST_CONNECTION",
    "load-balance-profile" : "LB_PROF_HTTPS"
  }
spec:
  host: testfad.com
  path: "/info"
  port:
    targetPort: 1234-tcp
  to:
    kind: Service
    name: fad-app

```

Deploy the Routes with the following command.

```
oc apply -f fad_route.yaml
```

Describe Routes.

```

[root@ocpexp openshift]# oc describe route fad-demo-app
Name:          fad-demo-app
Namespace:    fadgw
Created:       54 seconds ago
Labels:        name=fad-app
Annotations:   fortiadc-ctrl-log=enable
               fortiadc-ip=172.23.133.171
               fortiadc-login=fad-login
               fortiadc-vdom=root
               ingress-controller=fadc-ingress-controller
               load-balance-method=LB_METHOD_LEAST_CONNECTION
               load-balance-profile=LB_PROF_HTTPS
               virtual-server-interface=port2
               virtual-server-ip=10.0.0.104
               virtual-server-port=443
Requested Host: testfad.com
                exposed on router default (host router-default.apps.ocp411.ocpexp.com)
54 seconds ago
Path:          /info

```

```

TLS Termination:      <none>
Insecure Policy:     <none>
Endpoint Port:       1234-tcp

Service:             fad-app
Weight:              100 (100%)
Endpoints:           10.128.0.139:1234, 10.128.0.140:1234, 10.128.0.82:1234

```

Check the deployed Routes in FortiADC

After deploying the Routes, you can check the corresponding configuration and monitor in FortiADC to ensure the entries have been created successfully.

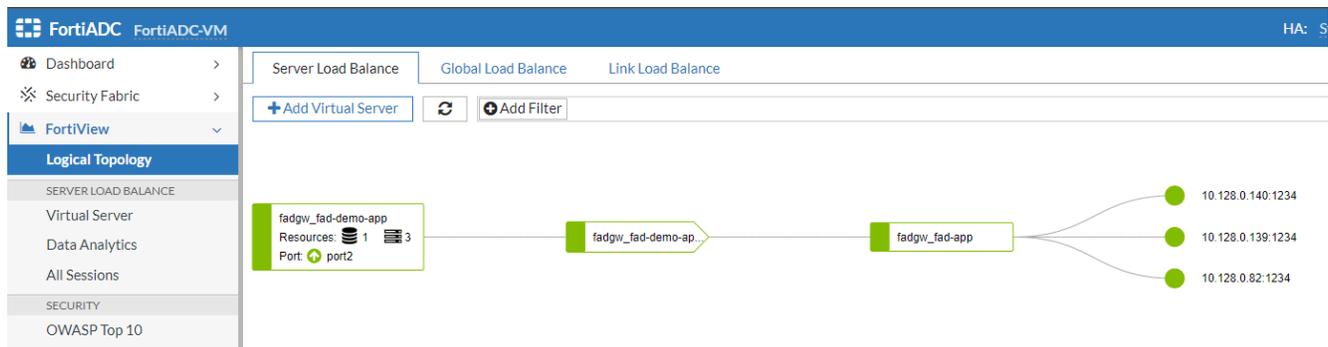
Static Routing

To check that the static Route entry for each node has been added successfully, go to **Network > Routing > Static** tab.

| ID | Destination | Gateway | Distance |
|----|---------------|---------------|----------|
| 1 | 0.0.0.0/0 | 172.23.133.1 | 10 |
| 2 | 10.131.0.0/23 | 172.23.133.46 | 10 |
| 3 | 10.128.2.0/23 | 172.23.133.47 | 10 |
| 4 | 10.128.0.0/23 | 172.23.133.43 | 10 |
| 5 | 10.129.0.0/23 | 172.23.133.44 | 10 |
| 6 | 10.130.0.0/23 | 172.23.133.45 | 10 |

FortiView

Check the deployed Routes in the **FortiView > Logical Topology > Server Load Balance** tab.



Try to access testfad.com with the browser.



HELLO app 1!!

Update or delete the Routes

To update Routes, you can edit the YAML file and apply the file again. This is similar to the process for updating and deleting Kubernetes Ingress. Alternatively, you can just edit the Routes.

```
oc edit route fad-demo-app
```

Delete the Routes with the following command.

```
oc delete route fad-demo-app
```

Update Endpoints and EndpointSlices

Starting from version **3.0.0**, the FortiADC Kubernetes Controller uses the **EndpointSlice** resource (discovery.k8s.io/v1) instead of the legacy **Endpoint** (v1) API, in alignment with Kubernetes' updated service discovery framework.

In OpenShift, **Endpoints** and **EndpointSlices** are automatically generated to store the IP addresses and ports of Pods associated with a Service.

You can view the **EndpointSlice resources** with the following command.

```
[root@ocpexp openshift]# oc get endpointslices
NAME          ADDRESSTYPE  PORTS          ENDPOINTS          AGE
fad-app-lrpcm IPv4          1234 10.128.0.94,10.128.0.95,10.128.0.93 10m
```

Since FortiADC accesses the Pod via the pod network, FortiADC Kubernetes Controller monitors the Endpoints with the ClusterIP service instead of Nodes.

To refresh the Endpoint and EndpointSlice information, modify the number of **replicas** in the deployment.

You can update the deployment file and re-apply it, or edit the deployment directly.

```
oc edit deployment fad-app
```

Configuration parameters

Annotation in Routes

Configuration parameters are required to be specified in the Routes annotation to enable FortiADC Kubernetes Controller to determine how to deploy the Routes resource.

| Parameter | Description | Default |
|--------------------------|---|--|
| ingress-controller | Set the value to fadc-ingress-controller . FortiADC Kubernetes Controller will use this to identify the OpenShift Route. Note: This parameter is required . | |
| fortiadc-ip | The Routes will be deployed on FortiADC with the given IP address or domain name. Note: This parameter is required . | |
| fortiadc-admin-port | FortiADC HTTPS service port. | 443 |
| fortiadc-login | The OpenShift secret name preserves the FortiADC authentication information. Note: This parameter is required . | |
| fortiadc-vdom | Specify which VDOM to deploy the Routes resource if VDOM is enabled on FortiADC. | root |
| fortiadc-ctrl-log | Enable/disable the FortiADC Kubernetes Controller log. Once enabled, FortiADC Kubernetes Controller will print the verbose log the next time the Routes is updated. | enable |
| virtual-server-ip | The virtual server IP of the virtual server to be configured on FortiADC. This IP will be used as the address of the Routes. Note: This parameter is required . | |
| virtual-server-interface | The FortiADC network interface for the client to access the virtual server. Note: This parameter is required . | |
| virtual-server-port | Default is 80. If TLS is specified in the Routes, then the default is 443. Note: If the fortiadc-ip is the same as the virtual-server-ip , you should specify virtual-server-port to be other than 80/443 or change the system default reserved HTTP/HTTPS port on FortiADC. | 80 for HTTP service. 443 for HTTPS service. |

| Parameter | Description | Default |
|--------------------------------|--|-------------------------------|
| | For more details, see the FortiADC Administration Guide on Management service ports . | |
| load-balance-method | Specify the predefined or user-defined method configuration name. For more details, see the FortiADC Administration Guide on load balancing methods . | LB_METHOD_ROUND_ROBIN |
| load-balance-profile | Default is LB_PROF_HTTP. If TLS is specified in the Routes, then the default is LB_PROF_HTTPS. | LB_PROF_HTTP LB_PROF_HTTPS |
| virtual-server-addr-type | IPv4 or IPv6. | ipv4 |
| virtual-server-traffic-group | Specify the traffic group for the virtual server. For more details, see the FortiADC Administration Guide on traffic groups . | default |
| virtual-server-nat-src-pool | Specify the NAT source pool. For more details, see the FortiADC Administration Guide on NAT source pools . | |
| virtual-server-waf-profile | Specify the WAF profile name. For more details, see the FortiADC Administration Guide on WAF profiles . | |
| virtual-server-av-profile | Specify the AV profile name. For more details, see the FortiADC Administration Guide on AV profiles . | |
| virtual-server-dos-profile | Specify the DoS profile name. For more details, see the FortiADC Administration Guide on DoS profiles . | |
| virtual-server-captcha-profile | Specify the Captcha profile name. For more details, see the FortiADC Administration Guide on Captcha profiles . Note: This field is available if WAF profile or DoS profile is specified. | |
| virtual-server-fortiview | Enable/disable FortiView. | disable |
| virtual-server-traffic-log | Enable/disable the traffic log. | disable |
| virtual-server-wccp | Enable/disable WCCP. For more details, see the FortiADC Administration Guide on WCCP . | disable |
| virtual-server-persistence | Specify a predefined or user-defined persistence configuration name. | |

| Parameter | Description | Default |
|---------------------------------------|---|---------|
| | For more details, see the FortiADC Administration Guide on persistence rules . | |
| virtual-server-fortigsb-publicip-type | Specify the public IP type for the virtual server as either IPv4 or IPv6. | ipv4 |
| virtual-server-fortigsb-publicip | Specify the virtual server public IP address. | |
| virtual-server-fortigsb-1clickgslb | Enable/disable the FortiGSLB One-click GSLB server. | disable |
| virtual-server-fortigsb-hostname | The Host Name option is available if One-click GSLB Server is enabled. Enter the hostname part of the FQDN. For example: www. Note: You can use @ to denote the zone root. The value substitute for @ is the preceding \$ORIGIN directive. | |
| virtual-server-fortigsb-domainname | The Domain Name option is available if One-click GSLB Server is enabled. The domain name must end with a period. For example: example.com. | |

For more details on configuring parameters with virtual-server prefix and load-balance prefix, please reference [FortiADC Administration Guide on Configuring virtual servers](#).

Annotation in Service

You can define the health check profile and SSL profile in the OpenShift service annotation.

The health check profile and SSL profile will be automatically configured in the corresponding real server pool on FortiADC.

| Parameter | Description | Default |
|-----------------------|--|---------|
| health-check-ctrl | Enable/disable the health checking for the real server pool. | disable |
| health-check-relation | <ul style="list-style-type: none"> • AND – All of the selected health checks must pass for the server to be considered available. • OR – One of the selected health checks must pass for the server to be considered available. | |
| health-check-list | One or more health check configuration names. Concatenate the health check names with a space between each name. For example: "LB_HLTHCK_ICMP LB_HLTHCK_HTTP". For more details, see the FortiADC Administration Guide on health checks . | |

| Parameter | Description | Default |
|-------------------------|--|---------|
| real-server-ssl-profile | Specify the real server SSL profile name. Real server profiles determine settings for communication between FortiADC and the backend real servers. The default is NONE, which is applicable for non-SSL traffic. For more details, see the FortiADC Administration Guide on SSL profiles . | NONE |



Due to PDF formatting limitations, the code example below would not retain indentations if copy and pasted directly into a YAML file. Without the proper indentations, the YAML will be invalid.

To copy the service YAML example, follow this link:

https://github.com/fortinet/fortiadc-kubernetes-controller/blob/main/service_examples/default-http-backend.yaml

Here is an example service.yaml with health check parameters:

```
kind: Service
apiVersion: v1
metadata:
  labels:
    name: default-http-backend
    namespace: default
  annotations: {
    "health-check-ctrl" : "enable",
    "health-check-relation" : "OR",
    "health-check-list" : "LB_HLTHCK_ICMP",
    "real-server-ssl-profile" : "NONE"
  }
spec:
  type: NodePort
  ports:
  - port: 80
    protocol: TCP
    targetPort: 80
  selector:
    app: nginx
  sessionAffinity: None
```

Debug

To see the debug log, you can use the `oc logs` command.

```
oc logs -n [namespace] -f [FortiADC Kubernetes Controller pod name]
```

The log shows which problems you have encountered. For example, the log below shows that you did not have the correct FortiADC Authentication Secret in the OpenShift cluster.

```
time="2023-04-19T03:22:15Z" level=warning msg="Get fortiadc-login secret failed for fadgw/fad-demo-app: secret \"fad-login2\" not found."
time="2023-04-19T03:22:15Z" level=info msg="Handle updating route fadgw/fad-demo-app done"
```

Based on the error message, you can correct it and use the `oc apply` command to reconfigure the Routes.

Some troubleshooting steps may require restarting the FortiADC Kubernetes Controller. For example, the FortiADC Kubernetes Controller may not connect to FortiADC after changing the network firewall rule. To fix this type of environment issue, you can restart the FortiADC Kubernetes Controller by using the following command.

```
oc -n [namespace] rollout restart deployment/[FortiADC Kubernetes Controller deployment name]
```



www.fortinet.com

Copyright© 2026 Fortinet, Inc. All rights reserved. Fortinet®, FortiGate®, FortiCare® and FortiGuard®, and certain other marks are registered trademarks of Fortinet, Inc., and other Fortinet names herein may also be registered and/or common law trademarks of Fortinet. All other product or company names may be trademarks of their respective owners. Performance and other metrics contained herein were attained in internal lab tests under ideal conditions, and actual performance and other results may vary. Network variables, different network environments and other conditions may affect performance results. Nothing herein represents any binding commitment by Fortinet, and Fortinet disclaims all warranties, whether express or implied, except to the extent Fortinet enters a binding written contract, signed by Fortinet's General Counsel, with a purchaser that expressly warrants that the identified product will perform according to certain expressly-identified performance metrics and, in such event, only the specific performance metrics expressly identified in such binding written contract shall be binding on Fortinet. For absolute clarity, any such warranty will be limited to performance in the same ideal conditions as in Fortinet's internal lab tests. Fortinet disclaims in full any covenants, representations, and guarantees pursuant hereto, whether express or implied. Fortinet reserves the right to change, modify, transfer, or otherwise revise this publication without notice, and the most current version of the publication shall be applicable.