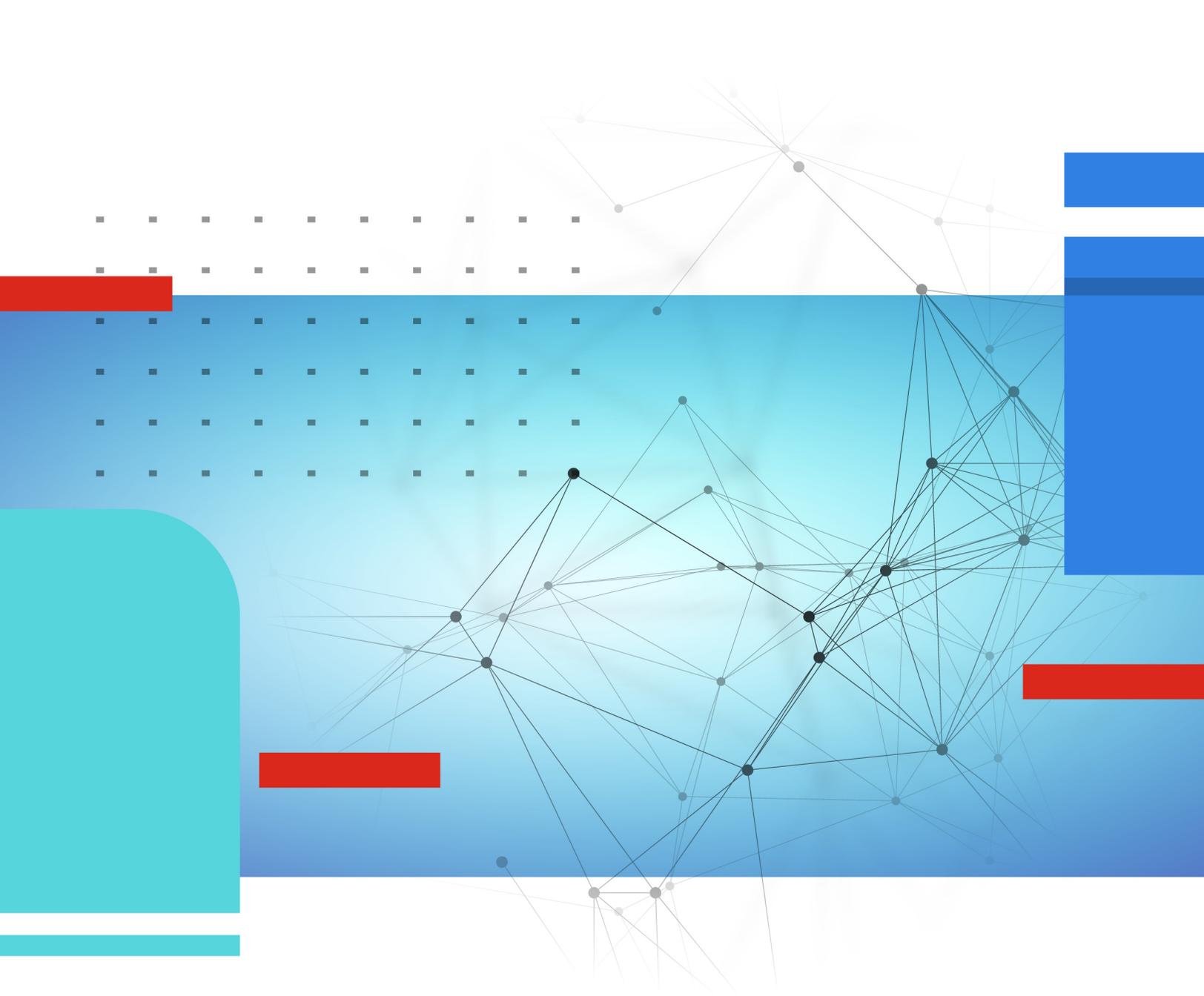




# FortiWeb Ingress Controller 2.1.0 - OpenShift Installation Guide

FortiWeb 7.6.0





# TABLE OF CONTENTS

<b>Change Log</b> .....	<b>4</b>
<b>Prerequisite knowledge</b> .....	<b>5</b>
<b>Supported Environments</b> .....	<b>6</b>
<b>FortiWeb Ingress Controller with OpenShift</b> .....	<b>7</b>
OpenShift Routes .....	7
Mapping of the Routes related resources with the FortiWeb objects .....	7
Supported Routes types .....	7
Installation in Static Route Mode .....	11
Installing FortiWeb Ingress Controller in OpenShift with Static Route .....	11
Deployment in Static Route Mode .....	18
Configuration parameters .....	24
Debug .....	27

# Change Log

Date	Change Description
January 9, 2025	Initial release

# Prerequisite knowledge

## OpenShift

Before you begin using FortiWeb Ingress Controller on the OpenShift Container Platform, you will need to have prerequisite knowledge of the OpenShift cluster and OpenShift Routes. For more information, please refer to the documents listed below:

- OpenShift Container Platform Concepts: [https://docs.openshift.com/container-platform/4.12/getting\\_started/openshift-overview.html](https://docs.openshift.com/container-platform/4.12/getting_started/openshift-overview.html)
- OpenShift Container Platform network plugin – OVN-Kubernetes: [https://docs.openshift.com/container-platform/4.12/networking/ovn\\_kubernetes\\_network\\_provider/about-ovn-kubernetes.html](https://docs.openshift.com/container-platform/4.12/networking/ovn_kubernetes_network_provider/about-ovn-kubernetes.html)
- OpenShift Routes: <https://docs.openshift.com/container-platform/4.12/networking/routes/route-configuration.html>

## Helm Charts

As Helm Charts are used in FortiWeb Ingress 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/](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/](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/>

# Supported Environments

## Supported Release and Version

Product	Version				
FortiWeb Ingress Controller	1.0.0	1.0.1	1.0.2	2.0.0	2.1.0
OpenShift Container Platform	Not supported	Not supported	Not supported	Not supported	4.7 - 4.12.x 4.13 - 4.15.x**
FortiWeb Version	7.6.x*				

\*\*FortiWeb Ingress Controller version 2.1.0 introduces support for Static Route Mode, which is supported in the OpenShift Container Platform versions 4.13 through 4.15.

# FortiWeb Ingress Controller with OpenShift

OpenShift Container Platform 4 supports FortiWeb Ingress Controller with Kubernetes Ingress when the service type is NodePort. FortiWeb Ingress Controller can also handle Add/Update/Delete events for watching for watching Routes resources and automatically implement corresponding actions on FortiWeb when the service Type is ClusterIP.

## OpenShift Routes

Routes is an OpenShift object that can host the application at a public URL with service type ClusterIP. The URL can be either secured or unsecured, depending on the user configuration.

FortiWeb Ingress Controller monitors OpenShift Routes events and implements OpenShift Routes load balancer onto FortiWeb.

FortiWeb Ingress Controller supports OpenShift Routes in Static Routes in versions 4.13 through 4.15, which relies on predefined network routes for traffic management.

Kubernetes Ingress can also be deployed in the OpenShift environment via FortiWeb Ingress Controller. Please refer to FortiWeb Ingress Controller 2.0 -Kubernetes document for more details.



There are some limitations to what FortiWeb Ingress 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 FortiWeb objects

OpenShift Objects	FortiWeb Objects
Routes	Virtual server Content Routing Scripting
Service	Real Server Pool
Endpoints	Real Server

## Supported Routes types

FortiWeb supports both insecure and secured Routes.

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

## Insecure Routes with path

```
apiVersion: route.openshift.io/v1
kind: Route
metadata:
  labels:
    name: minimal-ingress
  name: minimal-ingress
  namespace: marco
  annotations: {
    "ingress-controller" : "fwb-ingress-controller",
    "fortiweb-ip" : "172.23.133.148",
    "fortiweb-port" : "443",
    "fortiweb-login" : "fad-login1",
    "fortiweb-ctrl-log" : "enable",
    "virtual-server-ip" : "192.23.133.6",
    "virtual-server-addr-type" : "ipv4",
    "virtual-server-interface" : "port3",
    "server-policy-web-protection-profile" : "Inline Standard Protection",
    "server-policy-https-service" : "HTTPS",
    "server-policy-http-service" : "HTTP",
    "server-policy-syn-cookie" : "enable",
    "static_route_intf" : "port1",
    "server-policy-http-to-https" : "disable"
  }
spec:
  host: testfad.com
  path: "/info"
  port:
    targetPort: 1234-tcp
  to:
    kind: Service
    name: fad-app
```

## Insecure Routes without path

```
apiVersion: route.openshift.io/v1
kind: Route
metadata:
  labels:
    name: minimal-ingress
  name: minimal-ingress
  namespace: marco
  annotations: {
    "ingress-controller" : "fwb-ingress-controller",
    "fortiweb-ip" : "172.23.133.148",
    "fortiweb-port" : "443",
```

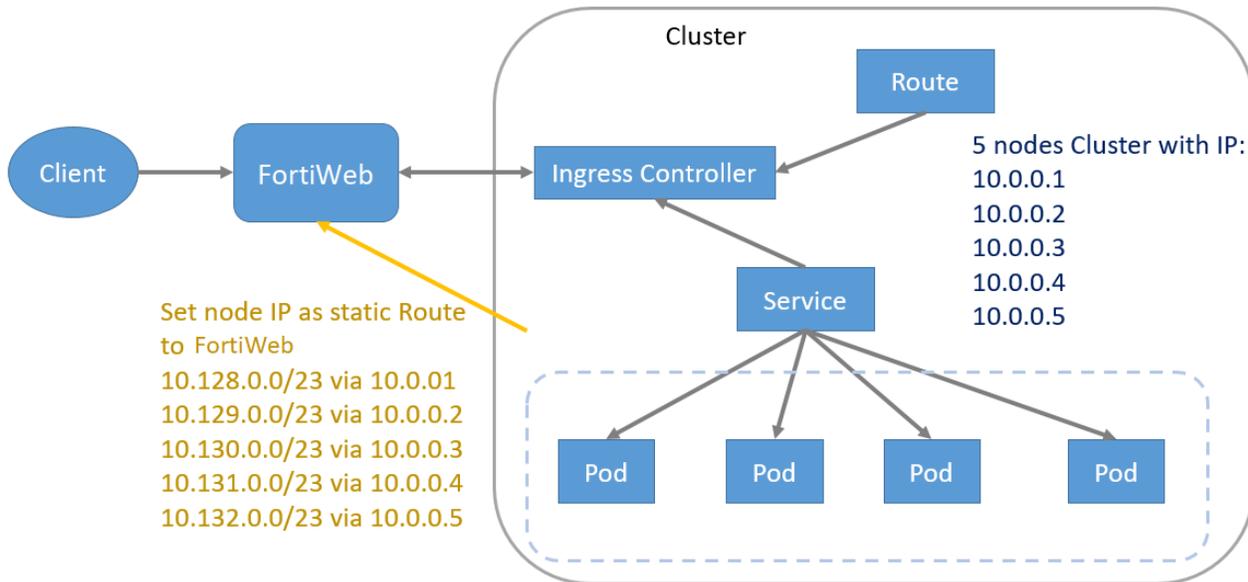
```
    "fortiweb-login" : "fad-login1",
    "fortiweb-ctrl-log" : "enable",
    "virtual-server-ip" : "192.23.133.6",
    "virtual-server-addr-type" : "ipv4",
    "virtual-server-interface" : "port3",
    "server-policy-web-protection-profile" : "Inline Standard Protection",
    "server-policy-https-service" : "HTTPS",
    "server-policy-http-service" : "HTTP",
    "server-policy-syn-cookie" : "enable",
    "static_route_intf" : "port1",
    "server-policy-http-to-https" : "disable"
  }
spec:
  host: testfad.com
  port:
    targetPort: 1234-tcp
  to:
    kind: Service
    name: fad-app
```

## Secured Routes

```
apiVersion: route.openshift.io/v1
kind: Route
metadata:
  labels:
    name: tls
  name: tls
  namespace: marco
annotations: {
  "ingress-controller" : "fwb-ingress-controller",
  "fortiweb-ip" : "172.23.133.148",
  "fortiweb-port" : "443",
  "fortiweb-login" : "fad-login1",
  "fortiweb-ctrl-log" : "enable",
  "virtual-server-ip" : "192.23.133.6",
  "virtual-server-addr-type" : "ipv4",
  "virtual-server-interface" : "port3",
  "server-policy-web-protection-profile" : "Inline Standard Protection",
  "server-policy-https-service" : "HTTPS",
  "server-policy-http-service" : "HTTP",
  "server-policy-syn-cookie" : "enable",
  "static_route_intf" : "port1",
  "server-policy-http-to-https" : "disable",
  "content-routing-web-protect-profile" : "test.com_fad-app:Inline Alert Only"
}
spec:
  host: test.com
```

```
tls:
  certificate: |
    -----BEGIN CERTIFICATE-----
    -----END CERTIFICATE-----
  key: |
    -----BEGIN PRIVATE KEY-----
    -----END PRIVATE KEY-----
  termination: edge
path: "/info"
port:
  targetPort: 1234-tcp
to:
  kind: Service
  name: fad-app
```

## Installation in Static Route Mode



In Static Route mode, the FortiWeb Ingress Controller configures static route entries for each node's pod network subnet and node IP on FortiWeb 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 FortiWeb OpenShift connector with Static Routes supports OpenShift Container Platform versions 4.13 to 4.15.x.

## Installing FortiWeb Ingress Controller in OpenShift with Static Route

Install FortiWeb Ingress 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 FortiWeb Ingress Controller in the OpenShift cluster. By using the Helm 3 installation tool, most of the OpenShift objects required for FortiWeb Ingress Controller can be deployed in one simple command.

The OpenShift objects required for FortiWeb Ingress Controller are listed below:

OpenShift object	Description
Deployment	By configuring the replica and pod template in the OpenShift deployment, the deployment ensures FortiWeb Ingress Controller provides a non-terminated service.
Service Account	The service account is used in FortiWeb Ingress 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 FortiWeb Ingress Controller, allowing FortiWeb Ingress Controller to access and operate the OpenShift cluster-scoped Routes-related objects.
Ingress Class	The IngressClass "fweb-ingress-controller" is created for FortiWeb Ingress Controller to identify the Ingress resource. If the Ingress is defined with the IngressClass "fweb-ingress-controller", FortiWeb Ingress Controller will manage this Ingress resource as FortiWeb Ingress 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 FortiWeb Ingress 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.

```
image:

  repository: fortinet/fortiweb-ingress

  pullPolicy: IfNotPresent

  tag: "2.1.0"

  nameOverride: ""

  fullnameOverride: ""

  serviceAccount:

  create: true

  annotations: {}

  name: "fortiweb-ingress"

  podAnnotations: {}
```

---

```
podSecurityContext: {}

securityContext: {}

# capabilities:

# drop:

# - ALL

# readOnlyRootFilesystem: true

# runAsNonRoot: true

# runAsUser: 1000

nodeSelector: {}

tolerations:

- effect: "NoExecute"

key: "node.kubernetes.io/not-ready"

operator: "Exists"

tolerationSeconds: 30

- effect: "NoExecute"

key: "node.kubernetes.io/unreachable"

operator: "Exists"

tolerationSeconds: 30

affinity: {}

# Define Ingress Class for FortiWEB Ingress Controller

controller:

ingressClassResource:
```

---

```
        name: "fwb-ingress-controller"

        enabled: true

        default: true

        controllerValue: "fortinet.com/fw-ingress-controller"

        parameters:

            openshiftRouteSupport: "no"
image:
    repository: fortinet/fortiweb-ingress
    pullPolicy: IfNotPresent
    tag: "2.1.0"

nameOverride: ""
fullnameOverride: ""

serviceAccount:
    create: true
    annotations: {}
    name: "fortiweb-ingress"

podAnnotations: {}

podSecurityContext: {}

securityContext: {}
    # capabilities:
    # drop:
    # - ALL
    # readOnlyRootFilesystem: true
    # runAsNonRoot: true
    # runAsUser: 1000
```

---

```
nodeSelector: {}

tolerations:
  - effect: "NoExecute"
    key: "node.kubernetes.io/not-ready"
    operator: "Exists"
    tolerationSeconds: 30
  - effect: "NoExecute"
    key: "node.kubernetes.io/unreachable"
    operator: "Exists"
    tolerationSeconds: 30

affinity: {}

# Define Ingress Class for FortiWEB Ingress Controller
controller:
  ingressClassResource:
    name: "fwb-ingress-controller"
    enabled: true
    default: true
    controllerValue: "fortinet.com/fwbingress-controller"

parameters:
  openshiftRouteSupport: "no"
```



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 16](#) and [Installation in Static Route Mode on page 11](#)). For more details on the parameters, see [Configuration parameters on page 24](#).

---

---

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

## Get Repo Information

To get the repository information:

```
helm repo add fortibweb-ingress-controller https://fortinet.github.io/fortibweb-  
ingress/  
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**.

To enable Ingress 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 FortiWeb Ingress 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 FortiWeb Ingress Controller.

```
helm install --set parameters.openshiftRouteSupport="yes" \  
--namespace [OpenShift Project] [RELEASE_NAME] fwb-ingress-controller/fwb-k8s-  
ctrl
```

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

```
helm install --set parameters.openshiftRouteSupport="yes" --namespace marco  
first-release fwb-ingress-controller/fwb-k8s-ctrl
```

## 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   fwk-k8s-ctrl-2.0.2-1  2.0.2-1
Install complete
```

Check if FortiWeb Ingress Controller is installed correctly.

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

Check the FortiWeb Ingress Controller log.

```
[root@ocpexp openshift]# oc logs first-release-fwk-k8s-ctrl-7cdcfd6df6-fds6h
Stopping fortiweb ingress controller
Starting fortiweb ingress controller
time="2024-06-05T05:42:38Z" level=info msg="Starting FortiWeb Ingress controller"
time="2024-06-05T05:42:39Z" level=info msg="Routes Monitor Enabled"
time="2024-06-05T05:42:39Z" level=info msg="Static Routes Enabled"
```

---

## Deployment in Static Route Mode

The following example shows how the FortiWeb 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 FortiWeb.

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

### Deploy the Pod and Service

Deploy the deployment `fad_app/`.

#### **fad\_deploy.yaml:**

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: fad-app
  namespace: marco
  labels:
    app: fad-app
spec:
  replicas: 2
  selector:
    matchLabels:
      deployment: fad-app
  template:
    metadata:
      labels:
        deployment: fad-app
    spec:
      containers:
      - env:
        - name: service_name
          value: fad-app
        - name: WORD
          value: "HELLO app 1!!"
        image: hsandy123/simpleserver:1.0.0-004
        imagePullPolicy: Always
        name: fad-app
```

---

```
ports:
- containerPort: 1234
  protocol: TCP
securityContext:
  allowPrivilegeEscalation: false
  runAsNonRoot: true
  seccompProfile:
    type: RuntimeDefault
  capabilities:
    drop: ["ALL"]
```

### **fad\_service.yaml:**

```
apiVersion: v1
kind: Service
metadata:
  name: fad-app
  namespace: marco
  labels:
    app: fad-app
spec:
  ports:
  - name: 1234-tcp
    port: 1234
    protocol: TCP
    targetPort: 1234
  type: ClusterIP
  selector:
  deployment: 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. FortiWeb Ingress 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/fortiweb-ingress/blob/main/route\\_examples/minimal-ingress.yaml](https://github.com/fortinet/fortiweb-ingress/blob/main/route_examples/minimal-ingress.yaml)

```
apiVersion: route.openshift.io/v1
kind: Route
metadata:
  labels:
    name: minimal-ingress
name: minimal-ingress
annotations: {
  "ingress-controller" : "fwb-ingress-controller",
  "fortiweb-ip" : "172.23.133.161",
    "fortiweb-port": "443",
  "fortiweb-login" : "fad-login",
  "fortiweb-ctrl-log" : "enable",
    "virtual-server-ip" : "172.23.133.162",
    "virtual-server-addr-type" : "ipv4",
    "virtual-server-interface" : "port1",
    "server-policy-web-protection-profile" : "Inline Standard Protection",
    "server-policy-https-service" : "HTTPS",
    "server-policy-http-service" : "HTTP",
    "server-policy-syn-cookie" : "enable",
    "static_route_intf" : "port1",
    "server-policy-http-to-https" : "disable",
}
spec:
  host: test.com
  path: "/info"
  port:
    targetPort: 1234-tcp
  to:
    kind: Service
    name: fad-app
```

Deploy the Routes with the following command.

```
oc apply -f minimal-ingress.yaml
```

Describe Routes.

```
[root@bastion]# oc describe route minimal-ingress
Name:          minimal-ingress
Created:       About a minute ago
Labels:        name=minimal-ingress
Annotations:   fortiweb-ctrl-log=enable
               fortiweb-ip=172.23.133.161
```

```
fortiweb-login=fad-login
fortiweb-port=443
ingress-controller=fwb-ingress-controller
```

```
kubectl.kubernetes.io/last-applied-configuration={"apiVersion":"route.openshift.io/v1","kind":"Route","metadata":{"annotations":{"fortiweb-ctrl-log":"enable","fortiweb-ip":"172.23.133.161","fortiweb-login":"fad-login","fortiweb-port":"443","ingress-controller":"fwb-ingress-controller","machine-learning-anomaly-add-domain":"ti5.com",
```

```
"machine-learning-anomaly-add-ip":"4.4.5.6-4.4.5.7","machine-learning-api-add-domain":"t45","machine-learning-api-add-ip":"13.1.2.3-13.1.2.4","machine-learning-bot-add-ip":"6.5.6.7-6.5.6.8","server-policy-http-service":"HTTP","server-policy-http-to-https":"disable","server-policy-https-service":"HTTPS","server-policy-syn-cookie":"enable","server-policy-web-protection-profile":"Inline Standard Protection","static_route_intf":"port1","virtual-server-addr-type":"ipv4","virtual-server-interface":"port1","virtual-server-ip":"172.23.133.162"},"labels":{"name":"minimal-ingress"},"namespace":"dh"},"spec":{"host":"test.com","path":"/info","port":{"targetPort":"1234-tcp"},"to":{"kind":"Service","name":"fad-app"}}}
```

```
server-policy-http-service=HTTP
```

```
server-policy-http-to-https=disable
```

```
server-policy-https-service=HTTPS
```

```
server-policy-syn-cookie=enable
```

```
server-policy-web-protection-profile=Inline Standard Protection
```

```
static_route_intf=port1
```

```
virtual-server-addr-type=ipv4
```

```
virtual-server-interface=port1
```

```
virtual-server-ip=172.23.133.162
```

```
Requested Host: test.com
```

```
minute ago
```

```
exposed on router default (host router-default.apps.ocp.fortinet) ab
```

```
Path: /info
```

```
TLS Termination: <none>
```

```
Insecure Policy: <none>
```

---

Endpoint Port: 1234-tcp

Service: fad-app  
Weight: 100 (100%)  
Endpoints: 10.128.3.181:1234, 10.131.0.53:1234

## Check the deployed Routes in FortiWeb

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

### Static Routing

Try to access testfad.com with the browser.



## 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 minimal-ingress
```

Delete the Routes with the following command.

```
oc delete route minimal-ingress
```

## 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@bastion duncan]# oc get endpoints
```

NAME	ENDPOINTS	AGE
fad-app	10.128.3.181:1234,10.131.0.53:1234	26d

---

```
service2 10.128.2.247:80
```

```
56d
```

Since FortiWeb accesses the Pod via the pod network, FortiWeb Ingress 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
```

# Configuration parameters

## Annotation in Routes

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

Parameter	Description
ingress-controller	Set the value to <b>fwb-ingress-controller</b> . FortiWeb Ingress Controller will use this to identify the OpenShift Route. <b>Note:</b> This parameter is <b>required</b> .
fortiweb-ip	The IP address to log in to FortiWeb.
fortiweb-login	The Kubernetes secret name preserves the FortiWeb authentication information.
fortiweb-port	The network port to log in to Fortiweb.
fortiweb-ctrl-log	The virtual server IP of the virtual server to be configured on FortiWeb. This IP will be used as the address of the Ingress Controller.
machine-learning-anomaly-add-domain	The domain list added to Anomaly detection. For more information, see "ML Based Anomaly Detection" in FortiWeb Administration Guide.
machine-learning-anomaly-add-ip	The IP list added to Anomaly detection. For more information, see "ML Based Anomaly Detection" in FortiWeb Administration Guide.
machine-learning-anomaly-ip-list-type	The IP address should block or accept. For more information, see "ML Based Anomaly Detection" in FortiWeb Administration Guide.
machine-learning-anomaly-del-domain-list	The domain list deleted from Anomaly detection. For more information, see "ML Based Anomaly Detection" in FortiWeb Administration Guide.
machine-learning-anomaly-del-ip	The IP list deleted from Anomaly detection. For more information, see "ML Based Anomaly Detection" in FortiWeb Administration Guide.
machine-learning-bot-add-ip	The IP list added to ML Based Bot Detection. For more information, see "Configuring ML Based Bot Detection policy" in FortiWeb Administration Guide.
machine-learning-bot-del-ip	The IP list deleted from ML Based Bot Detection. For more information, see "Configuring ML Based Bot Detection policy" in FortiWeb Administration Guide.

Parameter	Description
machine-learning-api-add-domain	The domain list added to ML Based API Protection. For more information, see "Configuring ML Based API Protection policy" in FortiWeb Administration Guide.
machine-learning-api-del-domain	The domain list deleted from ML Based API Protection. For more information, see "Configuring ML Based API Protection policy" in FortiWeb Administration Guide.
machine-learning-api-add-ip	The IP list added to ML Based API Protection. For more information, see "Configuring ML Based API Protection policy" in FortiWeb Administration Guide.
machine-learning-api-del-ip	The IP list deleted from ML Based API Protection. For more information, see "Configuring ML Based API Protection policy" in FortiWeb Administration Guide.
machine-learning-api-ip-list-type	The IP address should block or accept in ML Based API Protection. For more information, see "Configuring ML Based API Protection policy" in FortiWeb Administration Guide.
virtual-server-addr-type	IPv4 or IPv6.
virtual-server-ip	The virtual server IP of the virtual server to be configured on FortiWeb. This IP will be used as the address of the Ingress.
virtual-server-interface	The FortiWeb network interface for the client to access the virtual server.
server-policy-web-protection-profile	The name of the web protection profile to be applied to the virtual server.
server-policy-https-service	The HTTPS service name.
server-policy-http-service	The HTTP service name.
server-policy-syn-cookie	Enable/Disable the SYN cookie.
server-policy-http-to-https	Enable/Disable the http-to-https function.
azure-funapp	This parameter is the azure funapp URL to retrieve the primary node's public IP when there is a FortiWeb autoscale cluster, for instance, <a href="https://funappmue2gs2ywvg4i.azurewebsites.net/api/AutoscaleHandler">https://funappmue2gs2ywvg4i.azurewebsites.net/api/AutoscaleHandler</a> .

For more details on configuring parameters with virtual-server prefix and load-balance prefix, please reference [FortiWeb Handbook 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 FortiWeb.

Parameter	Description	Parameter
health-check-ctrl	Enable/disable the health checking for the back-end server pool. For more information, see "Defining your web servers" in FortiWeb Administration Guide.	health-check-ctrl
lb-algo	Specify the load balancing algorithm. For more information, see "Defining your web servers" in FortiWeb Administration Guide.	lb-algo
persistence	Specify the persistence rule. For more information, see "Defining your web servers" in FortiWeb Administration Guide.	persistence



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/fortiweb-ingress/blob/main/service\\_examples/default-http-backend.yaml](https://github.com/fortinet/fortiweb-ingress/blob/main/service_examples/default-http-backend.yaml)

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

```
kind: Service
apiVersion: v1
metadata:
  name: service2
  namespace: default
  annotations: {
    "health-check-ctrl" : "HLTHCK_ICMP",
    "lb-algo" : "round-robin",
  }
spec:
  type: NodePort
  ports:
  - port: 1242
    protocol: TCP
    targetPort: 80
  selector:
    run: nginx-demo
  sessionAffinity: None
```

---

## Debug

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

```
oc logs -n [namespace] -f [FortiWeb Ingress Controller pod name]
```

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

```
time="2023-04-19T03:22:15Z" level=warning msg="Get fortweb-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 FortiWeb Ingress Controller. For example, the FortiWeb Ingress Controller may not connect to FortiWeb after changing the network firewall rule. To fix this type of environment issue, you can restart the FortiWeb Ingress Controller by using the following command.

```
oc -n [namespace] rollout restart deployment/[ FortiWeb Ingress Controller deployment name]
```

