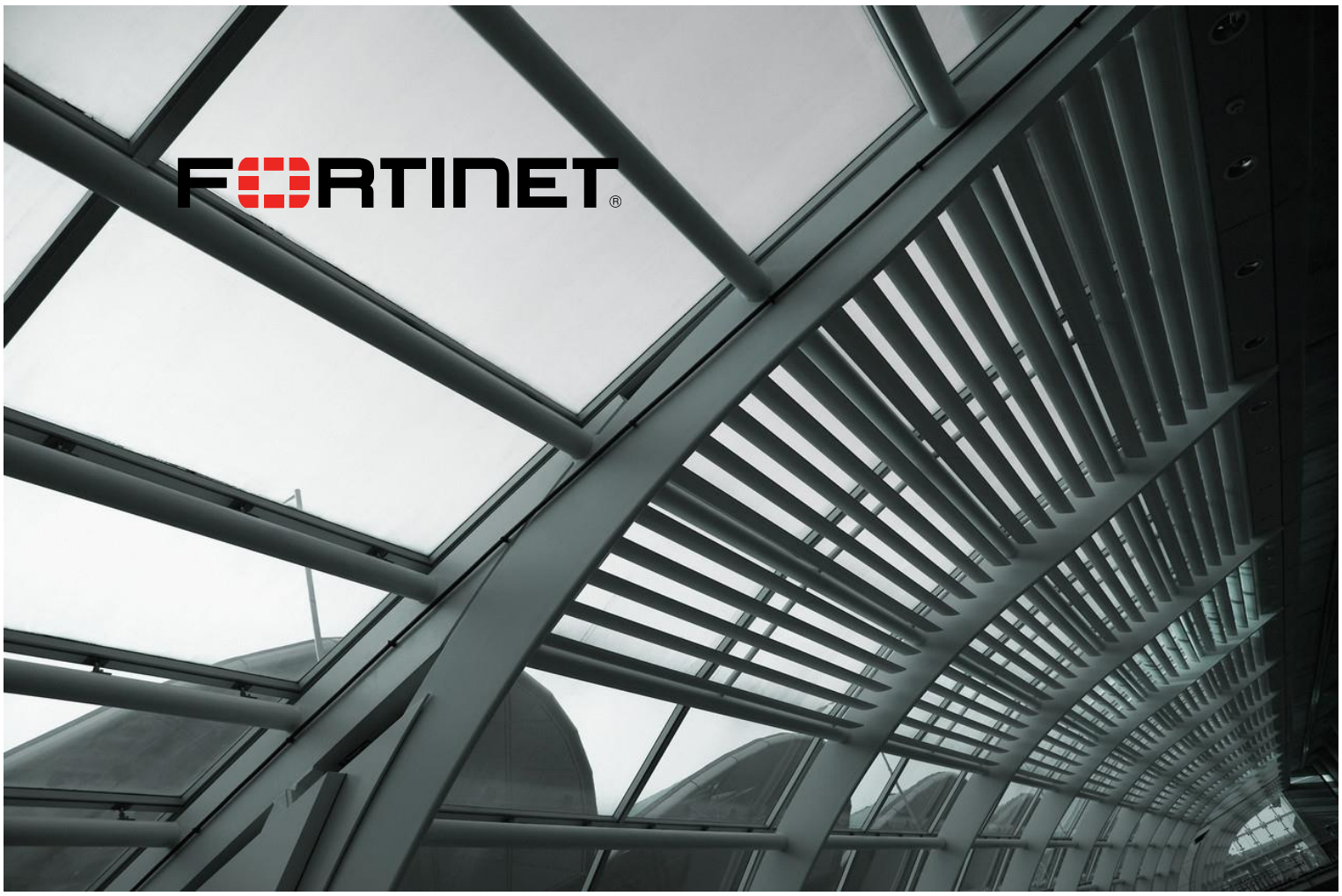


FortiADC ADFS proxy Deployment Guide

FortiADC ADFS proxy Deployment Guide

TABLE OF CONTENTS

1 ABOUT THIS GUIDE

This guide details the steps required to configure the FortiADC AD FS Proxy function. The AD FS Proxy is a service that brokers a connection between external users and your internal AD FS server. It acts as a reverse proxy and typically resides in your organization's perimeter network (aka DMZ). As far as the user is concerned, they do not know they are talking to an AD FS proxy server, as the federation services are accessed by the same URLs.

This guide describes the configuration for AD FS Proxy authentication in each scenario, whether for Office365 or Exchange.

When the FortiADC works through AD FS Proxy, it is quite similar to how the WAP works. Both WAP and ADC support two preauthentication methods: besides AD FS, there is pass-through.

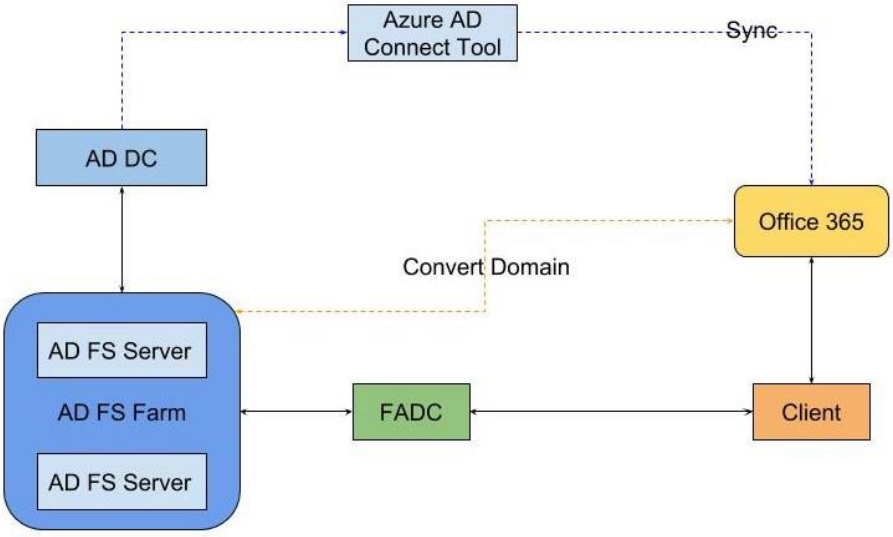
In the pass-through method, (1) no preauthentication is performed by AD FS Proxy, all requests are forwarded to the backend server. In AD FS(Active Directory Federation Services) method, (2) all unauthenticated client requests are redirected to the federation server. After successful authentication by AD FS, client requests are forwarded to the backend server.

Lastly, the FortiADC supports Office365 service in its pass-through method, since Office356 lies in the external web and the AD FS server in the internal web.

2 AD FS PROXY SCENARIO OVERVIEW

2.1 Scenario 1: Office365 in Pass Through Method

When the FortiADC devices are configured as AD FS proxy, FortiADC acts as the AD FS proxy between office365 and AD FS server. Client and office365 are both in the external web, while AD FS server is in the internal network. When client visits the office365 service, the request will be redirected to AD FS server to perform the authentication.



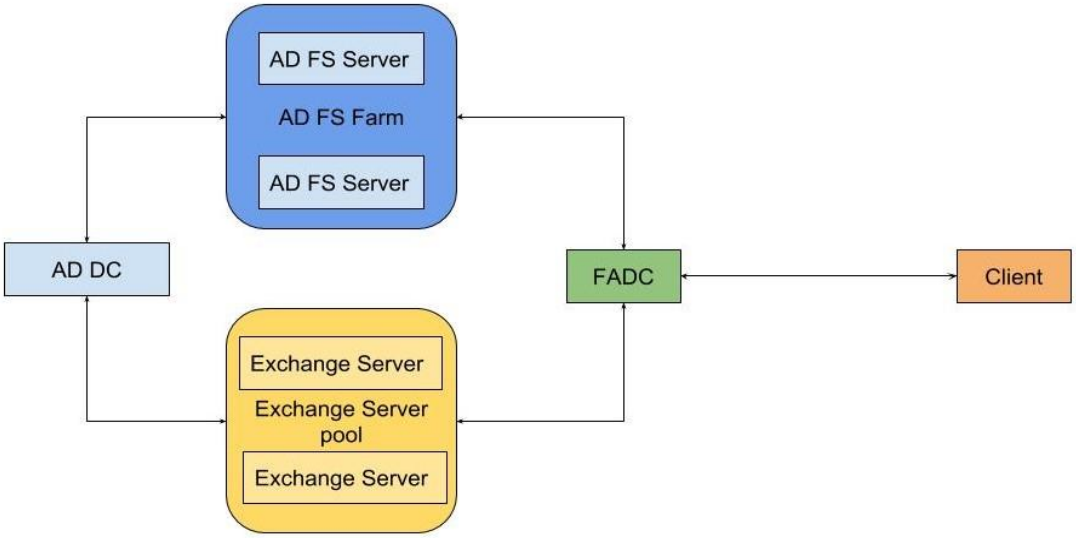
The chart above is the office365 in pass-through mode deployment. Normally, FortiADC receive the request from client to AD FS server, and load balances requests to the internal ADFS server.
The following is the traffic flow for this scenario.

1. Client accesses the Office 365 cloud service;
2. Client is redirected to FADC, FADC delivers the request to an AD FS server in the AD FS Farm;
3. The AD FS server returns a web page and request username/password;
4. Client posts user name and password to AD FS Server;
5. After authentication, AD FS server sets cookie to client;
6. Client accesses AD FS server with cookie;

7. AD FS Server returns SAML token to client;
8. Client accesses office 365 with SAML token.

2.2 Scenario 2: Exchange in Pass Through Method

In this method, exchange server and AD FS server are both in the internal network, and when the client visits the exchange service, no preauthentication is performed by FortiADC; all requests are forwarded to the backend exchange server, and then the exchange server will redirect the request to the AD FS server.

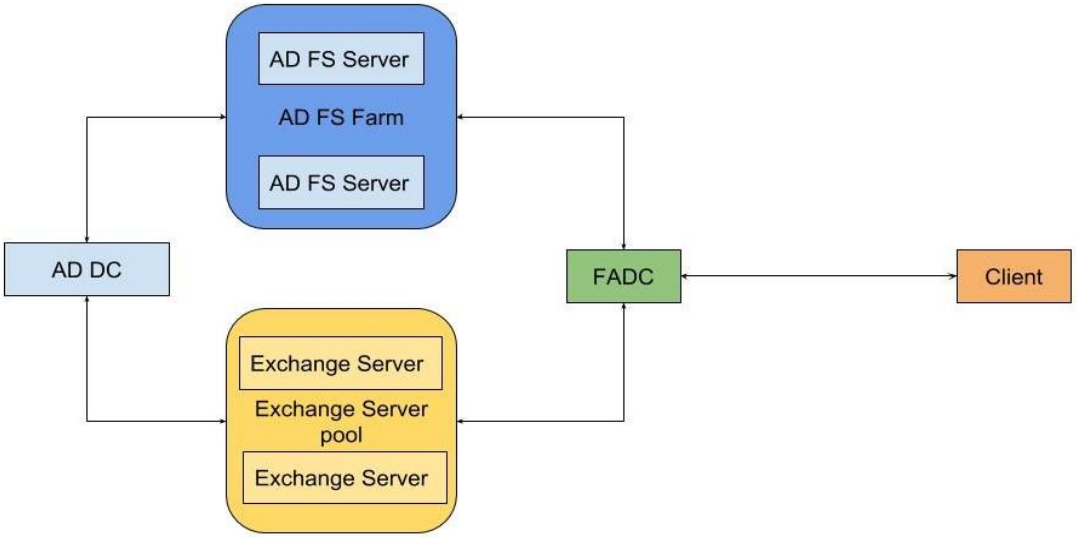


In this scenario, the traffic flow is same as the office365 scenario.
The following is the traffic flow for this scenario:

1. Client sends request to the exchange service (owa or ecp);
2. FADC forwards the request to the backend exchange server directly;
3. The exchange server receives the request and checks “not authenticated,” redirecting it to AD FS server;
4. Client requests that AD FS server perform the authentication. FADC delivers the request to an AD FS server in the AD FS Farm;
5. AD FS server returns a web page and request username/password;
6. Client posts user name and password to AD FS Server;
7. After authentication, the AD FS server sets cookie to client;
8. Client accesses AD FS server with cookie;
9. AD FS Server returns the SAML token to the client;
10. Client accesses exchange service with SAML token.

2.3 Scenario 3: Exchange in AD FS Method

In this method, exchange server and AD FS server are both in the internal network. All unauthenticated client requests are redirected to the federation server. After successful authentication by AD FS, client requests are forwarded to the backend exchange server.



The following is the traffic flow for this scenario.

1. Client sends request to FADC;
2. FADC redirects the request to AD FS Server;
3. AD FS Server sends response to client, and asks for the user name and password;
4. Client posts user name and password to AD FS Server;
5. After authentication, AD FS Server sets cookie to client;
6. AD FS Server redirects client's request to Exchange server with an authToken;
7. Client sends new GET request to Exchange server;
8. Exchange server sets cookie to client, and redirects client to AD FS Server;
9. AD FS Server returns SAML authentication message to client;
10. Client will POST SAML authentication message to Exchange Server;
11. After authentication, Exchange Server sets cookie to client;
12. Client accesses Exchange Server with cookie.

3 AD FS PROXY DEPLOYMENT CONFIGURATION

3.1 Deploy AD FS Proxy for office365

There are two methods to config AD FS Proxy for office365 scenario. The first method is to use AD FS publish service. The second is to use scripting and content routing. Here's how to configure the two methods.

3.1.1 Use AD FS publish service to deploy

1) It is recommended that the virtual server use AD FS publish service when office365 mode is deployed, because when the virtual server uses AD FS publish service, FortiADC will generate a script; in this script, some variables are set according to the AD FS publish service. The customer can also use the scripting and content routing to deploy office365 in chapter 3.1.2

Steps:

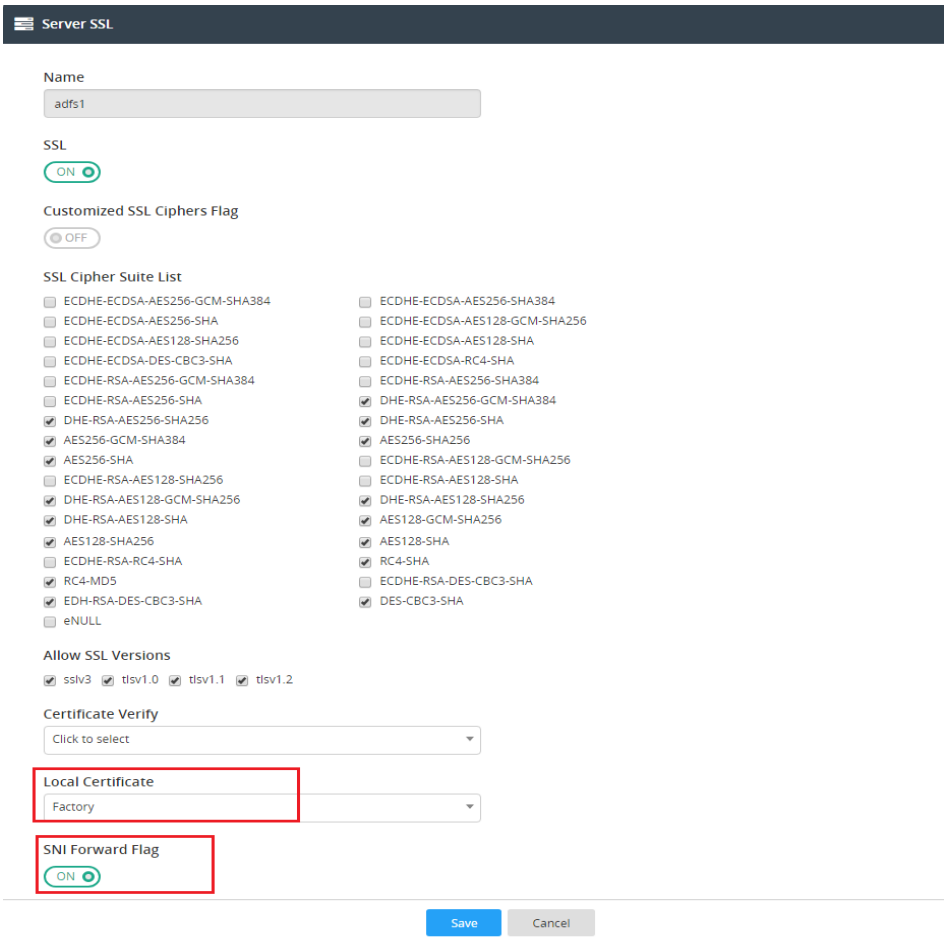
(1) Add AD FS server pool

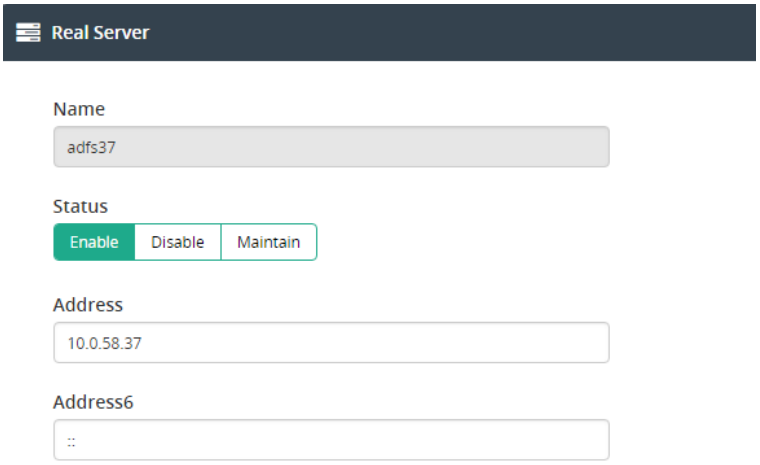
Pay attention:

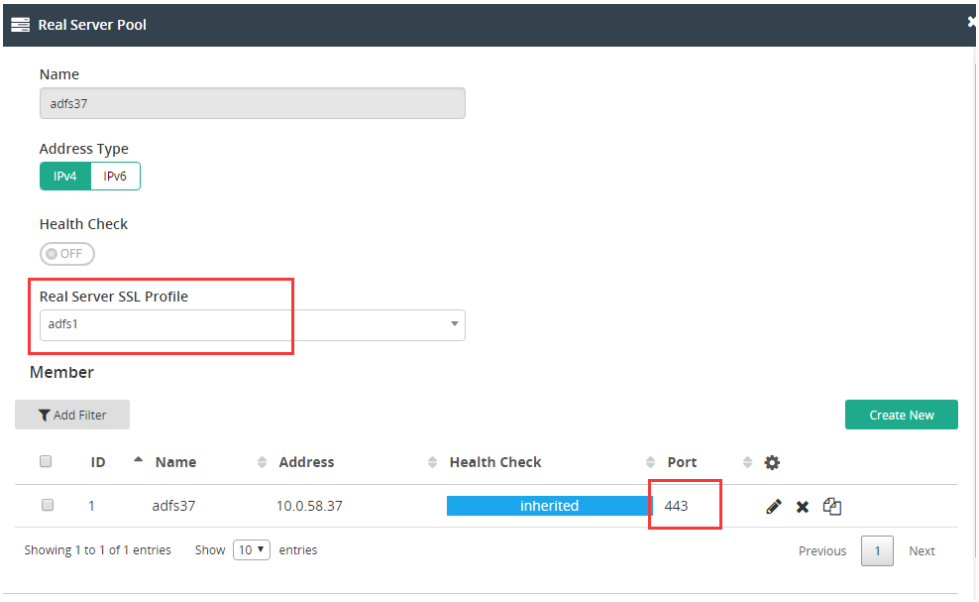
S server uses https (443) to connect. The AD FS server pool must use the 443 port; in order to make it work, it must set the real-server-ssl-profile. For real-server-ssl-profile, a local cert must be used, and the ssl-sni-forward must be set.

```
config load-balance real-server-ssl-profile
  edit "adfs1"
    set ssl enable
    set ssl-sni-forward enable
```

```
    set local-cert Factory
  next
end
config load-balance real-server
  edit "adfs37"
    set ip 10.0.58.37
  next
end
config load-balance pool
  edit "adfs37"
    set real-server-ssl-profile adfs1
    config  pool_member
      edit 1
        set pool_member_service_port 443
        set pool_member_cookie rs1
        set real-server adfs37
      next
    end
  next
end
```



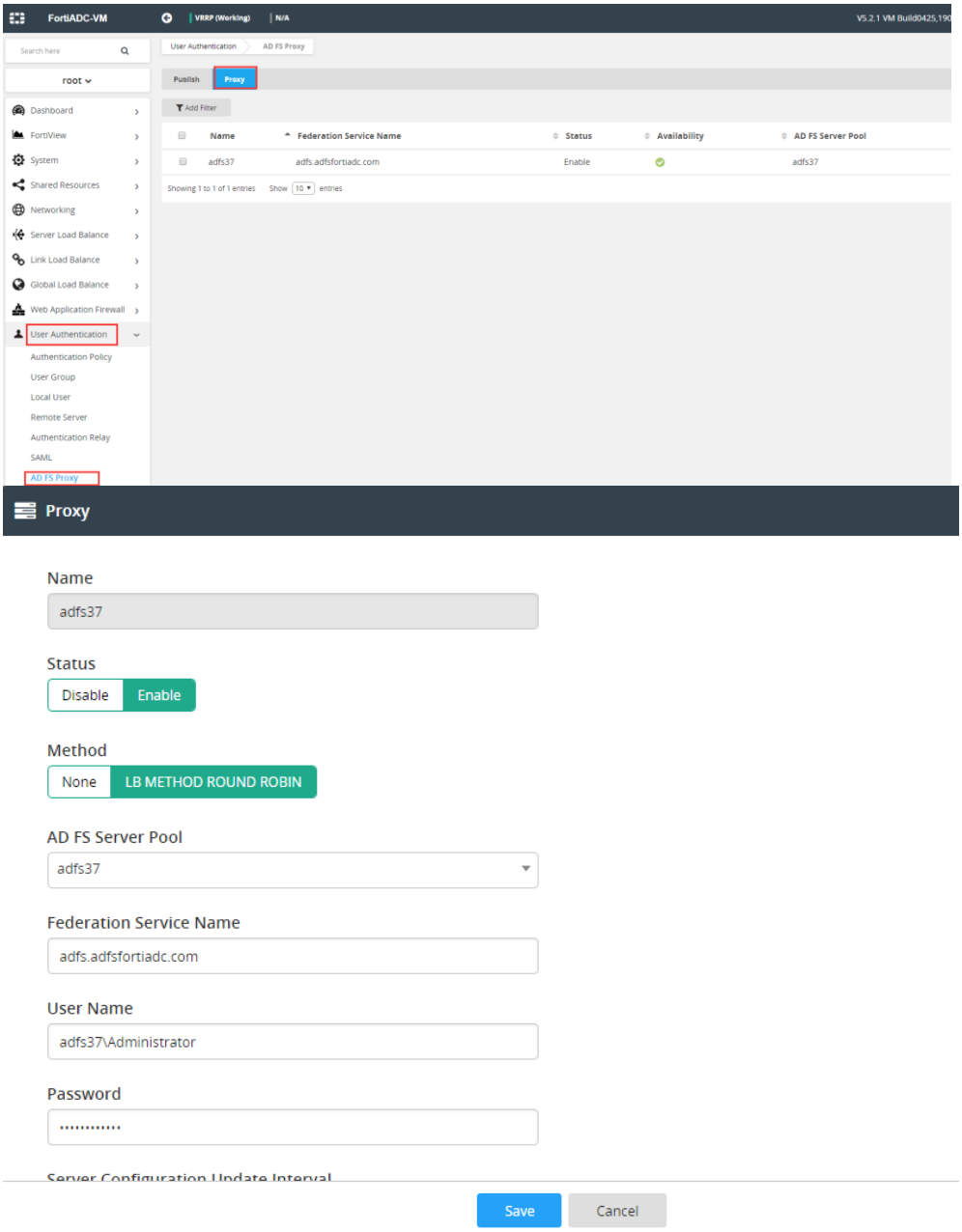




(2) Add AD FS proxy

FortiADC adds an adfs-proxy for registering to AD FS server. The configuration should be set according to AD FS server; the fqdn is the same as AD FS federation service; the username should be the local administrator account on the AD FS server.

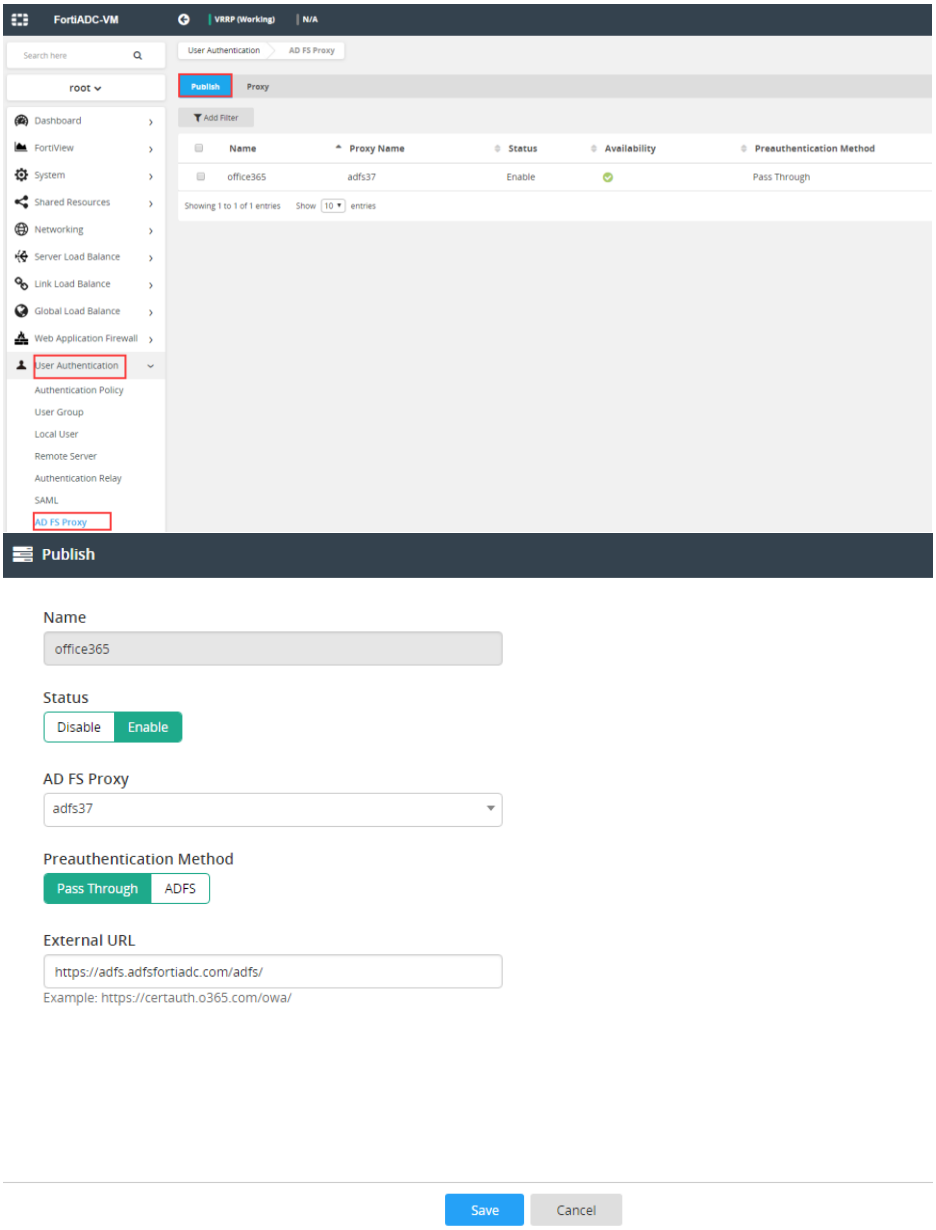
```
config user adfs-proxy
  edit "o365"
    set fqdn adfs.adfsfortiadc.com
    set load-balance-method LB_METHOD_ROUND_ROBIN
    set load-balance-pool adfs37
    set username "adfs37\\Administrator"
    set password ENC
skjgso1KvGVRmFh7RKUJkArz+X65udG2wAB8TajucCFOCbQe+zkNoH2eA+gL4Uv8yxnNzoG2Iq/Il
4qGv4L/nLdQhtj26FsgdvsoxoWSvu8x+Al1
  next
end
```



(3) Add AD FS publish

In office365 scenario, the method uses pass-through. The external-url should be same as it is in AD FS server.

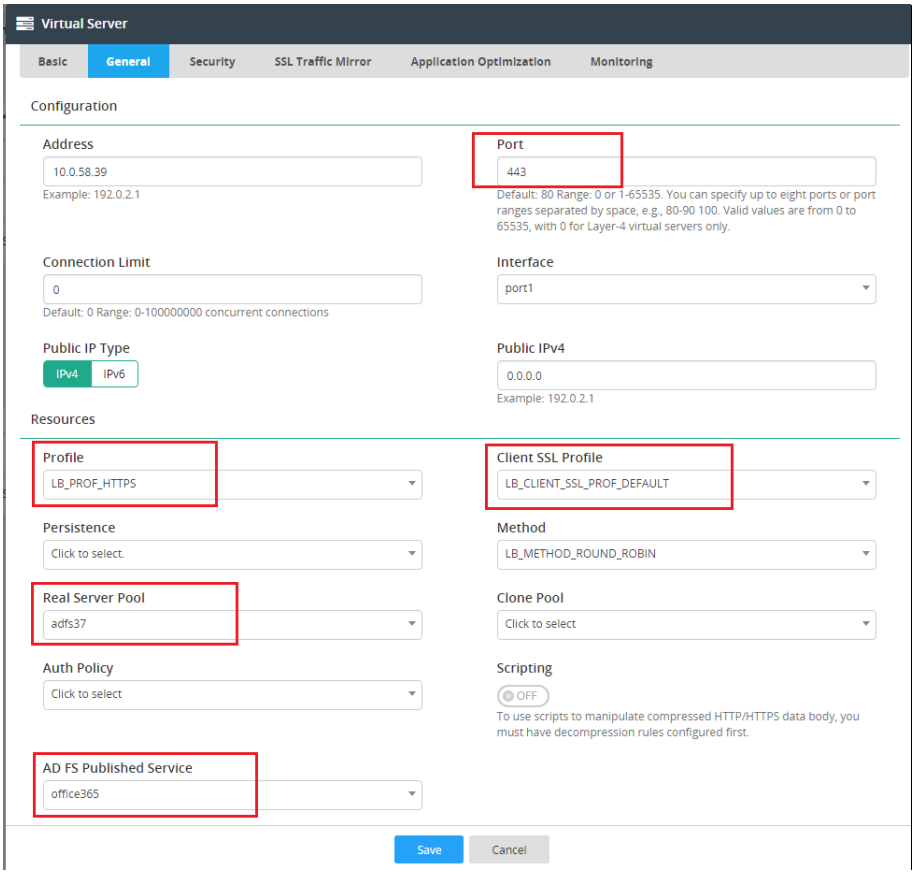
```
config user adfs-publish
  edit "o365"
    set adfs-proxy o365
    set external-url https://adfs.adfsfortiadc.com/adfs/
  next
end
```



(4) Set AD FS publish service to virtual server

As AD FS server uses the https(443) connection, the office365 virtual server must configure LB_PROF_HTTPS profile and the port must use 443.

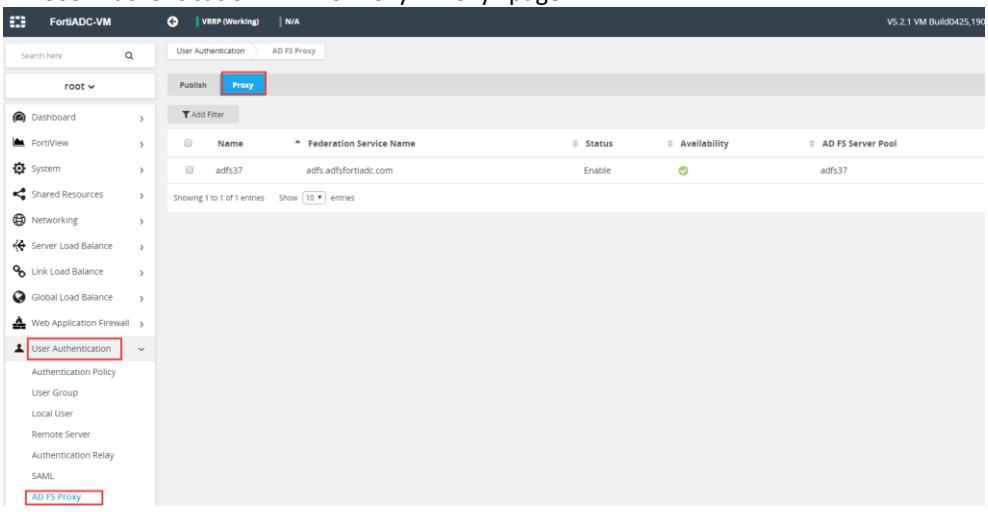
```
config load-balance virtual-server
 edit "o365"
  set type l7-load-balance
  set interface port1
  set ip 10.0.58.39
  set port 443
  set load-balance-profile LB_PROF_HTTPS
  set client-ssl-profile LB_CLIENT_SSL_PROF_DEFAULT
  set load-balance-method LB_METHOD_ROUND_ROBIN
  set load-balance-pool adfs37
  set traffic-group default
  set adfs-published-service office365
 next
end
```



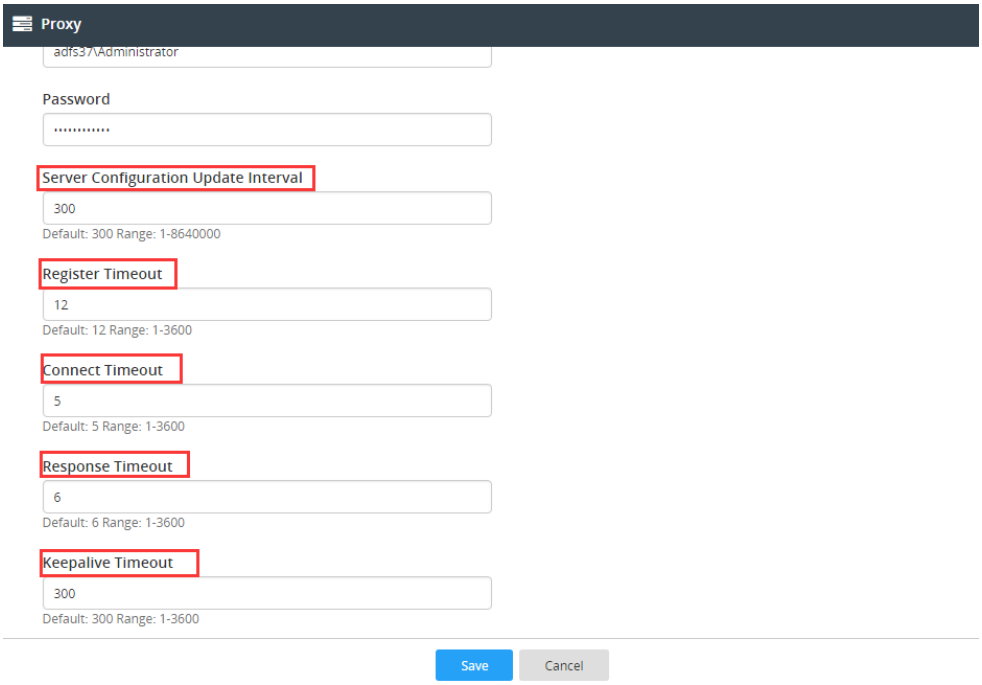
2) Configure adfs proxy advance option

As AD FS proxy will register to AD FS server, in the register connection, customer can configure some timeout to adapt to the AD FS server.

In “User Authentication->AD FS Proxy->Proxy” page:

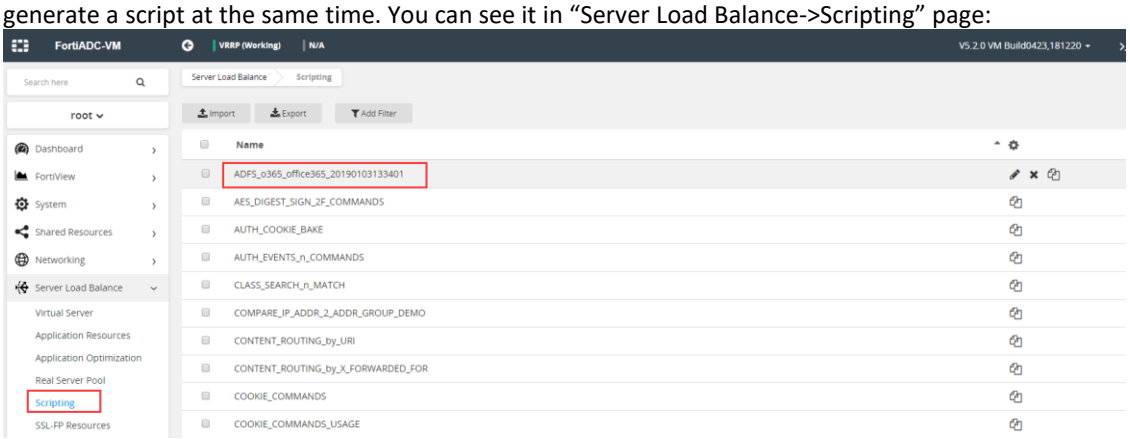


Open the proxy and configure the timeout as customer needed:



3.1.2 Configure office365 scenario using scripting

As configuration in 3.1.1, when the virtual server uses the AD FS publish service, FortiADC will generate a script at the same time. You can see it in “Server Load Balance->Scripting” page:



The script is named ADFS_VIRTUAL SERVERNAME_PUBLISHNAME_timestamp. This script defines the action that office365 scenario requires. When the virtual server unsets AD FS publish service, FortiADC will delete this script at the same time.

Scripting

Name

ADFS_o365_office365_20190103133401

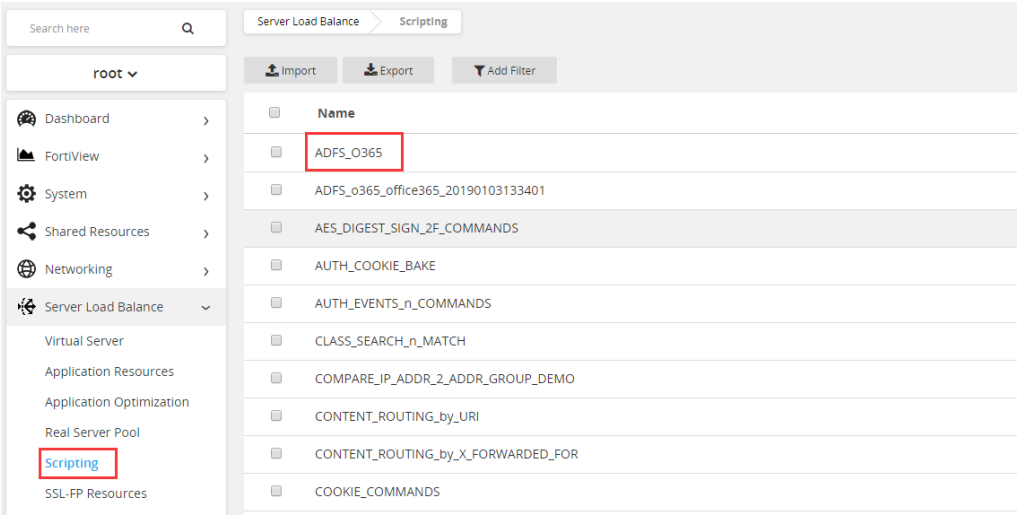
```
when RULE_INIT{
    --should be replaced according to the publish settings
    pub_uri ="/adfs/"
    external_host = "adfs.adfsfortiadc.com"
}

when HTTP_REQUEST{
    uri = HTTP:uri_get()
    host = HTTP:header_get_value("Host")
    path = HTTP:path_get()

    if host:find(external_host) and uri:starts_with(pub_uri) then
        --insert ADFS header
        --HTTP:header_insert(header_name, value)
        cip = IP:client_addr()  -- get client ip address
        HTTP:header_insert("X-MS-Endpoint-Absolute-Path", path)
        HTTP:header_insert("X-MS-Forwarded-Client-IP", cip)
        HTTP:header_insert("X-MS-Proxy", "FortiADC")
    else
        HTTP:close()
        debug("no matches domain for host: %s and uri %s\n", host, uri)
    end
}
```

Save Cancel

Customer can use scripting instead of AD FS publish service.

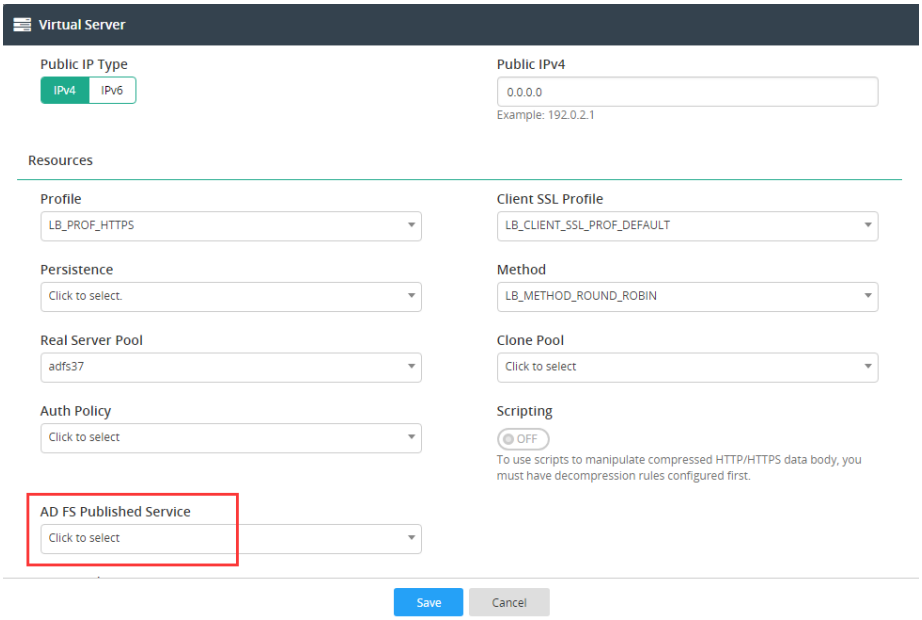
Steps:

(1) Copy this script.



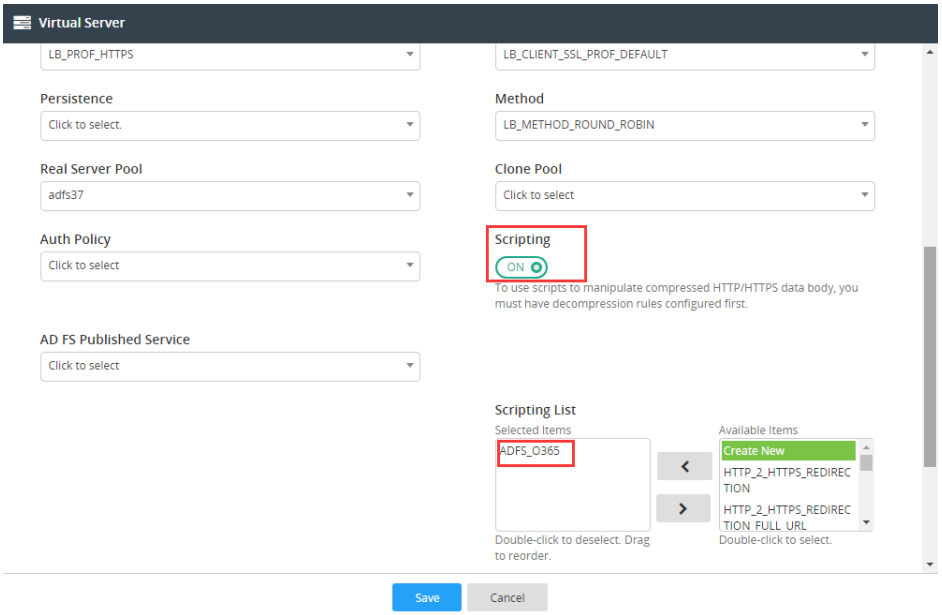
(2) In virtual server, unset AD FS publish service

```
config load-balance virtual-server
  edit "o365"
    unset adfs-published-service
  next
end
```



(3) Set script copied in (1) to virtual server

```
config load-balance virtual-server
  edit "o365"
    set type l7-load-balance
    set interface port1
    set ip 10.0.58.39
    set port 443
    set load-balance-profile LB_PROF_HTTPS
    set client-ssl-profile LB_CLIENT_SSL_PROF_DEFAULT
    set load-balance-method LB_METHOD_ROUND_ROBIN
    set load-balance-pool adfs37
    set scripting-flag enable
    set scripting-list ADFS_o365
    set traffic-group default
  next
end
```



Note:

(1) When configure AD FS publish service to virtual server, FortiADC generates a script. This script will be deleted by FortiADC when virtual server unsets AD FS publish service.

(2) As AD FS publish uses the adfs federation service, the client should configure the mapping between adfs federation service and virtual server ip address, such as:
adfs.adfsfortiadc.com 10.0.58.39 (10.0.58.39 is virtual server ip)
When client requests office365 service, like https://portal.microsoft.com, the request will be redirected to FADC(10.0.58.39) to perform authentication.

(3) AD FS publish cannot configure disabled AD FS proxy

(4) Virtual server cannot configure disabled AD FS publish

3.2 Deploy AD FS Proxy for Exchange in pass through mode

In this scenario, as the AD FS server and exchange server are both in the internal network, FortiADC should add two pools for AD FS server and exchange server.

3.2.1 use AD FS publish service to deploy

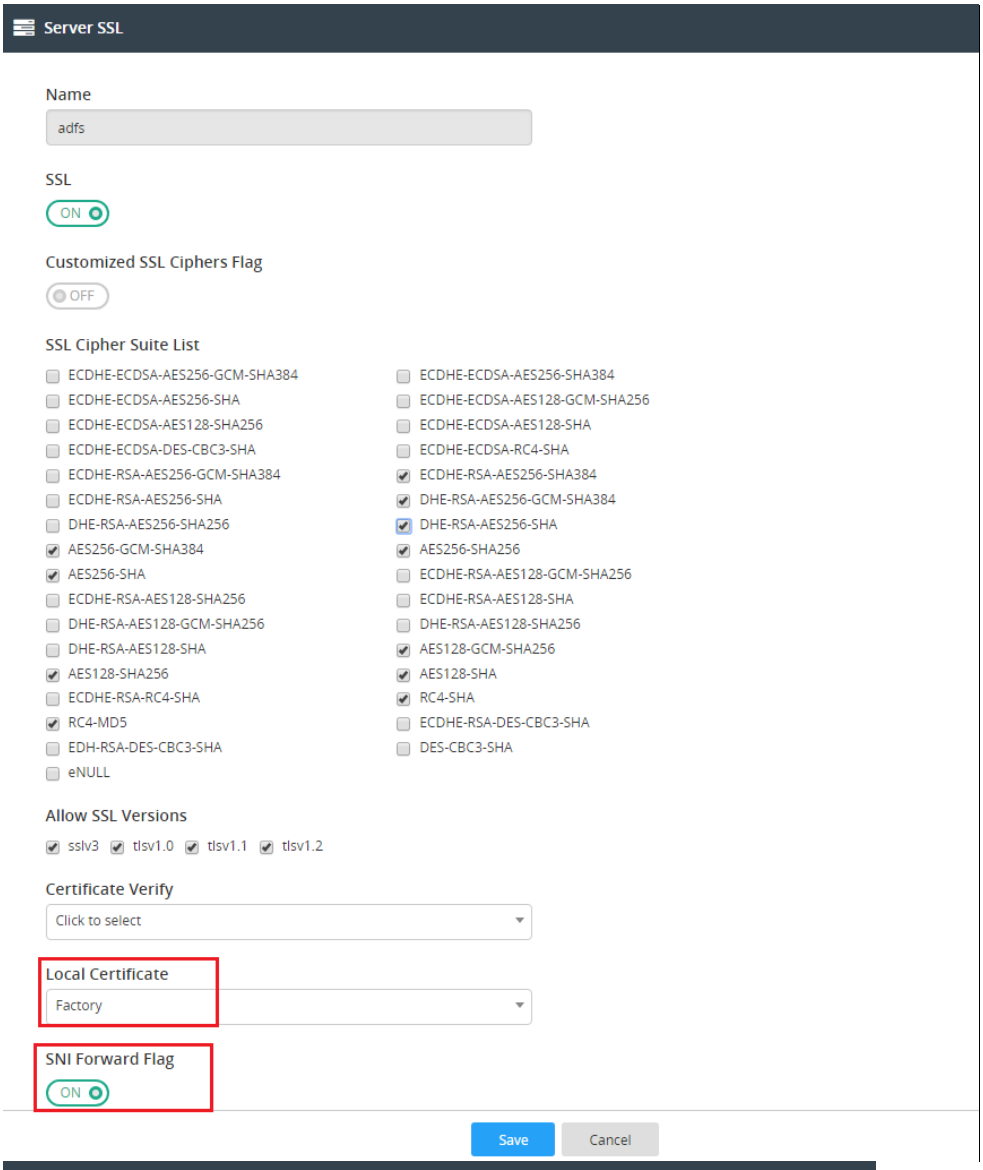
1) config Steps:

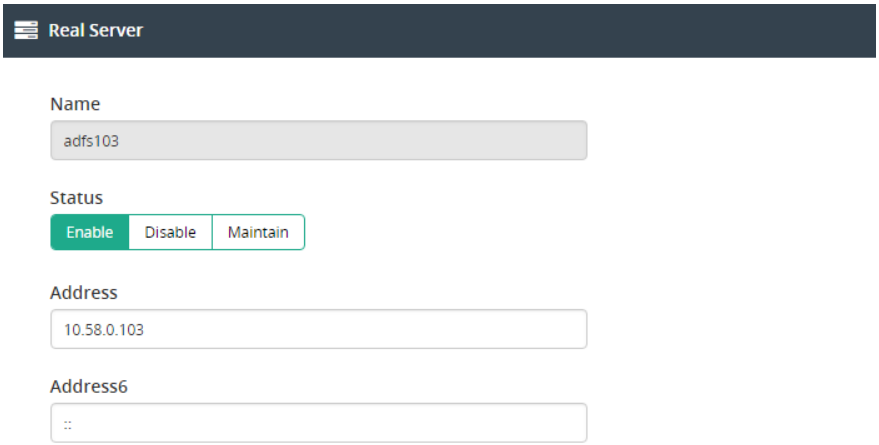
(1) Add AD FS server pool

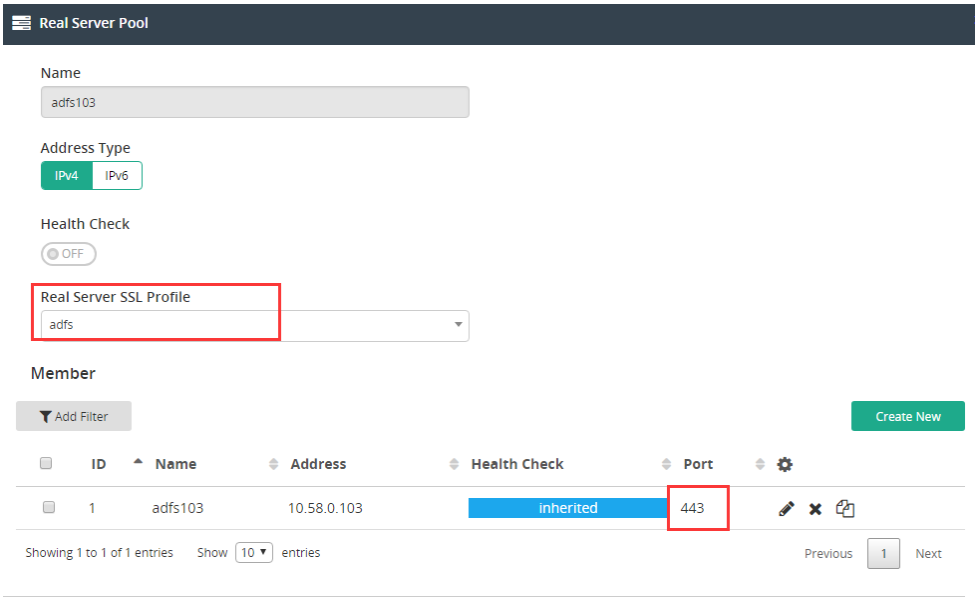
Pay attention:

As AD FS server uses https(443) to connect, the AD FS server pool must use 443 port and set real-server-ssl-profile. In real-server-ssl-profile, a local cert must be used, and the ssl-sni-forward must be set.

```
config load-balance real-server-ssl-profile
 edit "adfs"
  set ssl enable
  set ssl-sni-forward enable
  set local-cert Factory
 next
end
config load-balance real-server
 edit "adfs103"
  set ip 10.58.0.103
 next
end
config load-balance pool
 edit "adfs103"
  set real-server-ssl-profile adfs
  config  pool_member
   edit 1
    set pool_member_service_port 443
    set pool_member_cookie rs1
    set real-server adfs103
   next
  end
 next
end
```


Server SSL
Name
adfs
SSL
ON
Customized SSL Ciphers Flag
OFF
SSL Cipher Suite List
ECDHE-ECDSA-AES256-GCM-SHA384
ECDHE-ECDSA-AES256-SHA
ECDHE-ECDSA-AES128-SHA256
ECDHE-ECDSA-DES-CBC3-SHA
ECDHE-RSA-AES256-GCM-SHA384
ECDHE-RSA-AES256-SHA
DHE-RSA-AES256-SHA256
AES256-GCM-SHA384
AES256-SHA
ECDHE-RSA-AES128-SHA256
DHE-RSA-AES128-GCM-SHA256
DHE-RSA-AES128-SHA
AES128-SHA256
ECDHE-RSA-RC4-SHA
RC4-MD5
EDH-RSA-DES-CBC3-SHA
eNULL
ECDHE-ECDSA-AES256-SHA384
ECDHE-ECDSA-AES128-GCM-SHA256
ECDHE-ECDSA-AES128-SHA
ECDHE-ECDSA-RC4-SHA
ECDHE-RSA-AES256-SHA384
DHE-RSA-AES256-GCM-SHA384
DHE-RSA-AES256-SHA
AES256-SHA256
ECDHE-RSA-AES128-GCM-SHA256
ECDHE-RSA-AES128-SHA
DHE-RSA-AES128-SHA256
AES128-GCM-SHA256
AES128-SHA
RC4-SHA
ECDHE-RSA-DES-CBC3-SHA
DES-CBC3-SHA
Allow SSL Versions
sslv3
tlsv1.0
tlsv1.1
tlsv1.2
Certificate Verify
Click to select
Local Certificate
Factory
SNI Forward Flag
ON
Save
Cancel

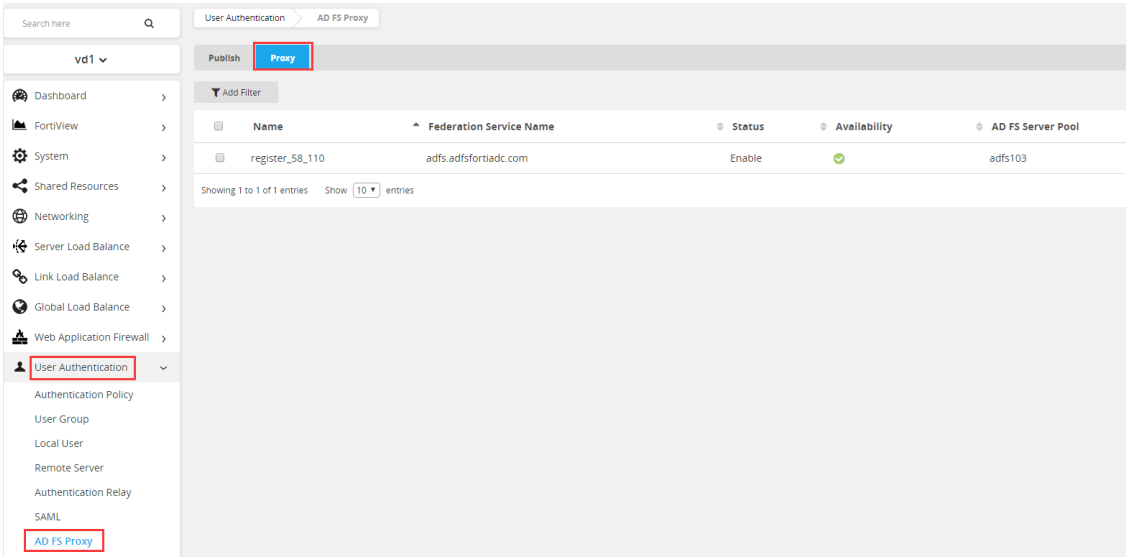

Real Server
Name
adfs103
Status
Enable
Disable
Maintain
Address
10.58.0.103
Address6
::

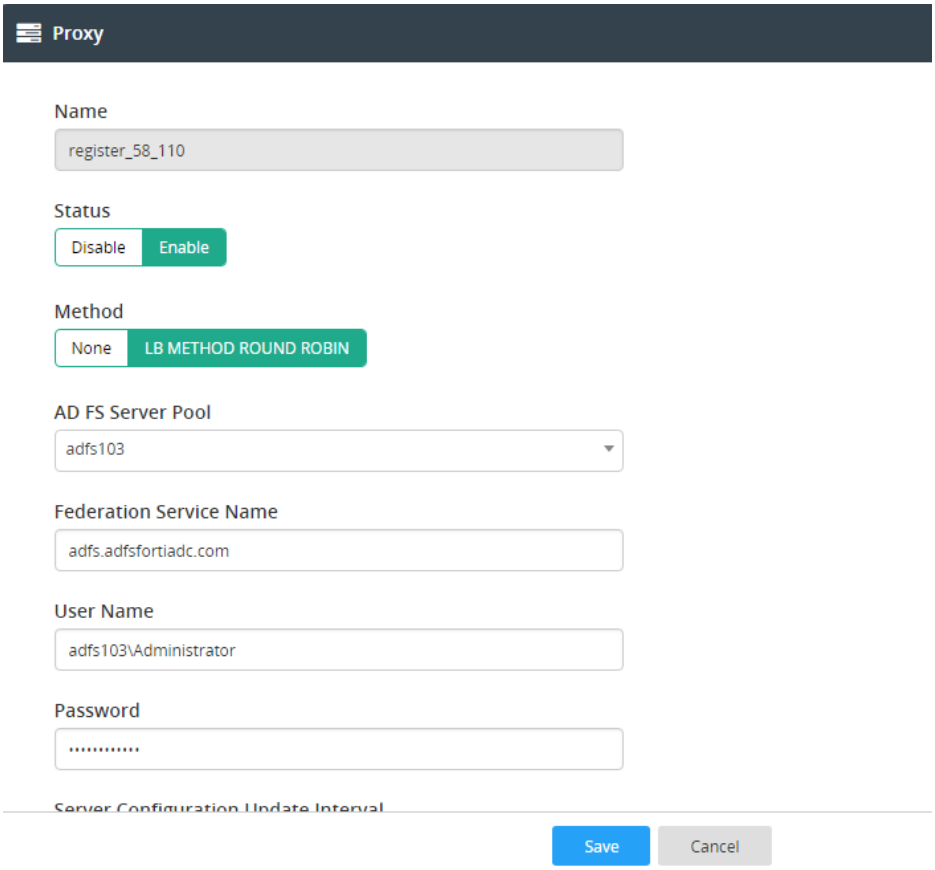


(2) Add AD FS proxy

FortiADC adds an adfs-proxy for registering to AD FS server. So, the configuration should be set according to AD FS server; the fqdn is the same as it is for AD FS federation service; the username should be the local administrator account on the AD FS server.

```
config user adfs-proxy
    edit "register_58_110"
      set fqdn adfs.adfsfortiadc.com
      set load-balance-method LB_METHOD_ROUND_ROBIN
      set load-balance-pool adfs103
      set username "adfs103\\Administrator"
      set password ENC
skjgso1KvGVRmFh7RKUJkArz+X65udG2wAB8TajucCFOCbQe+zkNoH2eA+gL4Uv8yxnNzoG2Iq/Il
4qGv4L/nLdQhtj26FsgdvsoxoWSvu8x+AI1
    next
end
```

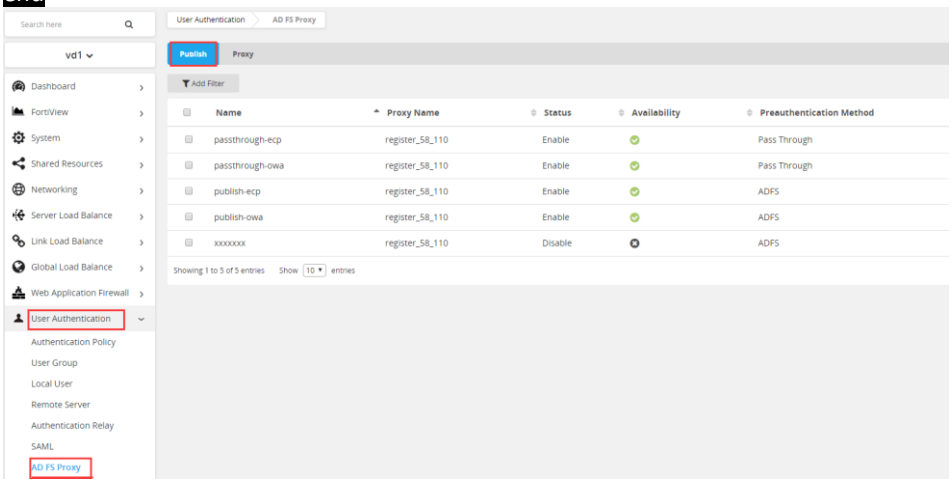


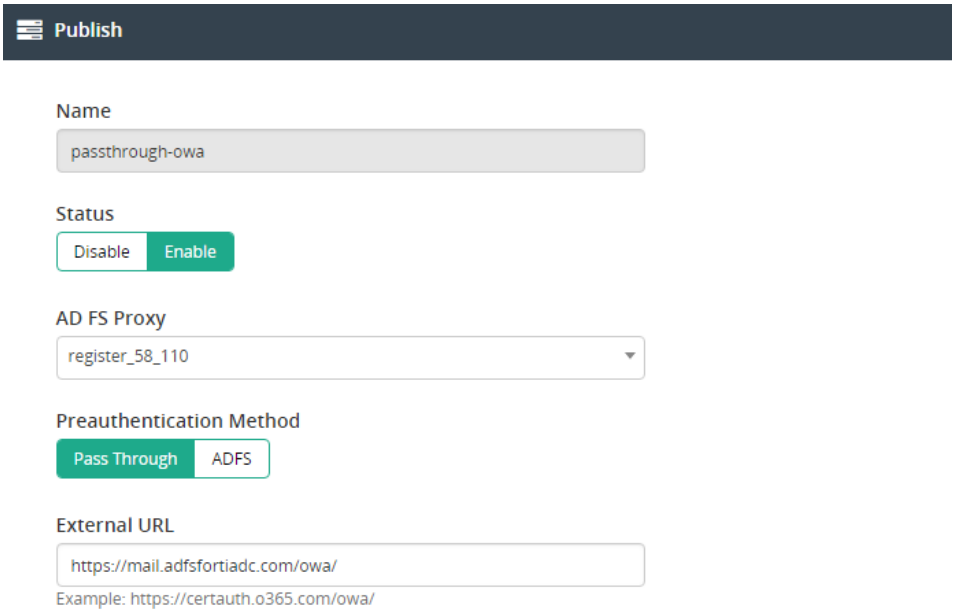


(3) Add AD FS publish

In this scenario, the method uses pass-through.

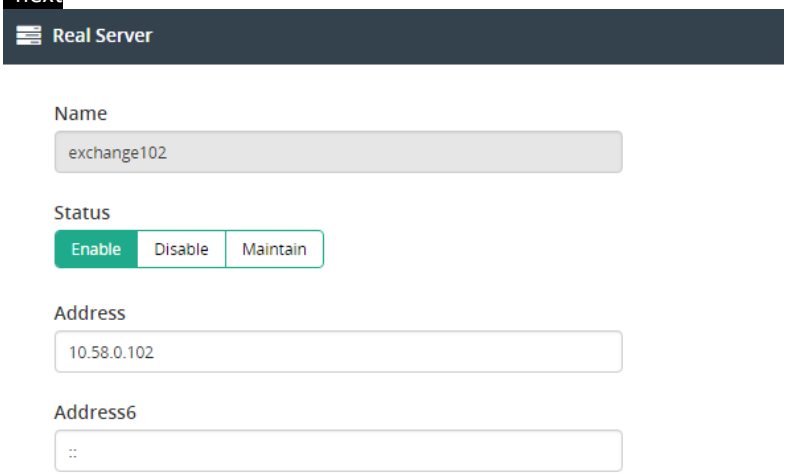
```
config user adfs-publish
  edit "passthrough-owa"
    set status enable
    set adfs-proxy register_58_110
    set pre-auth Pass-through
    set external-url https://mail.adfsfortiadc.com/owa/
  next
end
```

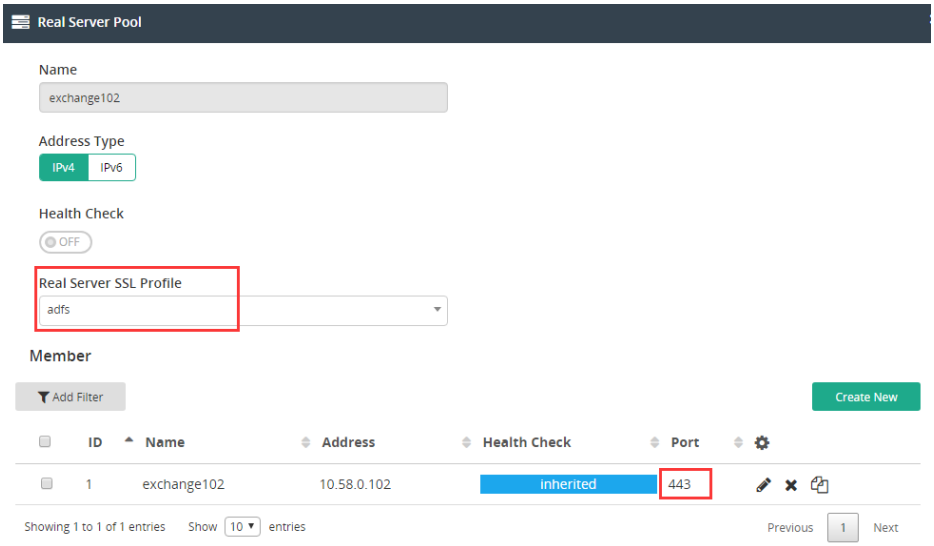




(4) Add exchange server pool

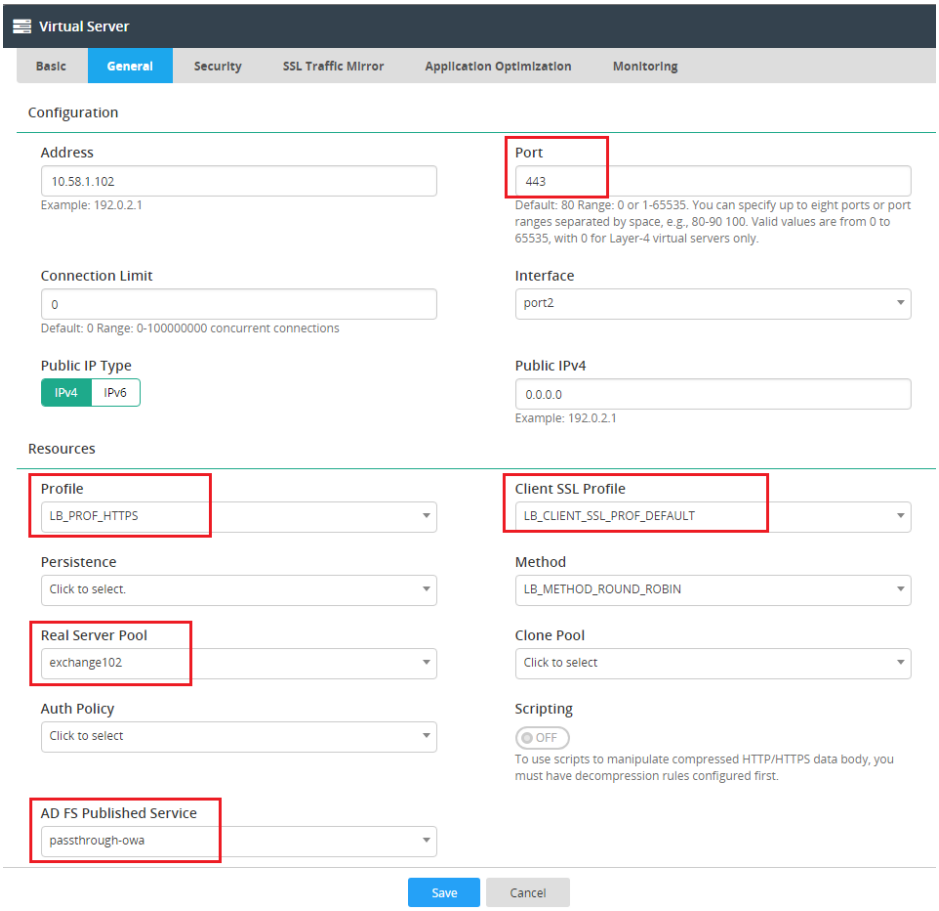
```
config load-balance real-server
 edit "exchange102"
  set ip 10.58.0.102
 next
end
config load-balance pool
 edit "exchange102"
  set real-server-ssl-profile adfs
  config  pool_member
   edit 1
    set pool_member_service_port 443
    set pool_member_cookie rs1
    set real-server exchange102
   next
  end
 next
```





(5) Add virtual server, set AD FS publish service to virtual server

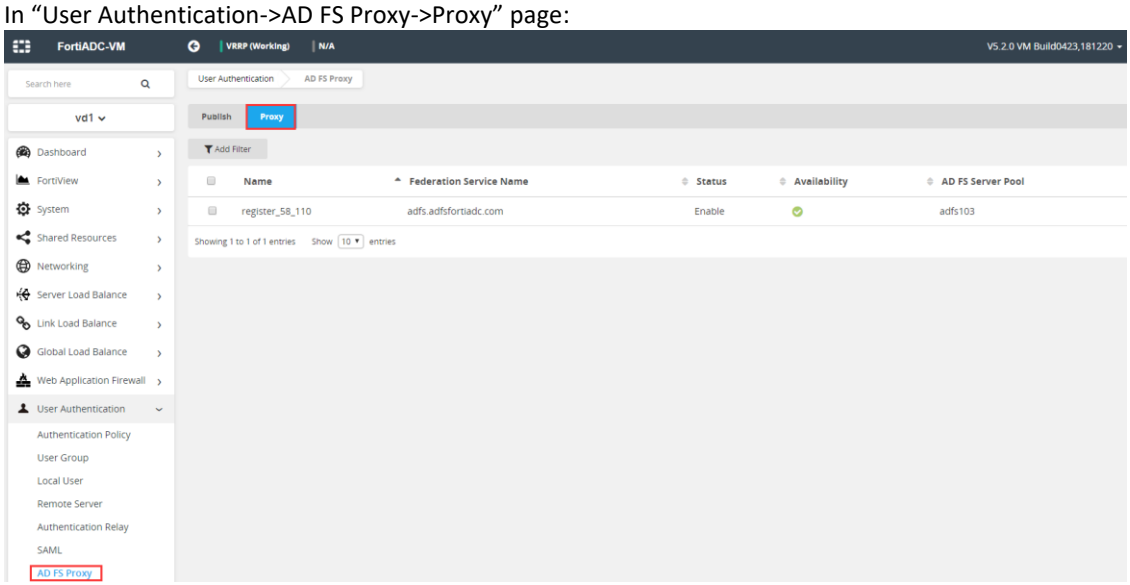
```
endconfig load-balance virtual-server
 edit "passthrough_owa"
  set type l7-load-balance
  set interface port2
  set ip 10.58.1.102
  set port 443
  set load-balance-profile LB_PROF_HTTPS
  set client-ssl-profile LB_CLIENT_SSL_PROF_DEFAULT
  set load-balance-method LB_METHOD_ROUND_ROBIN
  set load-balance-pool exchange102
  set traffic-group default
  set adfs-published-service passthrough-owa
 next
end
```



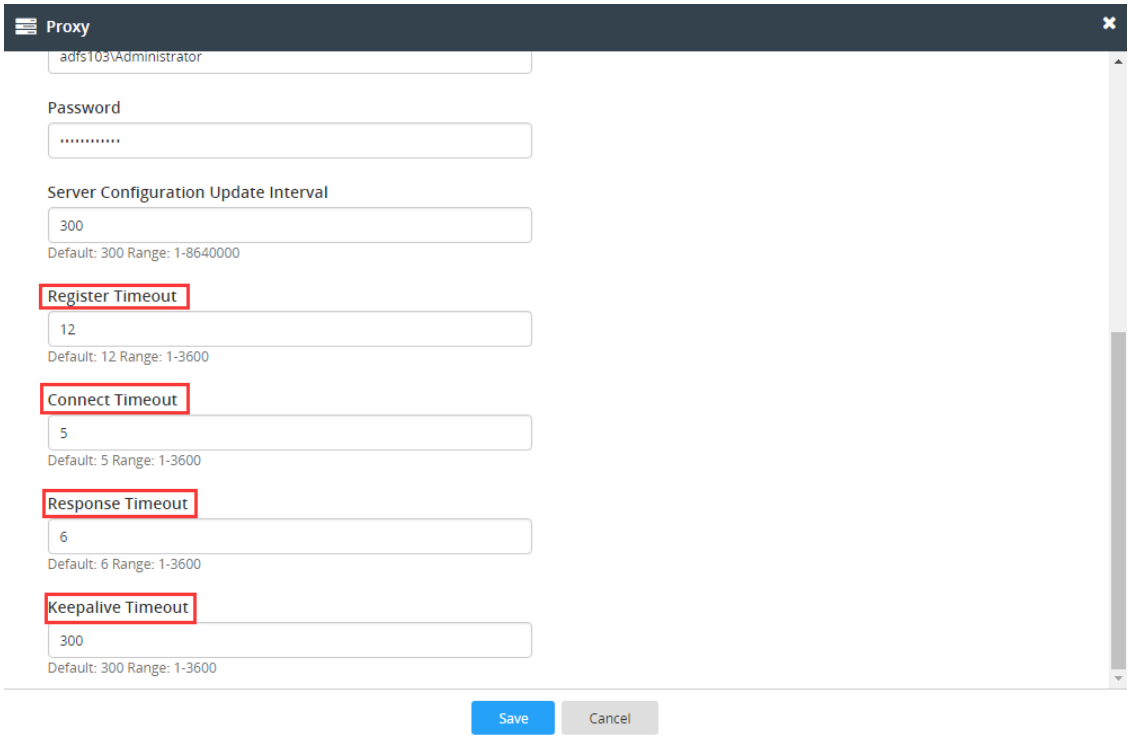
2) Configure adfs proxy advance option

AD FS proxy will register to the AD FS server. In the register connection, the customer can configure the timeouts to adapt to the AD FS server.

In “User Authentication->AD FS Proxy->Proxy” page:



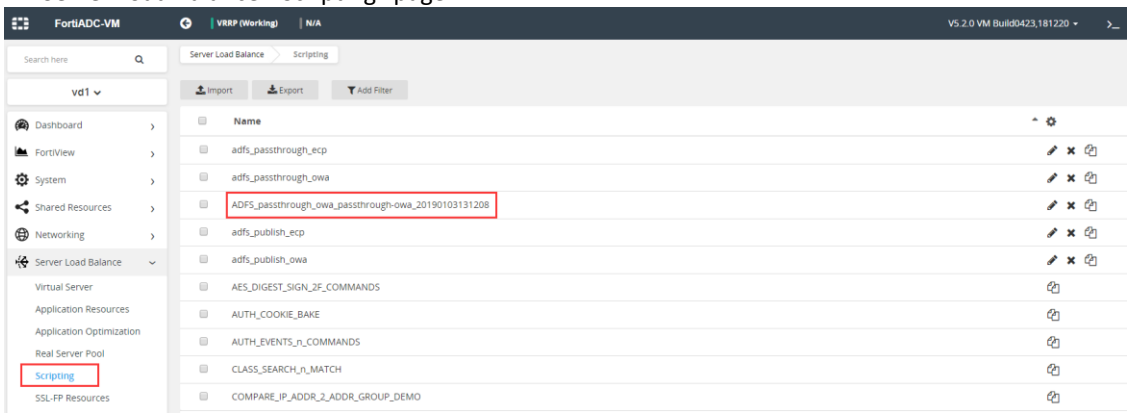
Open the proxy and configure the timeout as customer needed:



3.2.2 Configure scripting and content routing

As shown in 3.2.1, when configuring AD FS publish service to the virtual server, a script is generated automatically. Customer can use scripting instead of AD FS publish service.

In “Server Load Balance->Scripting” page:



Opening this script, you can get the action that FortiADC will then operate. In this script, FortiADC will forward requests to different backend server pools, by using content routing. So the customer should add content routings that as the same as those used in scripting.

Scripting

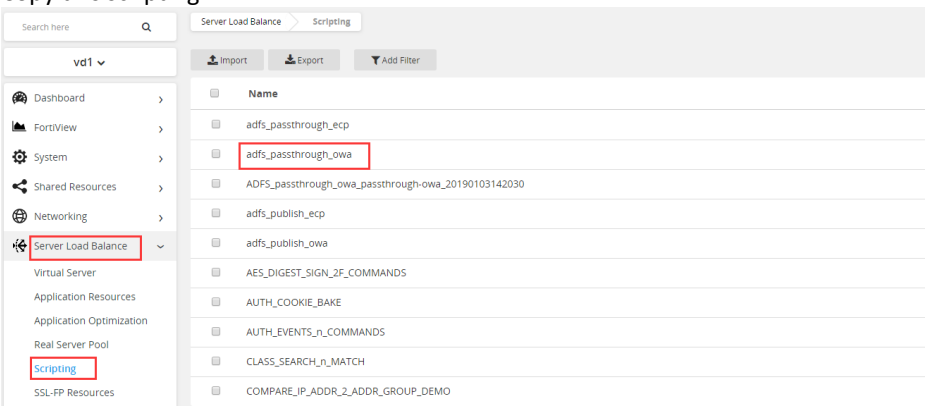
Name

ADFS_passthrough_owa_passthrough-owa_20190103131208

```
when RULE_INIT{
    pub_uri ="/owa/"
    external_host = "mail.adfsfortiadc.com"
    adfs_server_domain = "adfs.adfsfortiadc.com"
}

when HTTP_REQUEST{
    uri = HTTP:uri_get()
    host = HTTP:header_get_value("Host")
    path = HTTP:path_get()

    --content-routing
    --"msapp" "adfsserver" should have created in content-routing module
    if host:find(external_host) and uri:starts_with(pub_uri) then
        LB:routing("exchange102")
    elseif host:find(adfs_server_domain) then
        LB:routing("adfs_103")
    else
      debug("no matches domain for host: %s and uri %s\n", host, uri)
      HTTP:close()
    end
}
```

Customer can use scripting and content routing instead of AD FS publish service.
Steps:

(1) Copy this scripting.



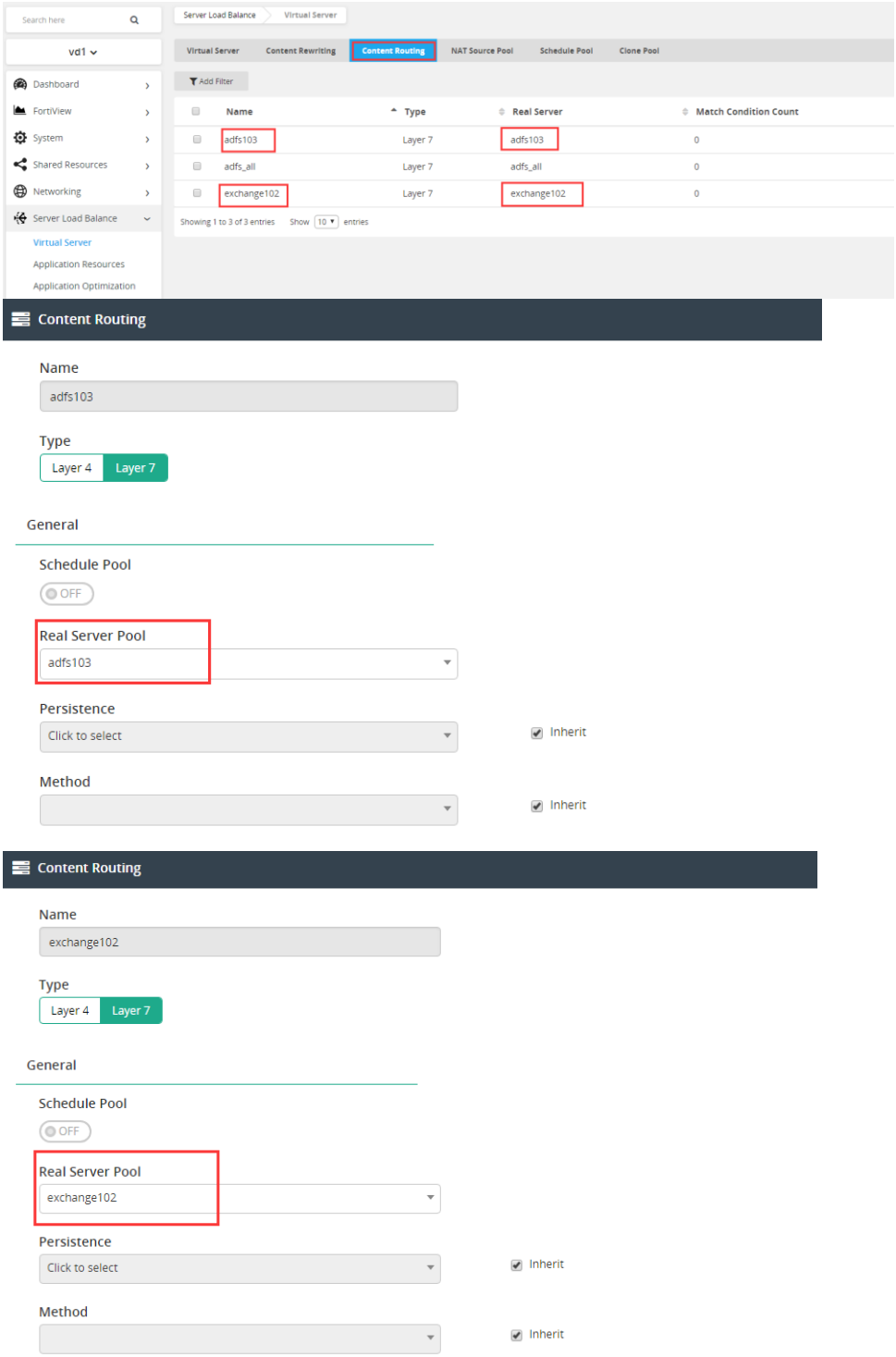
(2) In virtual server, unset AD FS publish service

```
config load-balance virtual-server
 edit "passthrough_owa"
  unset adfs-published-service
 next
end
```

(3) Add content routing

It is the same as that used in the scripting, like in the previous example, content routing “exchange102” using real-server-pool “exchange102”, and content routing “adfs103” using real-server-pool “adfs103”

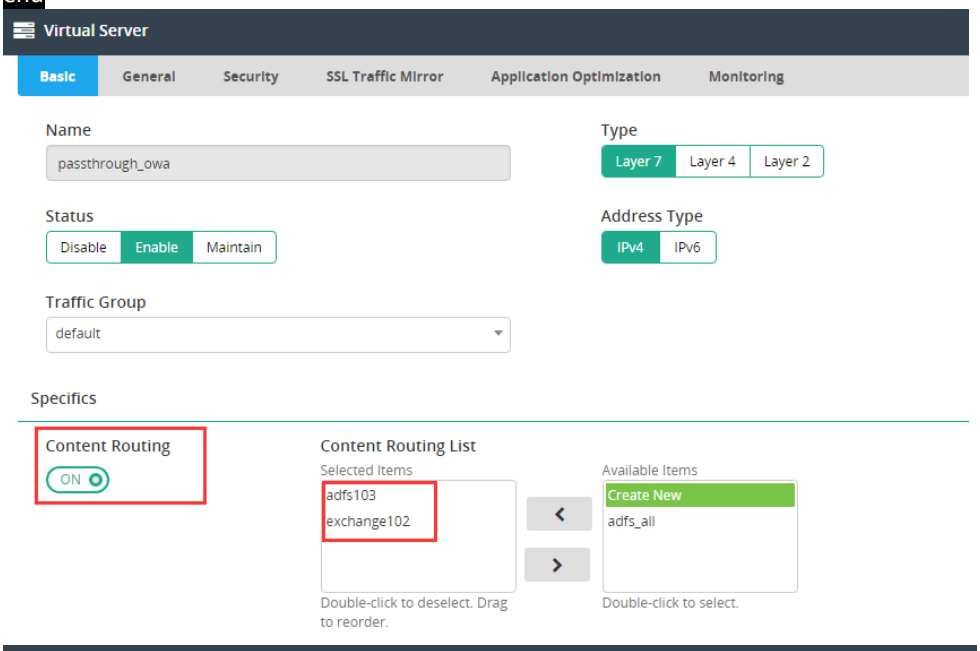
```
config load-balance content-routing
 edit "exchange102"
  set load-balance-pool exchange102
  config  match-condition
  end
 next
 edit "adfs103"
  set load-balance-pool adfs103
  config  match-condition
  end
 next
end
```

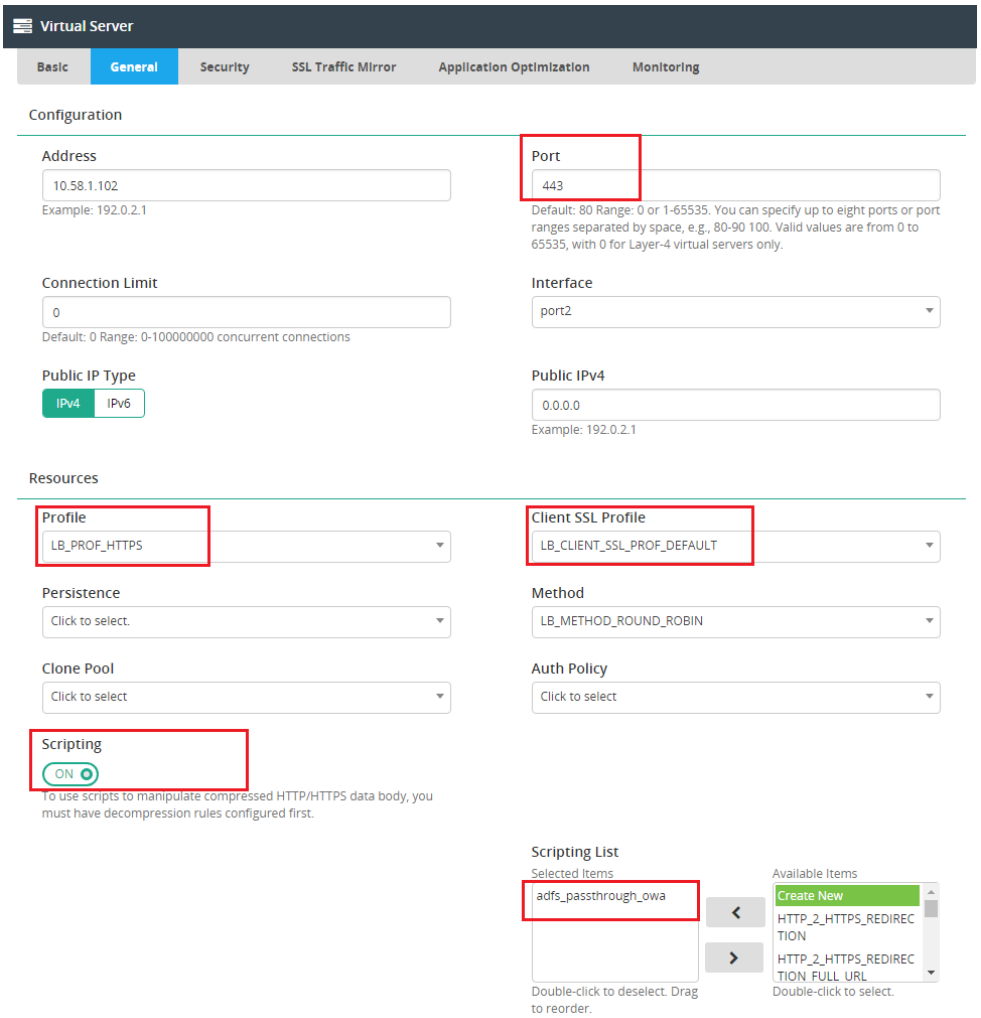


(4) For scripting copied in step 1 and content routing in step 3, set them to the virtual server.

```
config load-balance virtual-server
 edit "passthrough_owa"
  set type l7-load-balance
  set interface port2
  set ip 10.58.1.102
  set port 443
  set load-balance-profile LB_PROF_HTTPS
  set client-ssl-profile LB_CLIENT_SSL_PROF_DEFAULT
  set content-routing enable
  set content-routing-list adfs103 exchange102
  set load-balance-method LB_METHOD_ROUND_ROBIN
  set scripting-flag enable
```

```
    set scripting-list ADFS_passthrough_owa
    set traffic-group default
  next
end
```





Note:

(1) Upon configuring AD FS publish service to virtual server, FortiADC generates a script. This script will be deleted by FortiADC when virtual server unsets AD FS publish service.

(2) Since AD FS publish uses adfs federation service, the client should configure the mapping between adfs federation service and VIRTUAL SERVER ip address, such as:
adfs.adfsfortiadc.com 10.58.1.102(virtual server ip)
mail.adfsfortiadc.com 10.58.1.102
Then client requests exchange service: https://mail.adfsfortiadc.com/owa/
(3) AD FS publish cannot configure disabled AD FS proxy
(4) Virtual server cannot configure disabled AD FS publish

3.3 Deploy AD FS Proxy for Exchange in ADFS mode

3.3.1 use AD FS publish service to deploy

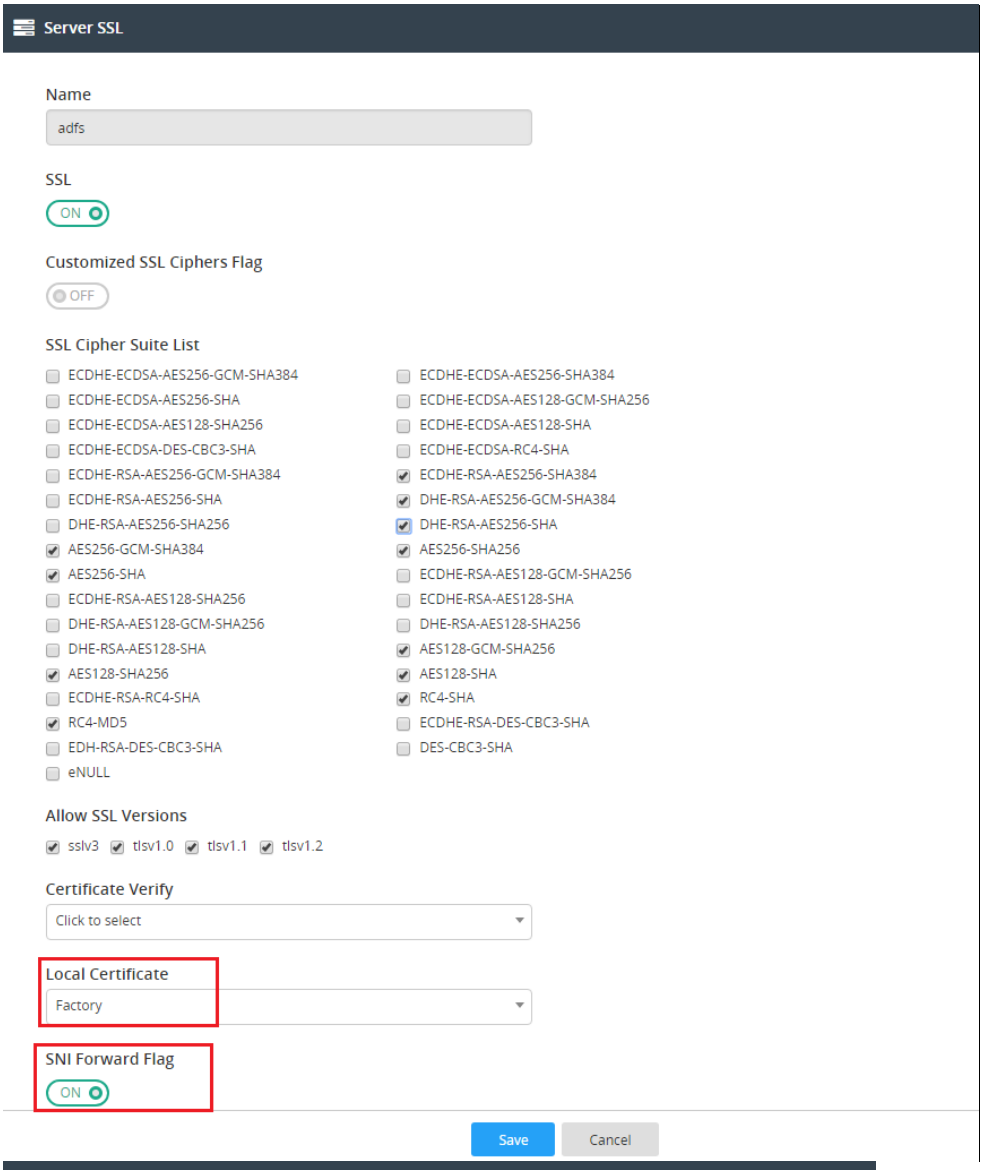
It is recommended that the virtual server use AD FS publish service when exchange mode is deployed. When virtual server uses AD FS publish service, FortiADC will generate a script; in this script, some variables are set according to the AD FS publish service. The customer can also use scripting and content routing to deploy exchange-ADFS.

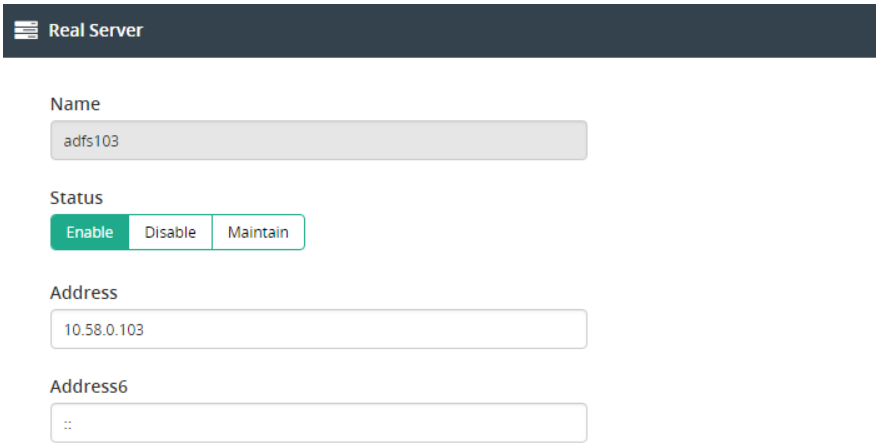
Steps:

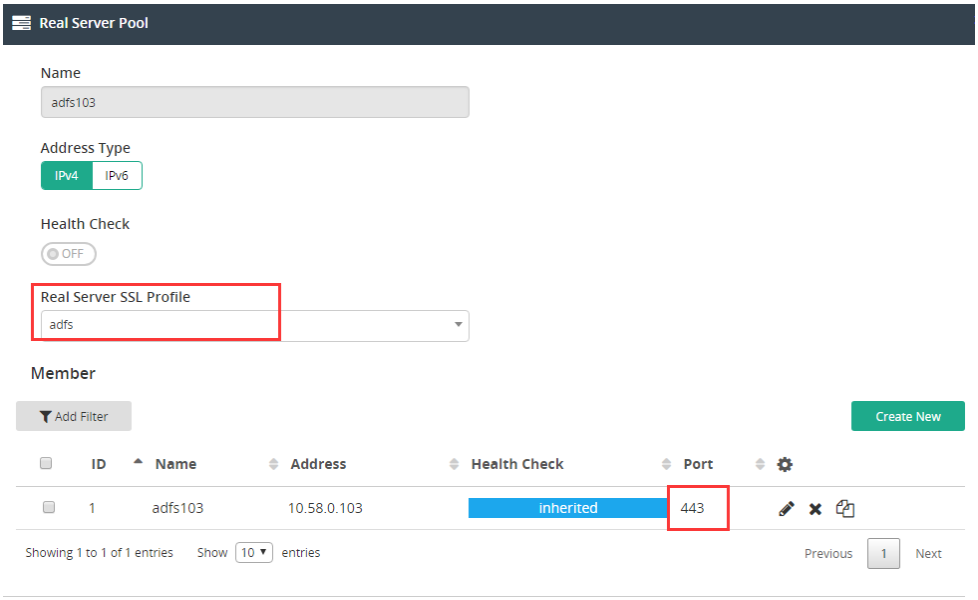
(1) Add AD FS server pool

Since the AD FS server uses https(443) to connect, the AD FS server pool must use 443 port and set real-server-ssl-profile. In real-server-ssl-profile, a local cert must be used, and the ssl-sni-forward must be set.

```
config load-balance real-server-ssl-profile
 edit "adfs"
  set ssl enable
  set ssl-sni-forward enable
  set local-cert Factory
 next
end
config load-balance real-server
 edit "adfs103"
  set ip 10.58.0.103
 next
end
config load-balance pool
 edit "adfs103"
  set real-server-ssl-profile adfs
  config  pool_member
   edit 1
    set pool_member_service_port 443
    set pool_member_cookie rs1
    set real-server adfs103
   next
  end
 next
end
```


Server SSL
Name
adfs
SSL
ON
Customized SSL Ciphers Flag
OFF
SSL Cipher Suite List
ECDHE-ECDSA-AES256-GCM-SHA384
ECDHE-ECDSA-AES256-SHA
ECDHE-ECDSA-AES128-SHA256
ECDHE-ECDSA-DES-CBC3-SHA
ECDHE-RSA-AES256-GCM-SHA384
ECDHE-RSA-AES256-SHA
DHE-RSA-AES256-SHA256
AES256-GCM-SHA384
AES256-SHA
ECDHE-RSA-AES128-SHA256
DHE-RSA-AES128-GCM-SHA256
DHE-RSA-AES128-SHA
AES128-SHA256
ECDHE-RSA-RC4-SHA
RC4-MD5
EDH-RSA-DES-CBC3-SHA
eNULL
ECDHE-ECDSA-AES256-SHA384
ECDHE-ECDSA-AES128-GCM-SHA256
ECDHE-ECDSA-AES128-SHA
ECDHE-ECDSA-RC4-SHA
ECDHE-RSA-AES256-SHA384
DHE-RSA-AES256-GCM-SHA384
DHE-RSA-AES256-SHA
AES256-SHA256
ECDHE-RSA-AES128-GCM-SHA256
ECDHE-RSA-AES128-SHA
DHE-RSA-AES128-SHA256
AES128-GCM-SHA256
AES128-SHA
RC4-SHA
ECDHE-RSA-DES-CBC3-SHA
DES-CBC3-SHA
Allow SSL Versions
sslv3
tlsv1.0
tlsv1.1
tlsv1.2
Certificate Verify
Click to select
Local Certificate
Factory
SNI Forward Flag
ON
Save
Cancel

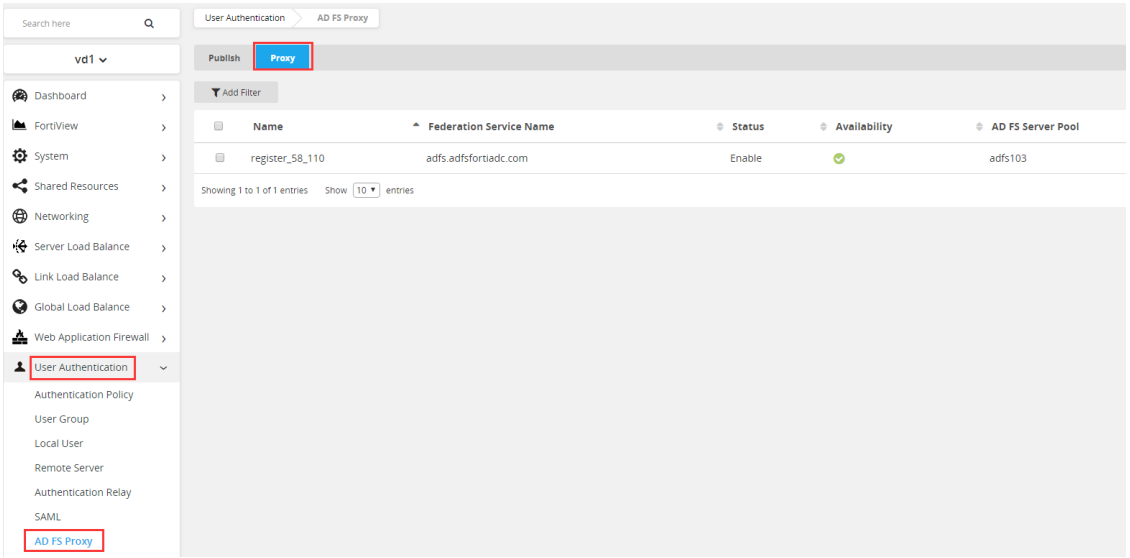

Real Server
Name
adfs103
Status
Enable
Disable
Maintain
Address
10.58.0.103
Address6
::

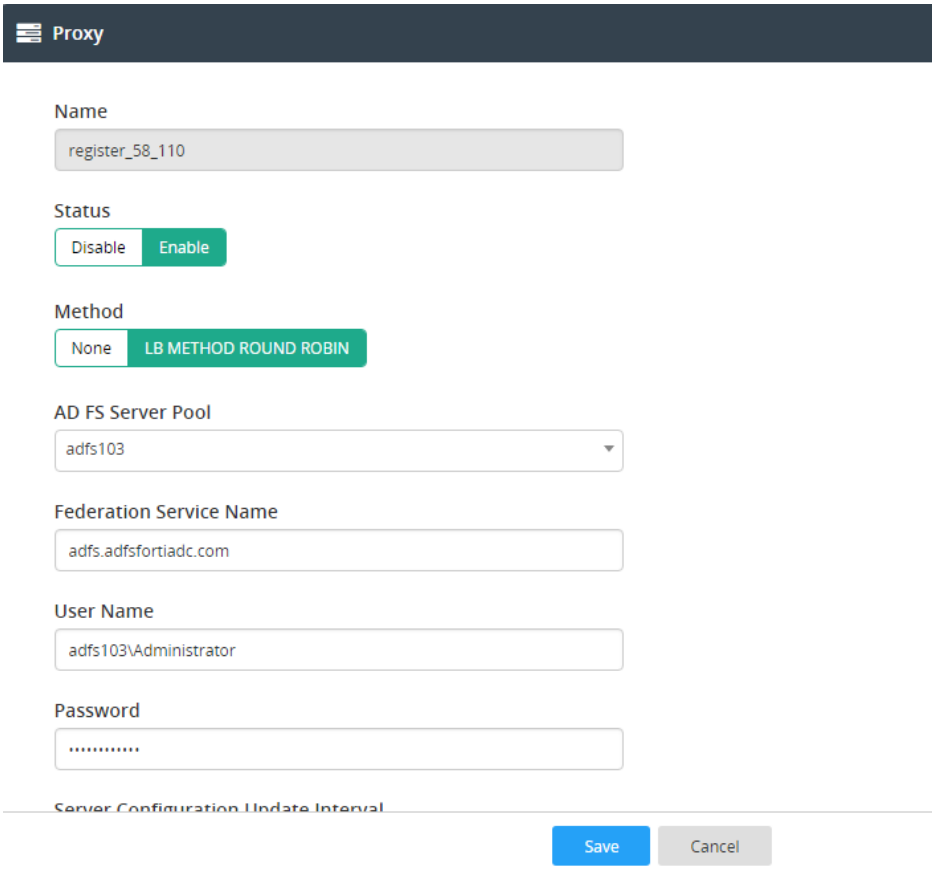


(2) Add AD FS proxy

FortiADC add an adfs-proxy for registering to AD FS server. So, the configuration should set according to AD FS server, the fqdn same as AD FS federation service, username should can login to the AD FS service.

```
config user adfs-proxy
  edit "register_58_110"
    set fqdn adfs.adfsfortiadc.com
    set load-balance-method LB_METHOD_ROUND_ROBIN
    set load-balance-pool adfs103
    set username "adfs103\\Administrator"
    set password ENC
skjgso1KvGVRmFh7RKUJkArz+X65udG2wAB8TajucCFOCbQe+zkNoH2eA+gL4Uv8yxnNzoG2Iq/Il
4qGv4L/nLdQhtj26FsgdvSoxoWSvu8x+Al1
  next
end
```





(3) After theh AD FS proxy registers successfully in AD FS server, FortiADC will get all the relyingpartytrust from AD FS server. For example, in my test environment, the relypartytrust is Device Registration Service, ExchangeOWA, ExchangeECP. Then FortiADC will save them as shown below, and in FADC, relyparttrust should not be edited, as they are required from AD FS server after AD FS Proxy in FADC registers successfully.

```
config user adfs-relying-party
 edit "register_58_110-Device_Registration_Service-1545054427"
  set proxy register_58_110
  set relying-party-trust "Device Registration Service"
 next
 edit "register_58_110-ExchangeOWA-1545054427"
  set proxy register_58_110
  set relying-party-trust ExchangeOWA
 next
 edit "register_58_110-ExchangeECP-1545054427"
  set proxy register_58_110
  set relying-party-trust ExchangeECP
 next
end
```

(4) Add AD FS publish

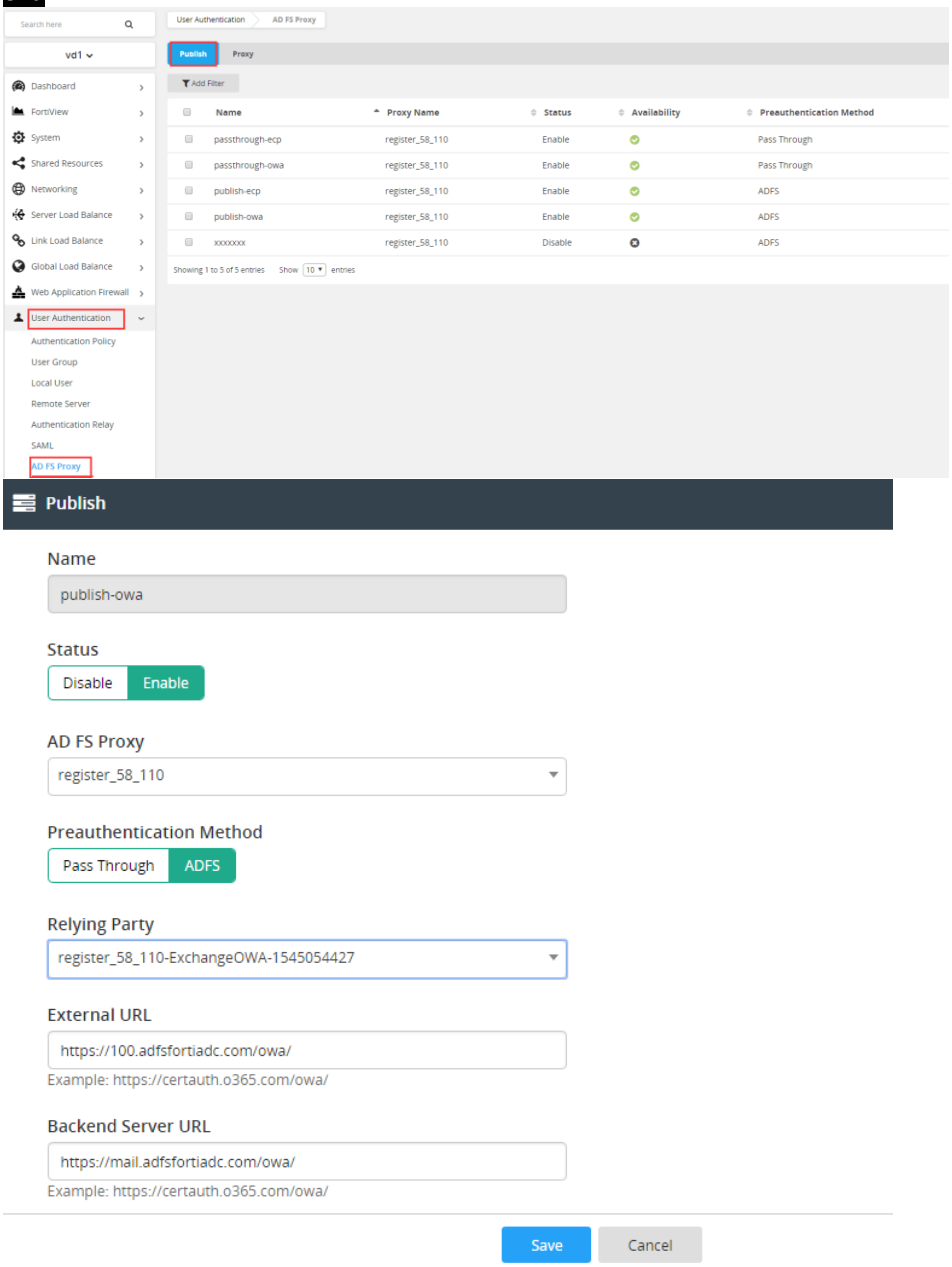
In this scenario, the method uses ADFS. The backend-server-url should be same as in AD FS server. The external-url is the url that client will visit.

That is to say, in my example, customer request exchange service via:

https://100.adfsfortiadc.com/owa/

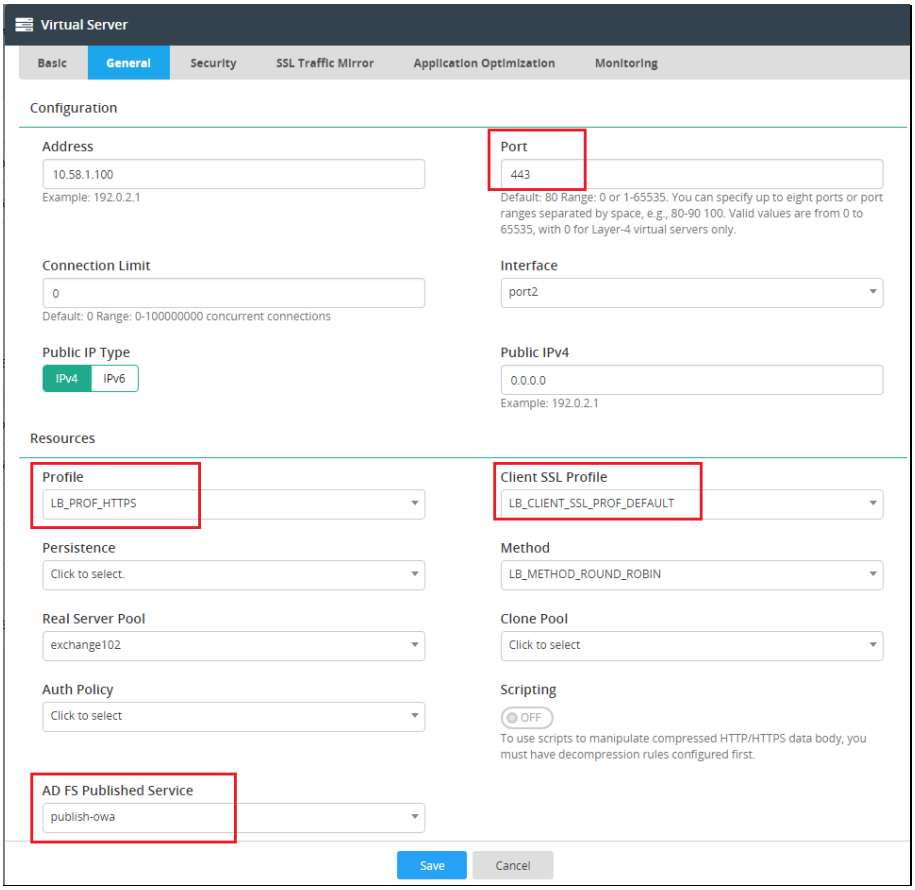
```
config user adfs-publish
 edit "publish-owa"
  set adfs-proxy register_58_110
  set pre-auth ADFS
  set relying-party register_58_110-ExchangeOWA-1545054427
  set external-url https://100.adfsfortiadc.com/owa/
  set backend-server-url https://mail.adfsfortiadc.com/owa/
```

```
 next
end
```



(5) Add exchange server pool and virtual server, set AD FS publish service to virtual server.

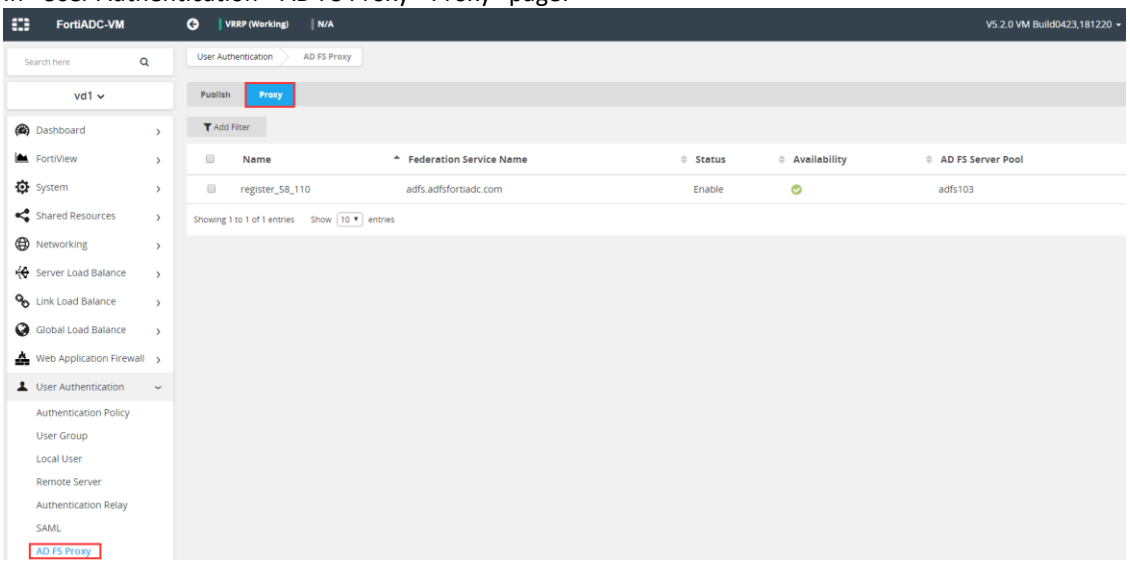
```
config load-balance virtual-server
 edit "publish_owa"
  set type l7-load-balance
  set interface port2
  set ip 10.58.1.100
  set port 443
  set load-balance-profile LB_PROF_HTTPS
  set client-ssl-profile LB_CLIENT_SSL_PROF_DEFAULT
  set load-balance-method LB_METHOD_ROUND_ROBIN
  set load-balance-pool exchange102
  set traffic-group default
  set adfs-published-service publish-owa
 next
end
```



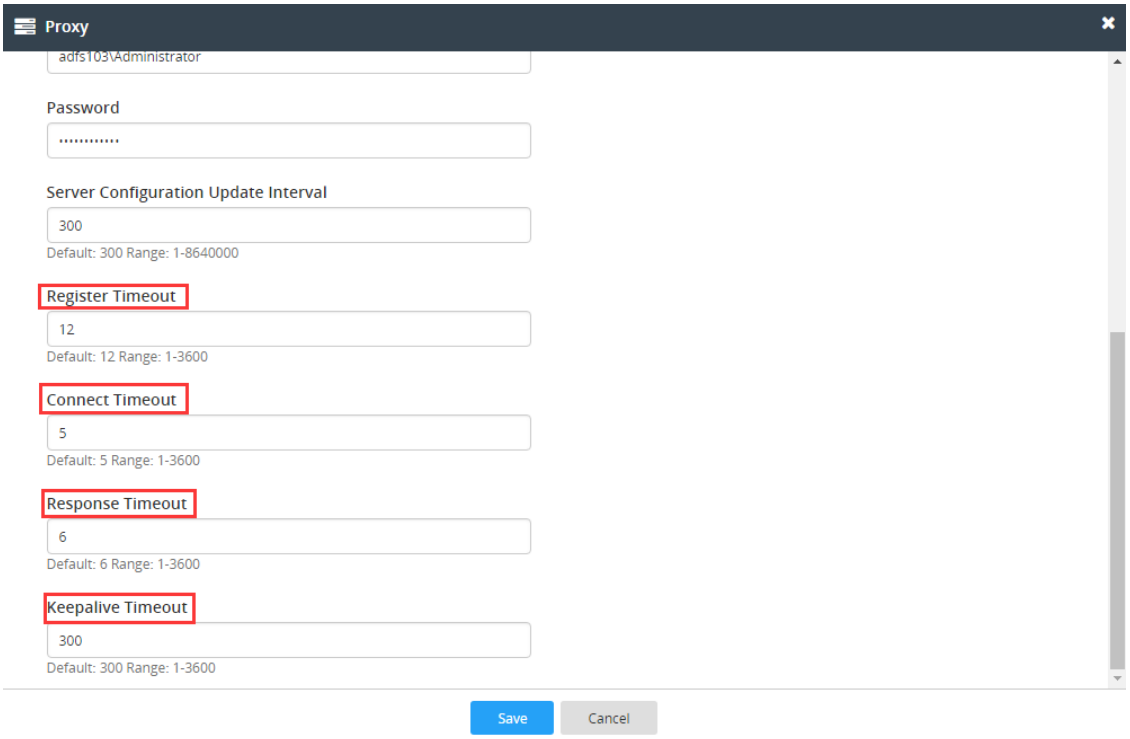
2) Configure adfs proxy advance option

AD FS proxy will register to AD FS server, and in the register connection, the customer can configure some timeouts to adapt to the AD FS server.

In “User Authentication->AD FS Proxy->Proxy” page:

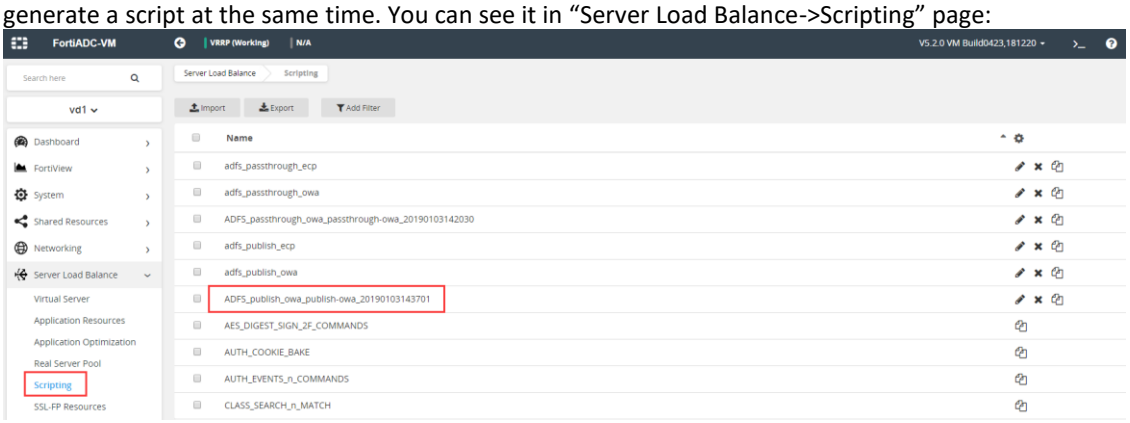


Open the proxy and configure the timeout as customer needed:



3.3.2 Configure using scripting

As shown in 3.3.1, when the virtual server uses the AD FS publish service, FortiADC will generate a script at the same time. You can see it in “Server Load Balance->Scripting” page:



Upon opening this script, you can get the action that FortiADC will operate. In this script, FortiADC will send the packets to a different backend server pool, which uses content routing. So the customer should add content routings that are the same as the script it is using.

Scripting

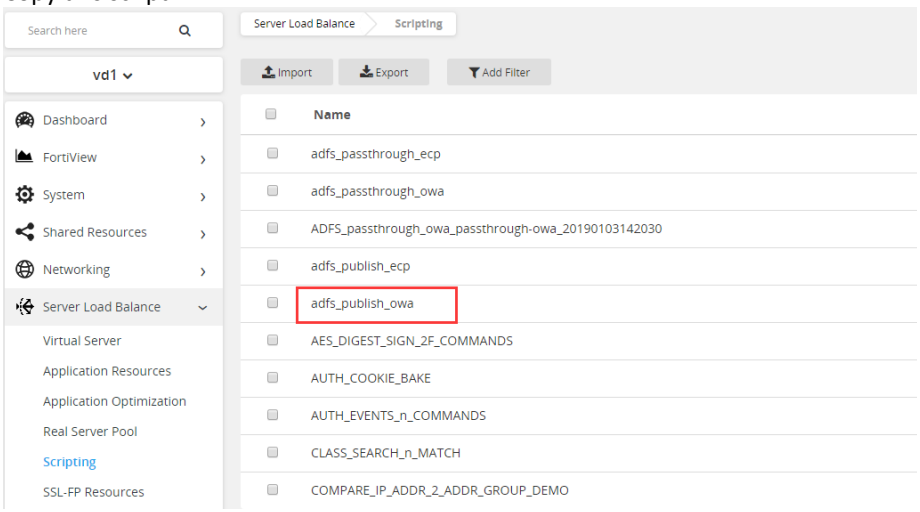
Name

ADFS_publish_owa_publish-owa_20190103143701

```
when RULE_INIT{
    cookie_name = "EdgeAccessCookie"
    cookie_value = "ADFSPROXYCOOKIE"
    target_cookie = string.format("%s=",  cookie_name)
    pub_uri ="/owa/"
    external_host = "100.adfsfortiadc.com"
    internal_uri = "/owa/"
    internal_domain = "mail.adfsfortiadc.com"
    adfs_server_domain = "adfs.adfsfortiadc.com"
    url_redirect="/adfs/ls?version=1.0&action=signin&realm=upn%3AAppProxy%3Acom&appRealm=ea879bbc-d7ca-e811-80b7-00
}

when HTTP_REQUEST{
    uri = HTTP:uri_get()
    host = HTTP:header_get_value("Host")
    path = HTTP:path_get()

    --content-routing
    --"msapp" "adfsserver" should have created in content-routing module
    if host:find(external_host) and uri:starts_with(pub_uri) then
        --redirect
        cookie_found = 0
        t={};
        t["name"]=cookie_name
        t["parameter"]="value"
        t["action"]="get"
        ret = HTTP:cookie(t)
        if ret then
            if cookie_value == ret then
                cookie_found = 1
            end
            --REMOVE this cookie
            t["parameter"]="cookie"
            t["action"]="remove"
            ret = HTTP:cookie(t)
            if ret then
                debug("remove cookie succeed %s\n", ret);
            else
                debug("remove cookie failed\n");
            end
        end
        return_url = string.format("https://%s%s",  host, uri)
        return_url = url_enc(return_url)
        redirect_url = string.format("https://%s%s",  adfs_server_domain, url_redirect)
        auth_token_invalid = 0
        if uri:find("authToken=") then
            --auth_token = get_uri_param(uri, "authToken")
            --if auth_token_verify(auth_token) then
                auth_token_invalid = 1
            --end
            HTTP:uri_set(path)
            --REMOVE all cookies
        ret = HTTP:cookie_list()
            for k,v in pairs(ret) do
                t["name"]=k
                t["parameter"]="cookie"
                t["action"]="remove"
                ret = HTTP:cookie(t)
                if ret then
                    debug("remove cookie succeed %s\n", ret);
                else
                    debug("remove cookie failed\n");
                end
            end
        end
        if cookie_found == 0 and auth_token_invalid == 0 then
            t={}
            t["code"] = 307;
            t["url"] = string.format("%s%s", redirect_url, return_url)
            HTTP:redirect_t(t);
        end
        HTTP:header_replace("Host", internal_domain)
        LB:routing("exchange102")
    elseif host:find(adfs_server_domain) then
        --insert ADFS header
        cip = IP:client_addr()  -- get client ip address
        HTTP:header_insert("X-MS-Endpoint-Absolute-Path", path)
        HTTP:header_insert("X-MS-Forwarded-Client-IP", cip)
        HTTP:header_insert("X-MS-Proxy", "FortiADC")
        LB:routing("adfs103")
    else
      debug("no matches domain for host: %s and uri %s\n", host, uri)
      HTTP:close()
    end
}

when HTTP_RESPONSE{
    location_header = "Location"
    if HTTP:header_exists(location_header) then
        location = HTTP:header_get_value(location_header)
        location_prefix = string.format("https://%s%s", adfs_server_domain, "/adfs/ls/")
        if location:starts_with(location_prefix) then
            signout = string.format("%s?wa=wsignout", location_prefix)
            if location:starts_with(signout) then
                insert_cookie = target_cookie
            else
                insert_cookie = string.format("%s%s", target_cookie, cookie_value)
            end
            cookie = string.format("%s; Path=%s; Secure; HttpOnly", insert_cookie, pub_uri)
            HTTP:header_insert("Set-Cookie", cookie)
        end
    end
}
```

Customer can use scripting and content routing instead of AD FS publish service.
Steps:

(1) Copy this script.



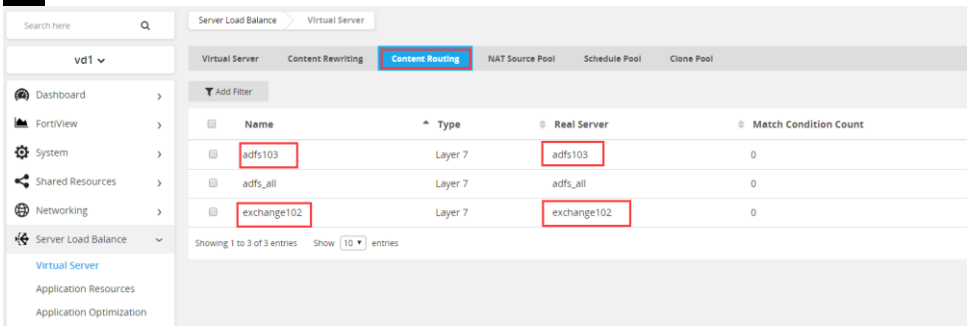
(2) In virtual server, unset AD FS publish service

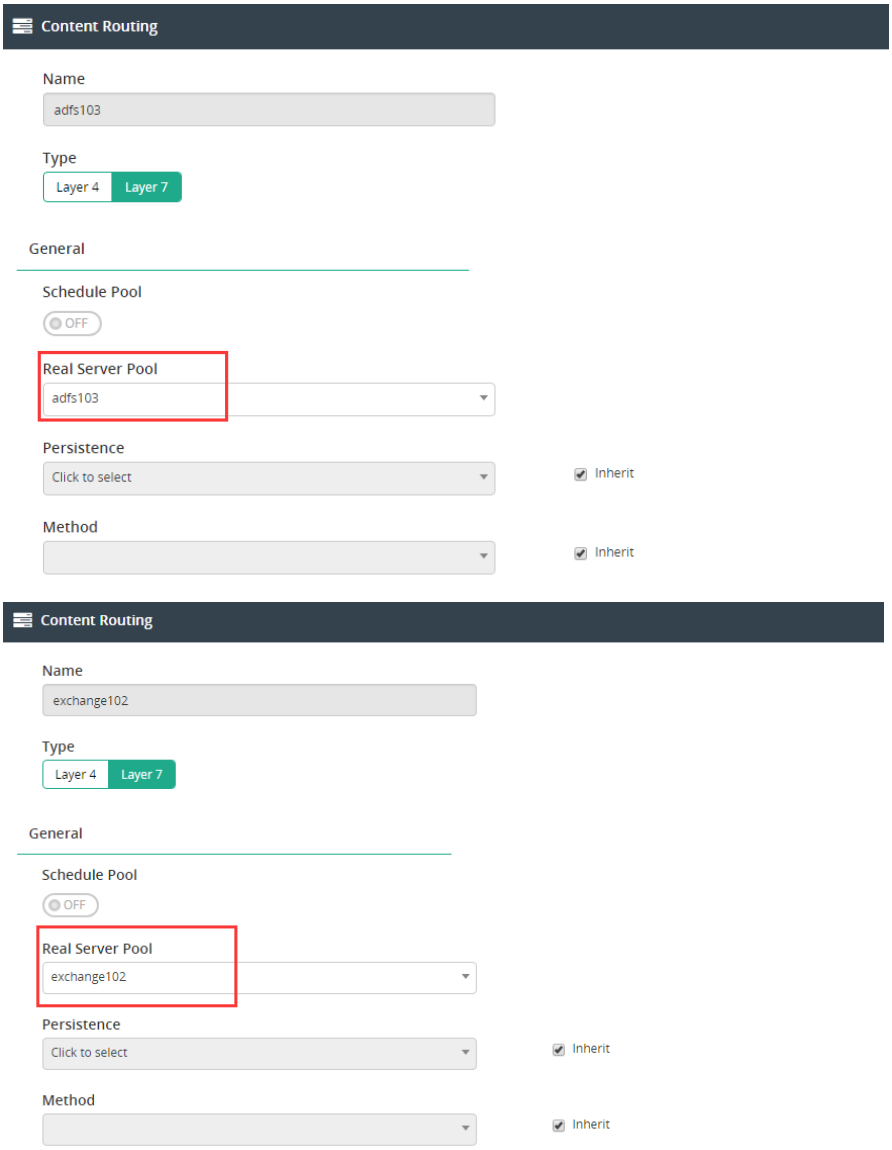
```
config load-balance virtual-server
 edit "publish_owa"
  unset adfs-published-service
 next
end
```

(3) Add content routing

It is the same as that used in the script, like in my example, content-routing “exchange102” using real-server-pool “exchange102”, and content-routing “adfs103” using real-server-pool “adfs103”

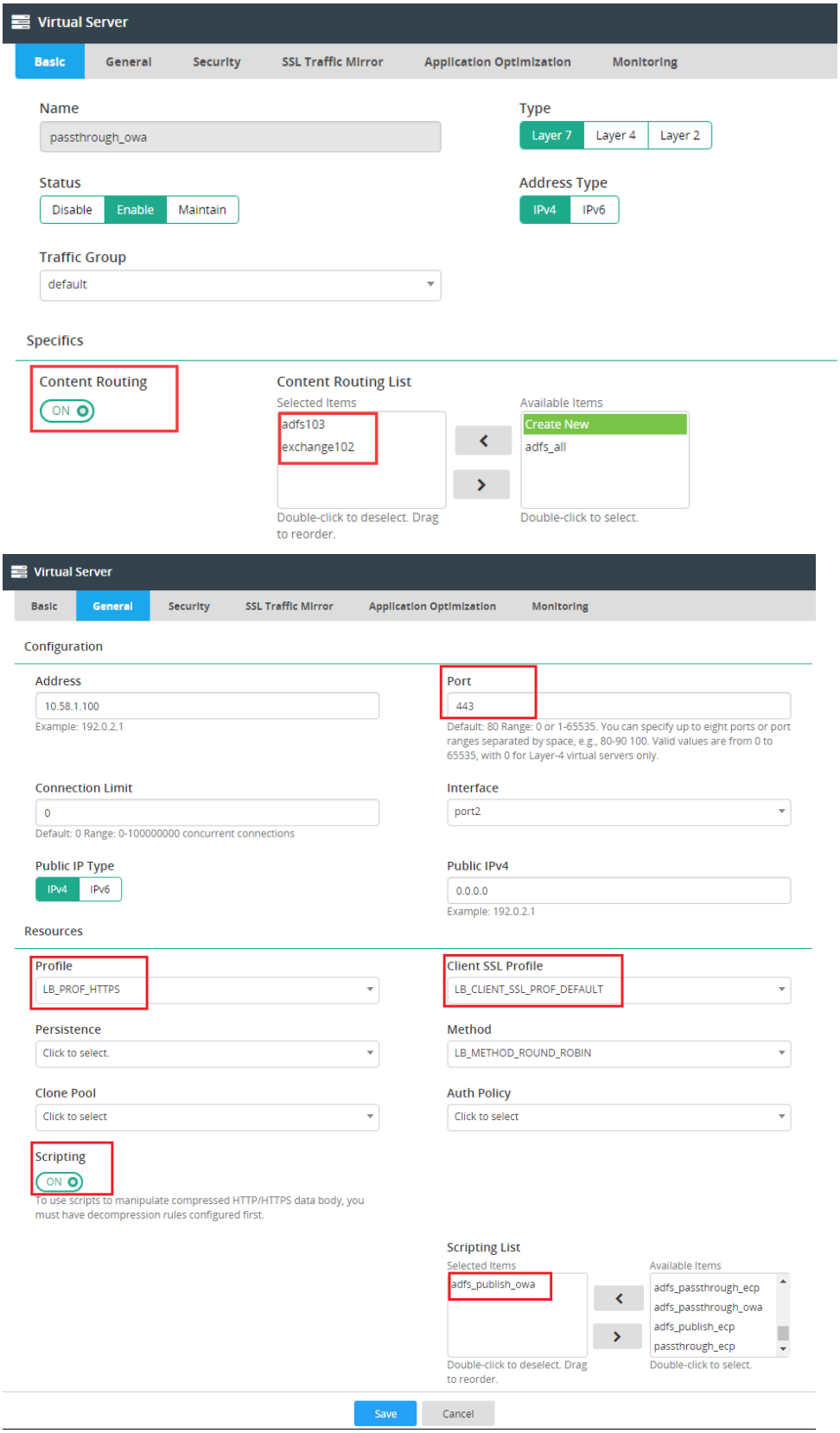
```
config load-balance content-routing
 edit "exchange102"
  set load-balance-pool exchange102
  config  match-condition
  end
 next
 edit "adfs103"
  set load-balance-pool adfs103
  config  match-condition
  end
 next
end
```





(4) Set the script that was copied in step 1 and in content routing in step 3; set it to the virtual server

```
config load-balance virtual-server
 edit "publish_owa"
  set type l7-load-balance
  set interface port2
  set ip 10.58.1.102
  set port 443
  set load-balance-profile LB_PROF_HTTPS
  set client-ssl-profile LB_CLIENT_SSL_PROF_DEFAULT
  set content-routing enable
  set content-routing-list adfs103 exchange102
  set load-balance-method LB_METHOD_ROUND_ROBIN
  set scripting-flag enable
  set scripting-list ADFS_publish_owa
  set traffic-group default
 next
end
```



Note:

(1) When configuring AD FS publish service to the virtual server, FortiADC generates a script. This script will be deleted by the FortiADC when the virtual server unsets AD FS publish service.

(2) Since AD FS publish uses adfs federation service, the client should configure the mapping between adfs federation service and virtual server ip address, between the external host and the virtual server ip address, such as:

adfs.adfsfortiadc.com 10.58.1.102(virtual server ip)

100.adfsfortiadc.com 10.58.1.102(virtual server ip)
Then client requests exchange service: https://100.adfsfortiadc.com/owa/

(3) AD FS publish cannot configure disabled AD FS proxy
(4) Virtual server cannot configure disabled AD FS publish

4 AD FS PROXY DEBUG

4.1 Enable AD FS Proxy debug

1) Enable system debug

```
FortiADC-VM # diagnose debug enable
```

2) Enable AD FS debug

```
FortiADC-VM # diagnose debug module adfs set
```

4.2 Get AD FS proxy and publish status

1) Get adfs proxy status

```
FortiADC-VM # execute adfs-proxy get  status register103
Avaliable
```

2) List adfs relyingparty of adfs-proxy

```
FortiADC-VM # execute adfs-proxy get  relying-party register103
{"RelyingParty":["Device Registration Service","ExchangeOWA","ExchangeECP"]}
```

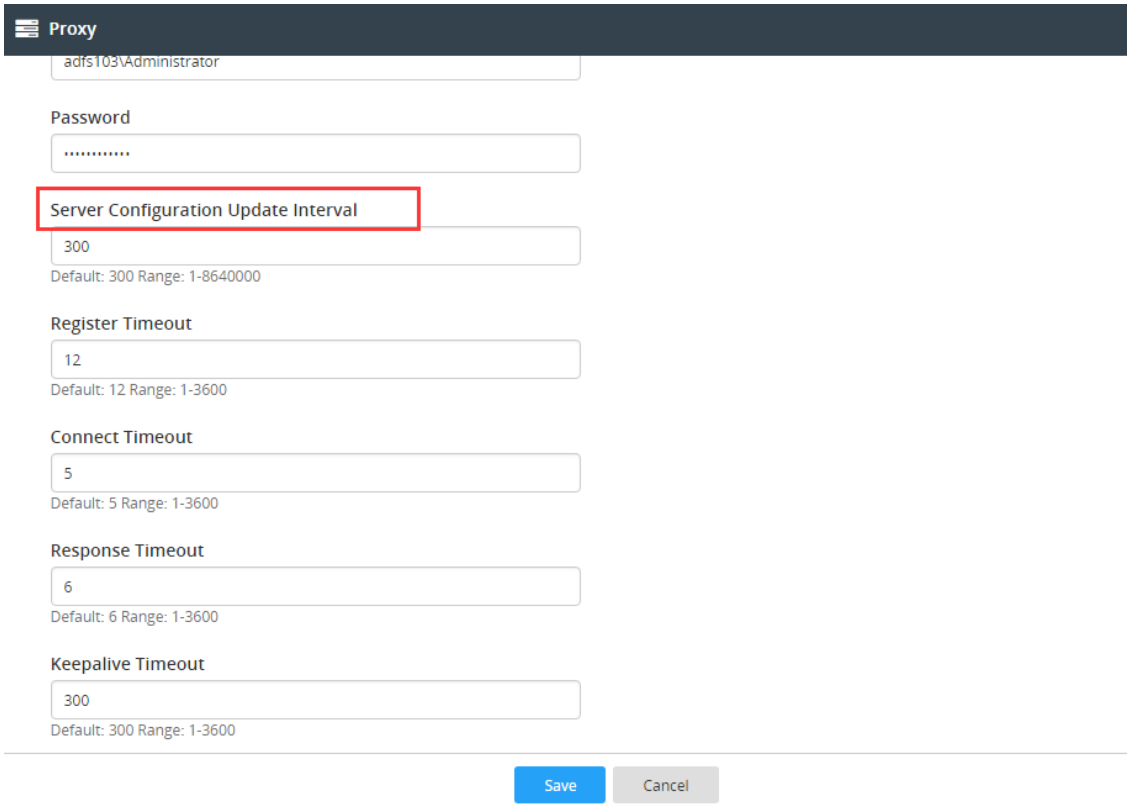
3) get adfs publish status

```
FortiADC-VM # execute adfs-publish get status  passthrough-owa proxy  register103
Avaliable
```

5 AD FS PROXY TROUBLESHOOTING

5.1 Server Configuration Update Interval

In order to decrease the request to AD FS server when getting an AD FS proxy, or getting publish and relyingpartytrust, a update interval can be configured in AD FS proxy.



Once AD FS proxy is registered to ADFS server and published successfully, FortiADC will cache relyingpartytrust and the status of proxy and publish. During the update interval, even though the customer can receive relyingpartytrust or status, FortiADC will not send requests to AD FS server until update interval timeout.

5.2 AD FS Proxy configuration sync in ha environment

If using AD FS publish service in an HA environment, the configuration and status of AD FS proxy and publish can sync to HA peers. The relyingpartyingtrust can also sync to HA peers.