# Web Application Proxy and AD FS on the AWS Cloud

Quick Start Reference Deployment

*Mike Pfeiffer*

*August 2015*
*Last updated: September 2015 (revisions)*

## Contents

Overview ........ 3
Web Application Proxy and AD FS on AWS ........ 3
Quick Links ........ 5
Cost and Licenses ........ 5
Architecture ........ 6
AWS Services ........ 7
Design Considerations ........ 7
Selecting an Instance Type ........ 8
Database Options ........ 8
Load Balancing ........ 9
Certificates ........ 9
Web Application Proxy Certificates ........ 9
AD FS Certificates ........ 10
Domain-Joined Proxies ........ 10
Authentication Scenarios ........ 10
Automated Deployment ........ 11
What We'll Cover ........ 11
Step 1. Prepare an AWS Account ........ 12
Step 2. Launch the Web Application Proxy and AD FS Stack ........ 14
Security ........ 16
Operating System Security ........ 17
Security Groups ........ 17
Additional Resources ........ 17
Appendix: Publishing Outlook Web App to the Internet with AD FS Pre-Authentication .. 20
Send Us Feedback ........ 29
Document Revisions ........ 29

## About This Guide

This Quick Start reference deployment guide discusses architectural considerations and configuration steps for deploying a Web Application Proxy and Active Directory Federation Services (AD FS) environment on the Amazon Web Services (AWS) cloud. It also provides links for viewing and launching AWS CloudFormation templates that automate the deployment.

The guide is for IT infrastructure architects, administrators, and DevOps professionals who are planning to implement or extend their Web Application Proxy and AD FS workloads on the AWS cloud.

Quick Starts are automated reference deployments for key enterprise workloads on the AWS cloud. Each Quick Start launches, configures, and runs the AWS compute, network, storage, and other services required to deploy a specific workload on AWS, using AWS best practices for security and availability.

# Overview

## Web Application Proxy and AD FS on AWS

Microsoft Active Directory Federation Services (AD FS) is a Windows Server role that provides identity federation and single sign-on (SSO) capabilities for users accessing applications in an AD FS-secured environment, or with federated partner organizations. Put simply, AD FS authenticates users and provides security tokens to applications or federated partner applications that trust AD FS.

For example, you could implement identity federation with AWS Identity and Access Management (IAM) and AD FS, and then use your Active Directory user name and password (instead of the AWS root account or IAM user credentials) to sign in to the AWS Management Console, or to make calls to AWS APIs.

Like domain controllers and other internal server workloads, AD FS servers are deployed in a private Amazon Virtual Private Cloud (Amazon VPC) subnet. In order to make AD FS accessible to external users, you can deploy the Web Application Proxy role on Windows Server 2012 R2. The Web Application Proxy server can proxy requests to the AD FS infrastructure for users who are connecting from an external location, without the need for VPN connectivity.

You can also use Web Application Proxy to selectively publish and pre-authenticate connections to internal web applications, allowing external users outside your organization to access those applications over the Internet.

In this guide, we'll take a look at using your own Active Directory Domain Services (AD DS) infrastructure in AWS, along with AD FS and Web Application Proxy, to provide seamless external access to web applications running in AWS.

Some of the benefits and features of publishing applications with Web Application Proxy and AD FS are:

- **Network isolation** – Publishing web applications through Web Application Proxy means that back-end servers are never directly exposed to the Internet. You can publish popular web-based workloads such as Microsoft SharePoint, Outlook Web App (OWA), Exchange ActiveSync, Lync (Skype for Business), and even custom web applications through Web Application Proxy.
- **Denial-of-service (DoS) protection** – The Web Application Proxy infrastructure uses several mechanisms to implement basic DoS protection, such as throttling and queuing, before routing connections to back-end web applications.
- **Multi-factor authentication** – Pre-authentication with AD FS provides support for smart cards, device authentication, and more.
- **Single sign-on (SSO)** – This functionality provides users with seamless access to applications without re-prompting for credentials after initial authentication.
- **Workplace Join** - Users can connect devices that are not typically domain-joined, such as personal laptops, tablets, and smartphones, to their company's resources. Known devices can be granted conditional access to applications, and you can require that devices register before gaining access to published applications.

For further details, see Planning to Publish Applications Using Web Application Proxy on Microsoft TechNet.

This guide and associated AWS CloudFormation template can be used in conjunction with other AWS Quick Starts to securely publish web applications running on SharePoint, Exchange, Lync, or your own web-based applications. The infrastructure deployed by this Quick Start enables external users to pre-authenticate to AD FS to access these web applications, without exposing the applications or AD FS infrastructure directly to the Internet. You can also use this infrastructure to enable federation with AWS.

## Quick Links

The links in this section are for your convenience. Before you launch the Quick Start, please review the architecture, configuration, network security, and other considerations discussed in this guide.

View template

Launch Quick Start

The default configuration deploys a number of servers into a new Amazon VPC across two Availability Zones, as illustrated in Figure 1. Each Availability Zone contains an Active Directory Domain Controller, an AD FS server, a Web Application Proxy server, a network address translation (NAT) server, and a Remote Desktop Gateway.

**Time to deploy:** Approximately 1.5 hours

## Cost and Licenses

You are responsible for the cost of the AWS services used while running this Quick Start reference deployment. There is no additional cost for using the Quick Start. As of the date of publication, the cost for using the Quick Start with default settings is approximately $5.50 an hour. Prices are subject to change. See the pricing pages for each AWS service you will be using in this Quick Start for full details.

This Quick Start launches the Amazon Machine Image (AMI) for Windows Server 2012 R2. The AMI is updated on a regular basis with the latest service pack for the operating system, so you don't have to install any updates.

AD FS and Web Application Proxy are server roles within Windows Server 2012 R2. **The architecture deployed by this Quick Start does not require any additional licenses from Microsoft.** The pay-as-you-go hourly cost for each EC2 instance covers your Windows Server license along with the Web Application Proxy and AD FS components.

There are a number of Microsoft enterprise applications that can be deployed and licensed through the Microsoft License Mobility through Software Assurance program. For development and test environments, you can leverage your existing MSDN licenses using Amazon EC2 Dedicated Instances. For details, see the MSDN on AWS page.

# Architecture

Deploying this Quick Start with the **default parameters** builds the following Web Application Proxy and AD FS environment in the AWS cloud.

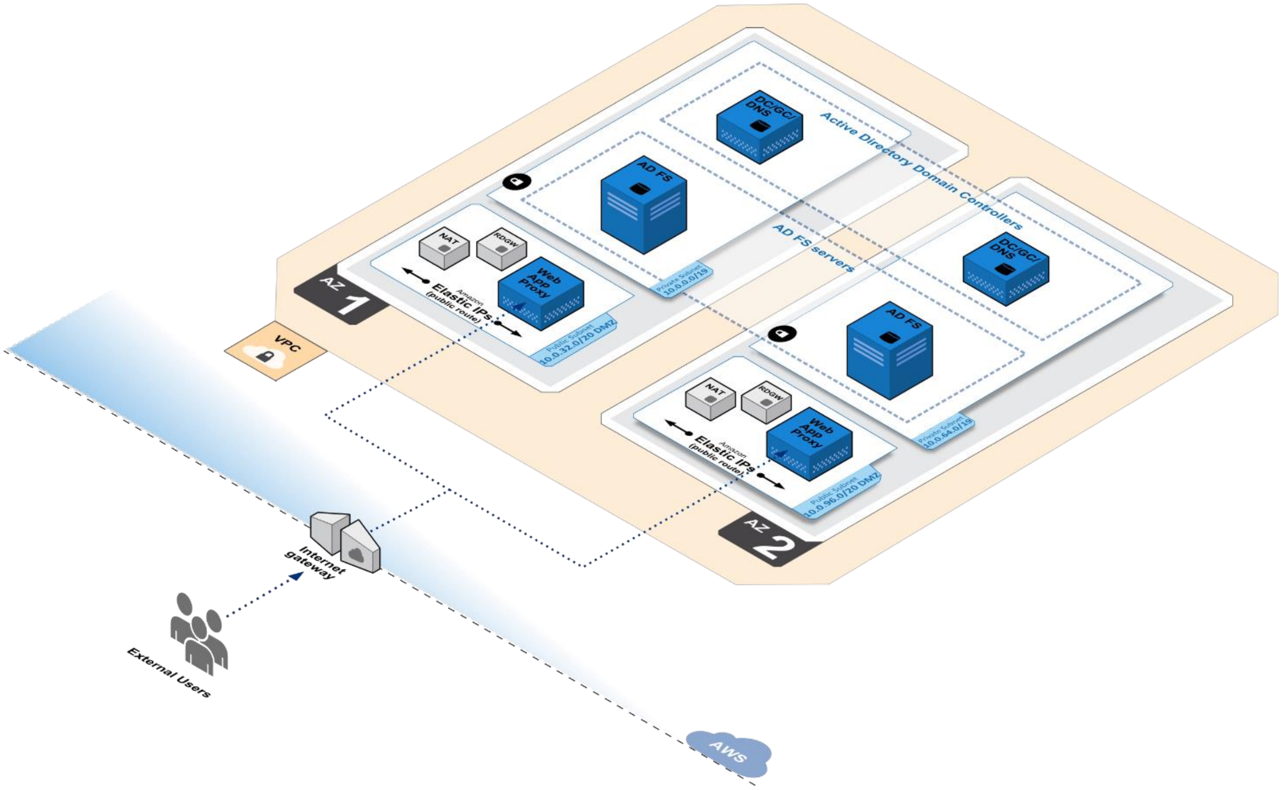

DC/GC/ DNS
Active Directory Domain Controllers
AD FS
AD FS servers
NAT
RDGW
Web App Proxy
Amazon Elastic IPs (public route)
Public Subnet 10.0.32.0/20 DMZ
Private Subnet 10.0.0.0/19
AZ 1
VPC
DC/GC/ DNS
AD FS
NAT
RDGW
Web App Proxy
Amazon Elastic IPs (public route)
Public Subnet 10.0.96.0/20 DMZ
Private Subnet 10.0.64.0/19
AZ 2
Internet gateway
External Users
AWS

**Figure 1: Quick Start Architecture for Web Application Proxy and AD FS on AWS**

The AWS CloudFormation template creates a fully functional AD FS federation server farm with Web Application Proxy on the AWS cloud. The template deploys the following components:

- An Amazon Virtual Private Cloud (Amazon VPC) with resources distributed across two Availability Zones.
- Public subnets in each Availability Zone that provide access to and from the Internet. The public subnets include network address translation (NAT) instances for outbound Internet access, and Remote Desktop Gateway (RD Gateway) instances for inbound remote administrative access. Web Application Proxy servers are deployed in the public subnets to help provide secure inbound connectivity to web applications.

- Private subnets in each Availability Zone for running enterprise workloads such as Active Directory domain controllers and AD FS servers, shielded from direct access over the Internet. The domain controllers act as enterprise certificate authorities (CAs) that issue the required SSL certificates to the AD FS infrastructure. For production deployments, you might want to consider commercial certificates issued from a public CA, and we'll cover this in greater detail later in this guide.
- Security groups to tightly control the flow of traffic between your Amazon EC2 instances.
- Two AD FS servers running on Windows Server 2012 R2, which are deployed in each Availability Zone to support high availability and load distribution.

## AWS Services

The core AWS components used by this Quick Start include the following AWS services. (If you are new to AWS, see the Getting Started section of the AWS documentation.)

- Amazon VPC – The Amazon Virtual Private Cloud (Amazon VPC) service lets you provision a private, isolated section of the AWS cloud where you can launch AWS services and other resources in a virtual network that you define. You have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways.
- Amazon EC2 – The Amazon Elastic Compute Cloud (Amazon EC2) service enables you to launch virtual machine instances with a variety of operating systems. You can choose from existing Amazon Machine Images (AMIs) or import your own virtual machine images.
- Amazon EBS – Amazon Elastic Block Store (Amazon EBS) provides persistent block-level storage volumes for use with Amazon EC2 instances in the AWS cloud. Each Amazon EBS volume is automatically replicated within its Availability Zone to protect you from component failure, offering high availability and durability. Amazon EBS volumes provide the consistent and low-latency performance needed to run your workloads.

# Design Considerations

This Quick Start is designed for a highly available AD FS implementation that supports 1,000 to 15,000 users, but there are a number of options available for architecting an AD FS deployment. Here are Microsoft's recommendations for determining the minimum number of servers to deploy:

- **Fewer than 1,000 users** – A small environment can use existing infrastructure instead of running dedicated AD FS servers. If you need to support fewer than 1,000 users, you can install AD FS on at least two of your domain controllers. Ideally, these domain controllers should be in two separate Availability Zones.
- **1,000 to 15,000 users** – In this scenario, Microsoft recommends using a dedicated AD FS and Web Application Proxy server infrastructure. The AD FS database can run using a Windows Internal Database (WID), so you'll need four servers (two Web Application Proxy, two AD FS) in this architecture, as shown in Figure 1.
- **15,000 to 60,000 users** – For large environments, Microsoft recommends using three to five dedicated AD FS servers, and at least two dedicated Web Application Proxy servers. Note that if you're scaling beyond five dedicated AD FS servers, you'll need a dedicated SQL Server instance instead of running a WID.

These recommendations are based on a hardware profile that supports 8 CPU cores, 4 GiB of RAM, and a 1 Gigabit network connection.

## Selecting an Instance Type

AD FS is considered a processor-bound workload, meaning that CPU resources are the highest in demand. This Quick Start uses C4 compute-optimized instances by default.

Specifically, the **c4.2xlarge** instance type is used to provide 8 vCPUs and 15 GiB of memory to meet or exceed requirements, based on the recommendations in the previous section. Additionally, the c4.2xlarge instance type supports Amazon EBS optimization, enhanced networking, and high network performance, which results in higher packets per second, lower latency, and lower jitter. Although c4.2xlarge provides more memory than required, it's a great candidate for workloads running in a production environment.

## Database Options

The recommended topology for AD FS is to create a federation server farm that includes at least two AD FS servers. When you install AD FS on the first server, the federation server farm is created. You can join the next server to the farm, and then load-balance those servers.

An AD FS federation server farm uses a database to hold configuration data. For farms with five or fewer servers, you can use a Windows Internal Database (WID). The primary AD FS server will have a read/write copy of this database. The secondary AD FS servers in the farm receive updates from the primary server to a read-only copy of the WID. If the primary AD FS server fails, the secondary server can still process authentication requests, but you cannot make configuration changes until either the primary server is brought back online or the secondary server is converted to primary.

For federation server farms that have more than five AD FS servers, you'll need to use a SQL Server database for the configuration database. When you use SQL Server for your AD FS database, all members in the federation server farm have write access to the configuration data.

Microsoft recommends that you use WID until you scale past five AD FS servers. This Quick Start utilizes WID with AD FS by default.

## Load Balancing

For production deployments, you should implement load balancing to make your Web Application Proxy and AD FS services highly available. You can use Elastic Load Balancing or a third-party virtual load balancer appliance.

Both the Web Application Proxy and AD FS layers can be load balanced individually with Elastic Load Balancing. You can deploy an Internet-facing load balancer for the Web Application Proxy layer that will service users accessing published web applications over the Internet. In addition, you can configure an internal load balancer for the AD FS servers. You would then configure the Web Application Proxy layer to point to the load-balanced DNS name (e.g., sts.example.com) that resolves to the internal load balancer.

## Certificates

Certificates are required to install both the Web Application Proxy and AD FS components. This Quick Start automates the deployment and installation of these components by using certificates issued from an internal enterprise certificate authority (CA), which runs on the domain controller infrastructure. You can replace these certificates in your own production deployments. The requirements and recommendations for both the Web Application Proxy and AD FS layers are discussed in the next two sections.

### Web Application Proxy Certificates

- **Issuing CA** – Typically, the Web Application Proxy infrastructure will use certificates issued from a commercial or public CA, such as DigiCert or Verisign, which should be installed in the computer's personal certificate store. Using a public CA generally prevents you from having to install root certificates on your client devices. You can use your own public key infrastructure (PKI) to issue the Web Application Proxy certificates, but you'll need to ensure that client devices trust the issuing CA, which typically involves installing the root certificate on the devices as well as making the URL of your certificate revocation list externally accessible.

- **Certificate FQDNs** – You can explicitly set the certificate subject and subject alternative name (SAN) fields, or you can choose to use a wildcard certificate (e.g., *.example.com). For explicit naming, you'll need to set the subject field to the AD FS federation service name (e.g., sts.example.com). If you plan to use the Workplace Join feature, you'll also need two SAN entries: one for the federation service name (e.g., sts.example.com) and one for enterprise registration in the format enterpriseregistration.*yourdomain*.com. Additionally, you'll want SAN entries for any fully qualified domain name (FQDN) that you will be publishing, such as externally facing SharePoint sites, OWA, etc.

### AD FS Certificates

- **Issuing CA** – Typically, the AD FS infrastructure will use certificates issued from an internal PKI, such as an enterprise Active Directory CA, because the servers in the infrastructure are not Internet facing. This is especially useful in an Active Directory domain environment where all domain-joined machines will trust the issuing CA by default. If you choose not to domain-join your Web Application Proxy servers, you can install the CA root certificate on those servers in the computer's trusted root certificate authority store. If you do not have an existing PKI implementation, it's probably easiest to use the same public certificate on both the Web Application Proxy and AD FS servers.
- **Certificate FQDNs** – The AD FS certificate requires the federation service name to be set on the subject field (e.g., sts.example.com), or you can use a wildcard certificate.

## Domain-Joined Proxies

You may choose not to domain-join your Web Application Proxy servers, because they will be placed in public virtual private cloud (VPC) subnets. This is a typical practice for server workloads running in a demilitarized zone (DMZ). However, if the web application you want to publish through Web Application Proxy must support Integrated Windows authentication, you should domain-join the Web Application Proxy server.

This Quick Start automatically joins the Web Application Proxy servers to the Active Directory Domain Services environment. See the appendix for an example of how to publish a web application that uses Integrated Windows authentication.

## Authentication Scenarios

Publishing web applications with Web Application Proxy supports three authentication scenarios:

- **AD FS pre-authentication** – In this scenario, users authenticate against AD FS before gaining access to the published web application. This requires that you add an AD FS relying party trust to the federation service. For detailed coverage on AD FS pre-authentication flow, see Publish Applications using AD FS Preauthentication in the Microsoft TechNet Library.
- **Client certificate pre-authentication** – In this scenario, one or more external servers connect to an on-premises web application through the Web Application Proxy infrastructure using a certificate for authentication. Despite the name, this scenario should not be used for client devices that connect to a published web application. For more information, see Publish Applications using Client Certificate Preauthentication in the Microsoft TechNet Library.
- **Pass-through pre-authentication** – In this scenario, access to the web application is proxied directly to the back-end server without pre-authentication against AD FS. For example, this is the option you would use to make AD FS externally accessible. Subsequently published applications that use AD FS pre-authentication will access AD FS via pass-through pre-authentication.

See the appendix for an example that covers both AD FS and pass-through pre-authentication.

# Automated Deployment

The AWS CloudFormation template provided with this Quick Start bootstraps the AWS infrastructure and automates the deployment of Web Application Proxy and AD FS on the AWS cloud from scratch. Follow the step-by-step instructions in this section to set up your AWS account, customize the template, and deploy the software into your account.

**Note** This automated deployment uses a nested AWS CloudFormation template to launch the Active Directory Domain Services environment, followed by the Web Application Proxy and AD FS infrastructure. You can use the Web Application Proxy and AD FS template directly for existing environments, or you can stack it on top of existing Quick Starts that you've already deployed. See the appendix for an example.

## What We'll Cover

The procedure for deploying the Web Application Proxy and AD FS architecture on AWS consists of the following steps. For detailed instructions, follow the links for each step.

Step 1. Prepare an AWS account

- Sign up for an AWS account, if you don't already have one.
- Choose the region where you want to deploy the stack on AWS.

- Create a key pair in the region.
- Review account limits for Amazon EC2 instances, and request a limit increase, if needed.

Step 2. Launch the stack

- Launch the AWS CloudFormation template into your AWS account.
- Enter a value for the required *KeyPairName* parameter.
- Review the other template parameters, and adjust if necessary.

## Step 1. Prepare an AWS Account

1. If you don't already have an AWS account, create one at http://aws.amazon.com by following the on-screen instructions. Part of the sign-up process involves receiving a phone call and entering a PIN using the phone keypad.

2. Use the region selector in the navigation bar to choose the Amazon EC2 region where you want to deploy Web Application Proxy and AD FS on AWS.

   Amazon EC2 locations are composed of *regions* and *Availability Zones*. Regions are dispersed and located in separate geographic areas. This Quick Start uses the **c4.2xlarge** instance type for the Web Application Proxy and AD FS portion of the deployment. c4.2xlarge instances are currently available in all AWS regions except China (Beijing) and South America (São Paulo).

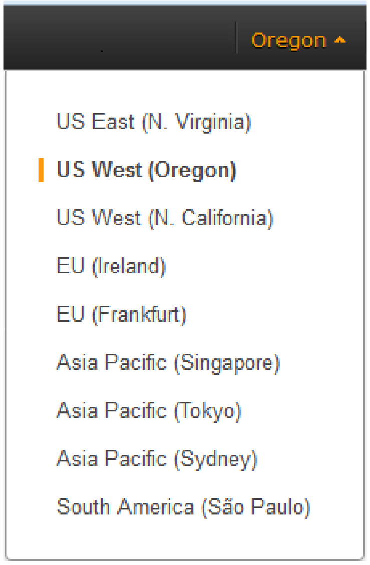


Figure 2: Choosing an Amazon EC2 Region

> **Tip** Consider choosing a region closest to your data center or corporate network to reduce network latency between systems running on AWS and the systems and users on your corporate network.

3. Create a key pair in your preferred region. To do this, in the navigation pane of the Amazon EC2 console, choose **Key Pairs**, **Create Key Pair**, type a name, and then choose **Create**.

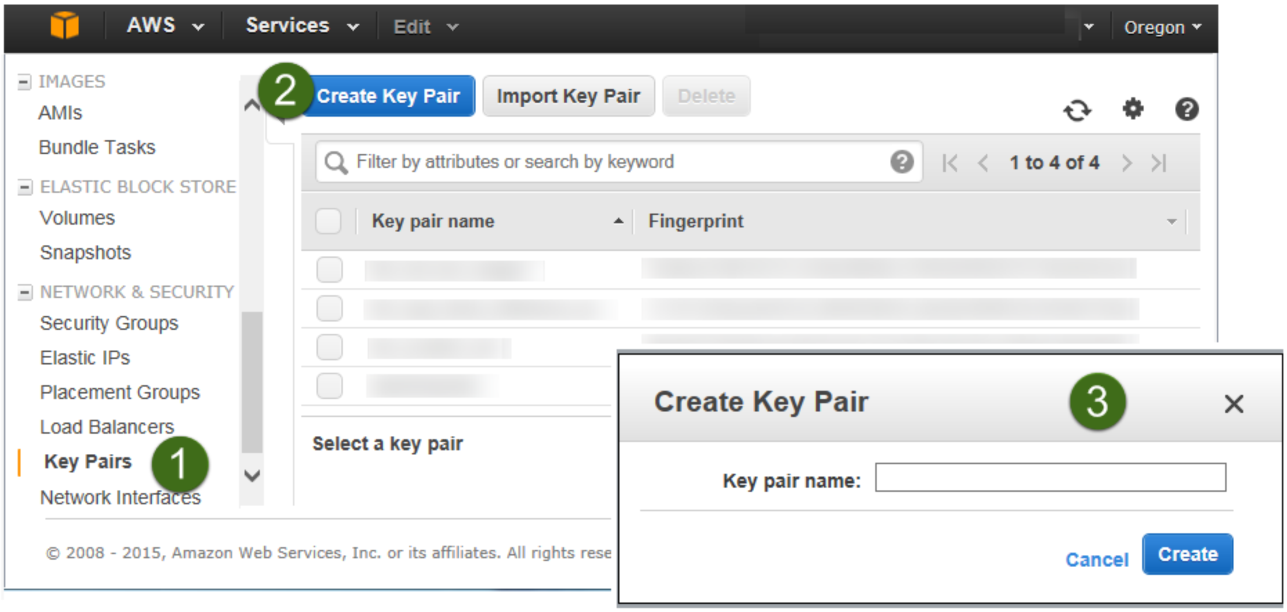
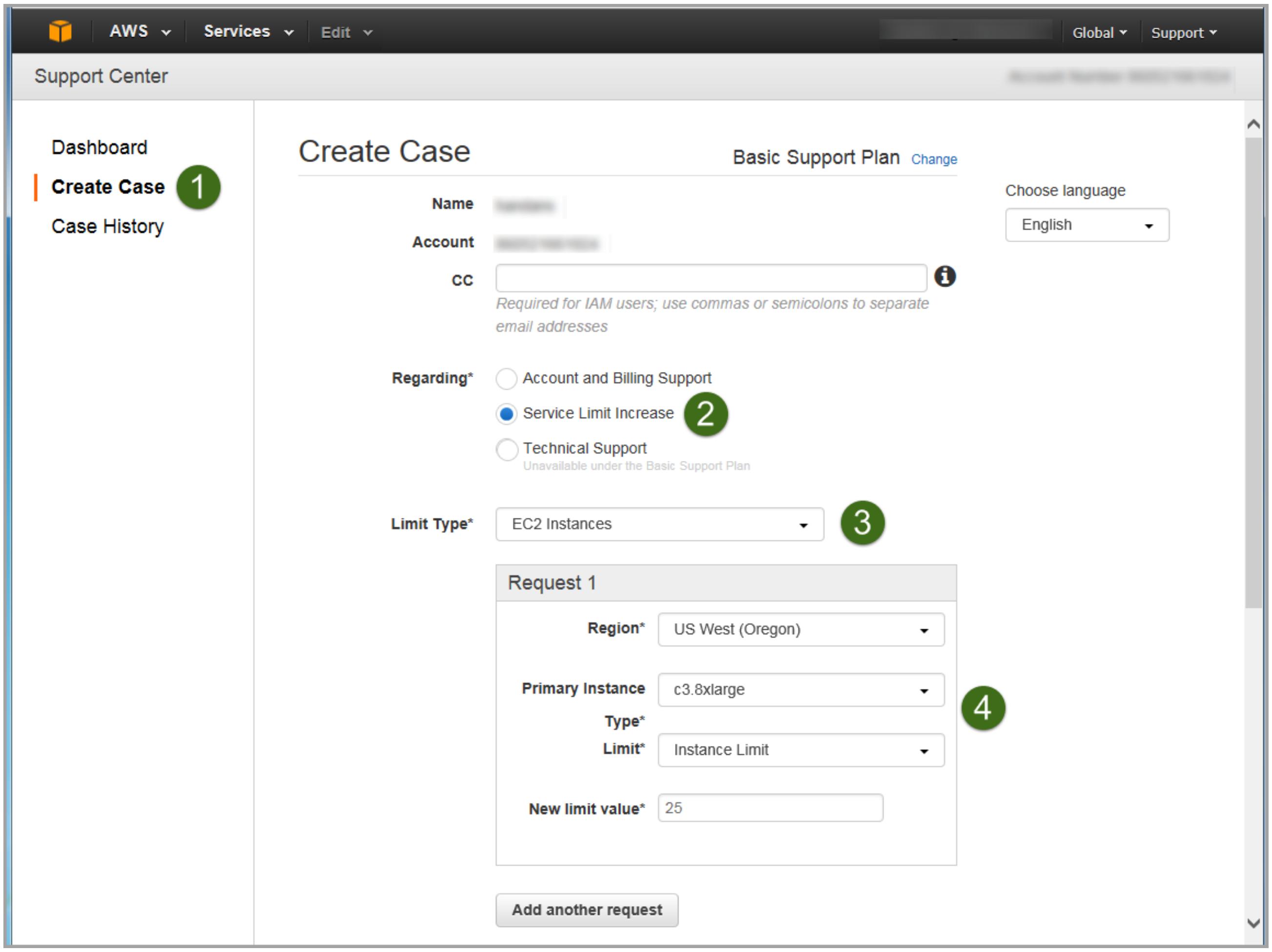


**Figure 3: Creating a Key Pair**

Amazon EC2 uses public-key cryptography to encrypt and decrypt login information. To be able to log in to your instances, you must create a key pair. With Windows instances, we use the key pair to obtain the administrator password via the Amazon EC2 console and then log in using Remote Desktop Protocol (RDP) as explained in the step-by-step instructions in the *Amazon Elastic Compute Cloud User Guide*.

4. If necessary, request a service limit increase for the Amazon EC2 c4.2xlarge instance type. To do this, in the AWS Support Center, choose **Create Case**, **Service Limit Increase**, **EC2 instances**, and then complete the fields in the limit increase form. The current default limit is 20 instances.

   You might need to request an increase if you already have an existing deployment that uses this instance type, and you think you might exceed the default limit with this reference deployment. It might take a few days for the new service limit to become effective. For more information, see Amazon EC2 Service Limits in the AWS documentation.



[redacted]

Name [redacted]

Account [redacted]

CC

*Required for IAM users; use commas or semicolons to separate email addresses*

Regarding*

○ Account and Billing Support

● Service Limit Increase

○ Technical Support
Unavailable under the Basic Support Plan

Limit Type* EC2 Instances

Request 1

Region* US West (Oregon)

Primary Instance Type* c3.8xlarge

Limit* Instance Limit

New limit value* 25

**Figure 4: Requesting a Service Limit Increase**

## Step 2. Launch the Web Application Proxy and AD FS Stack

This automated AWS CloudFormation template deploys Web Application Proxy and AD FS in multiple Availability Zones into an Amazon VPC. Please make sure that you've created a key pair in your chosen region before launching the stack.

1. Launch the AWS CloudFormation template into your AWS account.

   The template is launched in the US West (Oregon) region by default. You can change the region by using the region selector in the navigation bar.

   This stack takes approximately 1.5 hours to create.

**Launch stack**

**Note** You are responsible for the cost of the AWS services used while running this Quick Start reference deployment. There is no additional cost for using this Quick Start. As of the date of publication, the cost for using the Quick Start with default settings is approximately $5.50 an hour, and you can complete the initial deployment for about $8.25. Prices are subject to change. See the pricing pages for each AWS service you will be using in this Quick Start for full details.

2. On the **Select Template** page, keep the default settings for **Stack** and **Template Source**.
3. On the **Specify Parameters** page, review the parameters for the template. These are described in the following table.

   Provide values for the parameters that require your input. For all other parameters, the template provides default settings that you can customize.

| Parameter | Default | Description |
|---|---|---|
| **KeyPairName** | *Requires input* | Public/private key pair, which allows you to connect securely to your instance after it launches. This is the key pair you created in your preferred region in step 1. |
| **ADInstanceType** | m4.xlarge | Amazon EC2 instance type for the first Active Directory instance. |
| **AD2InstanceType** | m4.xlarge | Amazon EC2 instance type for the second Active Directory instance. |
| **NATInstanceType** | t2.small | Amazon EC2 instance type for the NAT instances. |
| **WAPADFSInstanceType** | c4.2xlarge | Amazon EC2 instance type for the Web Application Proxy and AD FS servers. |
| **RDGWInstanceType** | m4.xlarge | Amazon EC2 instance type for the Remote Desktop Gateway instances. |
| **DomainDNSName** | example.com | Fully qualified domain name (FQDN) of the forest root domain. |
| **DomainNetBIOSName** | example | NetBIOS name of the domain for users of earlier versions of Windows (up to 15 characters). |
| **ADServerNetBIOSName1** | DC1 | NetBIOS name of the first Active Directory server (up to 15 characters). |
| **ADServerNetBIOSName2** | DC2 | NetBIOS name of the second Active Directory server (up to 15 characters). |
| **RestoreModePassword** | *Requires input* | Password for a separate administrator account when the domain controller is in Restore Mode. This must be a complex password that's at least 8 characters long. |
| **DomainAdminUser** | StackAdmin | User name for the account that will be added as domain administrator. |

| Parameter | Default | Description |
|---|---|---|
| **DomainAdminPassword** | *Requires input* | Password for the domain administrator user (StackAdmin). This must be a complex password that's at least 8 characters long. |
| **DMZ1CIDR** | 10.0.32.0/20 | CIDR block for the public DMZ subnet located in Availability Zone 1. |
| **DMZ2CIDR** | 10.0.69.0/20 | CIDR block for the public DMZ subnet located in Availability Zone 2. |
| **PrivSub1CIDR** | 10.0.0.0/19 | CIDR block for the private subnet located in Availability Zone 1. |
| **PrivSub2CIDR** | 10.0.64.0/19 | CIDR block for the private subnet located in Availability Zone 2. |
| **VPCCIDR** | 10.0.0.0/16 | CIDR block for the Amazon VPC. |
| **AD1PrivateIp** | 10.0.0.10 | Primary private IP for the domain controller in Availability Zone 1. |
| **AD2PrivateIp** | 10.0.64.10 | Primary private IP for the domain controller in Availability Zone 2. |
| **UserCount** | 25 | Total number of test user accounts to create in Active Directory. |

**Note** You can also download the template and edit it to create your own parameters based on your specific deployment scenario.

4. On the **Options** page, you can specify tags (key-value pairs) for resources in your stack and set additional options. When you're done, choose **Next**.
5. On the **Review** page, review and confirm the settings.
6. Choose **Create** to deploy the stack.

# Security

When you build systems on the AWS infrastructure, security responsibilities are shared between you and AWS. This shared model can reduce your operational burden as AWS operates, manages, and controls the components from the host operating system and virtualization layer down to the physical security of the facilities in which the services operate. In turn, you assume responsibility and management of the guest operating system (including updates and security patches), other associated applications, as well as the configuration of the AWS-provided security group firewall. For more information about security on AWS, visit the AWS Security Center.

## Operating System Security

All the Windows Servers deployed by this Quick Start are domain-joined. You can authenticate to these instances by using the stackadmin@example.com domain administrator account. You can specify the password for this account as you launch the stack. You can retrieve the local administrator password for domain-joined instances by using the *KeyPairName* parameter specified during the launch.

Operating system patches are your responsibility and should be performed on a periodic basis.

## Security Groups

A *security group* acts as a firewall that controls the traffic for one or more instances. When you launch an instance, you associate one or more security groups with the instance. You add rules to each security group that allow traffic to or from its associated instances. You can modify the rules for a security group at any time. The new rules are automatically applied to all instances that are associated with the security group.

The security groups created and assigned to the individual instances as part of this solution are restricted as much as possible while allowing access to the various functions needed by AD FS and Web Application Proxy. We recommend that you review security groups and further restrict access as needed once the deployment is up and running.

# Additional Resources

### AWS services

- AWS CloudFormation
  http://aws.amazon.com/documentation/cloudformation/
- Amazon EBS
  - User guide:
    http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AmazonEBS.html
  - Volume types:
    http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSVolumeTypes.html
  - Optimized instances:
    http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSOptimized.html

- Amazon EC2
  - User guide for Microsoft Windows: http://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/
- Amazon VPC
  http://aws.amazon.com/documentation/vpc/

## Microsoft Web Application Proxy and AD FS

- Planning for AD FS Server Capacity
  https://technet.microsoft.com/en-us/library/gg749899.aspx
- Planning to Publish Applications Using Web Application Proxy
  https://technet.microsoft.com/en-us/library/dn383650.aspx
- Configure the Web Application Proxy Infrastructure
  https://technet.microsoft.com/en-us/library/dn383644.aspx
- Install and Configure the Web Application Proxy Server
  https://technet.microsoft.com/en-us/library/dn383662.aspx
- Publish Applications using AD FS Preauthentication
  https://technet.microsoft.com/en-us/library/dn383640.aspx
- Publish Applications using Pass-through Preauthentication
  https://technet.microsoft.com/en-us/library/dn383639.aspx
- Enabling Federation to AWS using Windows Active Directory, ADFS, and SAML 2.0
  http://blogs.aws.amazon.com/security/post/Tx71TWXXJ3UI14/Enabling-Federation-to-AWS-using-Windows-Active-Directory-ADFS-and-SAML-2-0

## Deploying Microsoft software on AWS

- Microsoft on AWS
  http://aws.amazon.com/microsoft/
- Secure Microsoft applications on AWS
  http://media.amazonwebservices.com/AWS_Microsoft_Platform_Security.pdf
- Microsoft Licensing Mobility
  http://aws.amazon.com/windows/mslicensemobility/
- MSDN on AWS
  http://aws.amazon.com/windows/msdn/
- AWS Windows and .NET Developer Center
  http://aws.amazon.com/net/

## Tools

- Best Practices Analyzer for Web Application Proxy
  https://technet.microsoft.com/en-us/library/dn383651.aspx
- Load-balancing solutions in the AWS Marketplace
  https://aws.amazon.com/marketplace/

## Associated Quick Start reference deployments

- Microsoft Active Directory on AWS
  http://docs.aws.amazon.com/quickstart/latest/active-directory-ds/
- Microsoft Remote Desktop Gateway on AWS
  http://docs.aws.amazon.com/quickstart/latest/rd-gateway/
- Additional reference deployments
  https://aws.amazon.com/quickstart/

# Appendix: Publishing Outlook Web App to the Internet with AD FS Pre-Authentication

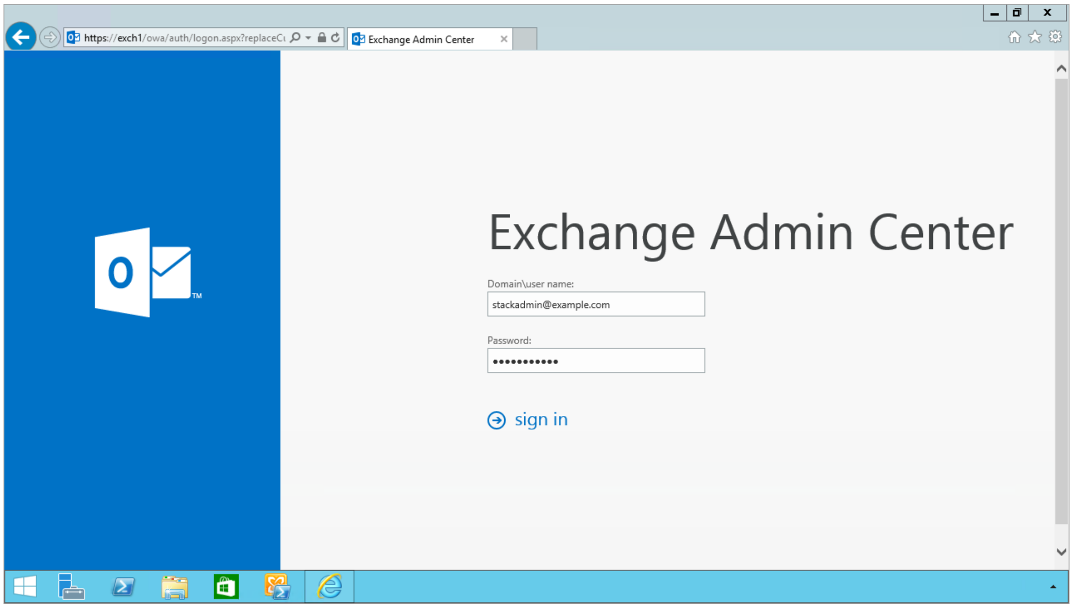
Instead of using the nested AWS CloudFormation template to launch a new environment, you can use the Web Application Proxy and AD FS template included with this Quick Start to launch the components into an existing Amazon VPC.

> **Important The sub-template for Web Application Proxy and AD FS provided with this guide is built to work with existing Amazon VPCs that have two public and two private subnets, and an existing Active Directory Domain Services implementation**. More specifically, it is designed to work with the existing Microsoft-based AWS Quick Starts, such as Exchange Server, SharePoint Server, and Lync Server.

In this appendix, we'll show you how to launch the Web Application Proxy and AD FS infrastructure on top of the Exchange Server 2013 Quick Start. Then we'll walk-through the steps to publish Outlook Web App (OWA) to the Internet using Web Application Proxy and AD FS.

> **Note** This walkthrough details the process of publishing OWA using Integrated Windows authentication. You can follow the same general process for Exchange Server 2010, or other web applications you want to publish with Integrated Windows authentication. It is also possible to publish OWA with claims-based authentication using Exchange Server 2013 SP1, but that scenario is beyond the scope of this guide.

1. Go to http://aws.amazon.com/quickstart and launch the Exchange Server 2013 Quick Start.
2. Once the Exchange Server 2013 stack has been created successfully, launch the Web Application Proxy and AD FS template. As shown previously in this guide, you'll need to specify the *KeyPairName* for your chosen region. Additionally, you'll need to specify the subnet CIDR ranges for both the public and private subnets, and the Domain Members security group ID. You can select these values from drop-down lists on the AWS CloudFormation template launch screen.
3. Initiate a Remote Desktop Protocol (RDP) connection to one of the RD Gateway instances. You can retrieve the Elastic IP for the RD Gateway servers in the Amazon EC2 console. From there, use RDP to connect to the EXCH1 server.
4. On EXCH1, navigate to the Exchange Admin Center (https://exch1/ecp) in a web browser. Sign in by using the stackadmin user account and password you specified when building the stack.



https://exch1/owa/auth/logon.aspx?replaceC

Exchange Admin Center

Exchange Admin Center

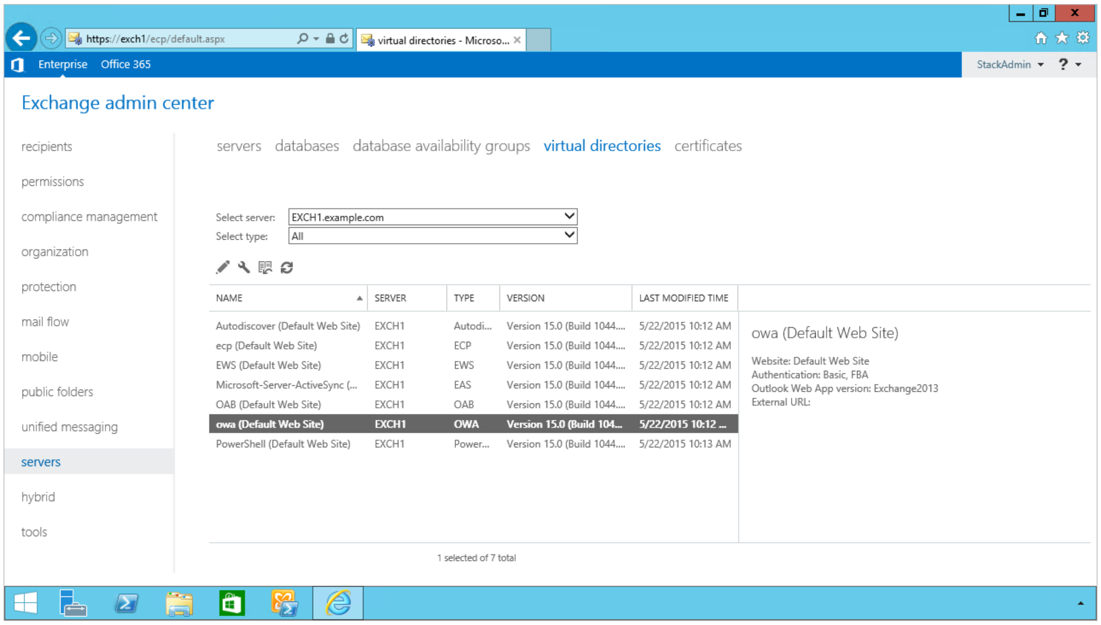
Domain\user name: stackadmin@example.com

Password: •••••••••••

sign in

Figure 5: Logging into the Exchange Admin Center

5. In the left pane, choose **Servers**, **Virtual directories**.



https://exch1/ecp/default.aspx

virtual directories - Microso...

Enterprise Office 365 StackAdmin ?

Exchange admin center

recipients<br>permissions<br>compliance management<br>organization<br>protection<br>mail flow<br>mobile<br>public folders<br>unified messaging<br>servers<br>hybrid<br>tools

servers databases database availability groups virtual directories certificates

Select server: EXCH1.example.com

Select type: All

| NAME | SERVER | TYPE | VERSION | LAST MODIFIED TIME |
|---|---|---|---|---|
| Autodiscover (Default Web Site) | EXCH1 | Autodi... | Version 15.0 (Build 1044.... | 5/22/2015 10:12 AM |
| ecp (Default Web Site) | EXCH1 | ECP | Version 15.0 (Build 1044.... | 5/22/2015 10:12 AM |
| EWS (Default Web Site) | EXCH1 | EWS | Version 15.0 (Build 1044.... | 5/22/2015 10:12 AM |
| Microsoft-Server-ActiveSync (... | EXCH1 | EAS | Version 15.0 (Build 1044.... | 5/22/2015 10:12 AM |
| OAB (Default Web Site) | EXCH1 | OAB | Version 15.0 (Build 1044.... | 5/22/2015 10:12 AM |
| owa (Default Web Site) | EXCH1 | OWA | Version 15.0 (Build 104... | 5/22/2015 10:12 ... |
| PowerShell (Default Web Site) | EXCH1 | Power... | Version 15.0 (Build 1044.... | 5/22/2015 10:13 AM |

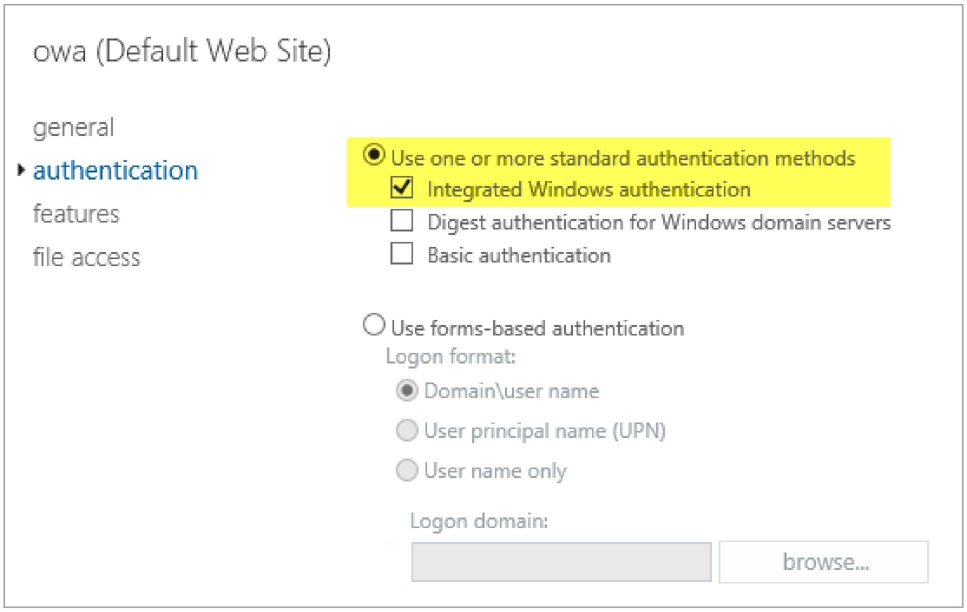
owa (Default Web Site)

Website: Default Web Site
Authentication: Basic, FBA
Outlook Web App version: Exchange2013
External URL:

1 selected of 7 total

Figure 6: Viewing the Virtual Directories on EXCH1

6. Double-click **owa (Default Web Site)** on the EXCH1 server. Choose **Authentication**, **Integrated Windows authentication**, and then choose **Save**. You should also change the corresponding setting on the ECP virtual directory on EXCH1.



owa (Default Web Site)

general

authentication

features

file access

● Use one or more standard authentication methods

☑ Integrated Windows authentication

☐ Digest authentication for Windows domain servers

☐ Basic authentication

○ Use forms-based authentication

Logon format:

● Domain\user name

○ User principal name (UPN)

○ User name only

Logon domain:

browse...

**Figure 7: Setting OWA Authentication to Integrated Windows**

**Note** In a load-balanced production environment, you would modify this setting on each Exchange server that is running the Client Access role.

7. Establish an RDP connection to the ADFS1 server. In **Control Panel**, choose **Administrative Tools**, and then launch the **ADFS Management** snap-in.
8. Open the context (right-click) menu for **Trust Relationships**, and then choose **Add Non-Claims-Aware Relying Party Trust** to start the wizard.

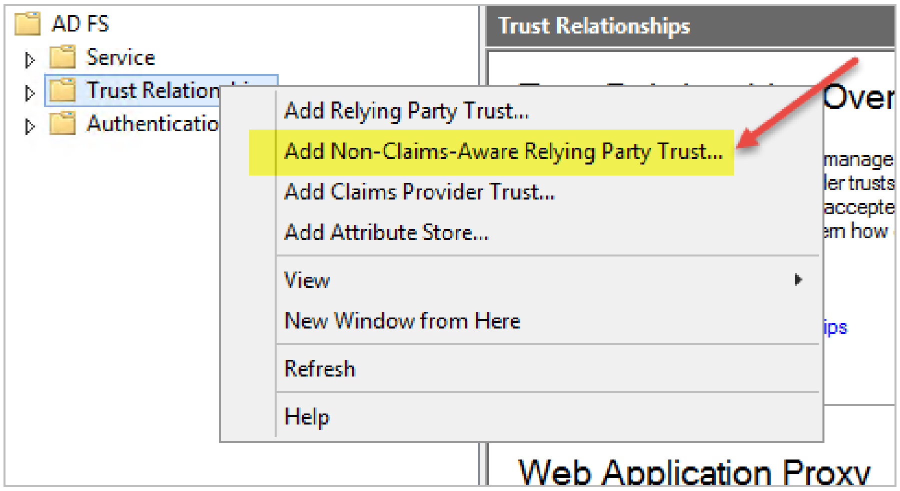
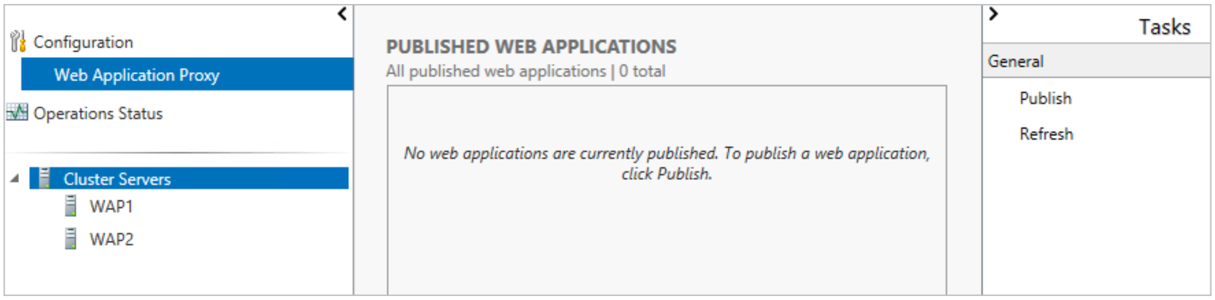


**Figure 8: Adding a Non-Claims-Aware Relying Party Trust**

9. On the welcome page of the wizard, choose **Start**, and type a display name such as **OWA**. Provide a unique identifier string for the non-claims-aware relying party trust. Use the default service name created by the Quick Start (e.g., http://sts.example.com/adfs/services/trust) for the URL.
10. Indicate that you do not want to configure multi-factor authentication, and then choose **Next**.
11. Go through the remaining screens without making changes. On the final screen, leave the **Open the Edit Issuance Authorization Rules** option selected, and then choose **Close**.
12. On the **Edit Claim Rules** screen, choose **Add Rule**, **Permit Access to All Users**, and then choose **Finish**.
13. Establish an RDP connection to the WAP1 server. In **Control Panel**, choose **Administrative Tools**, and then launch the **Remote Access Management** snap-in.

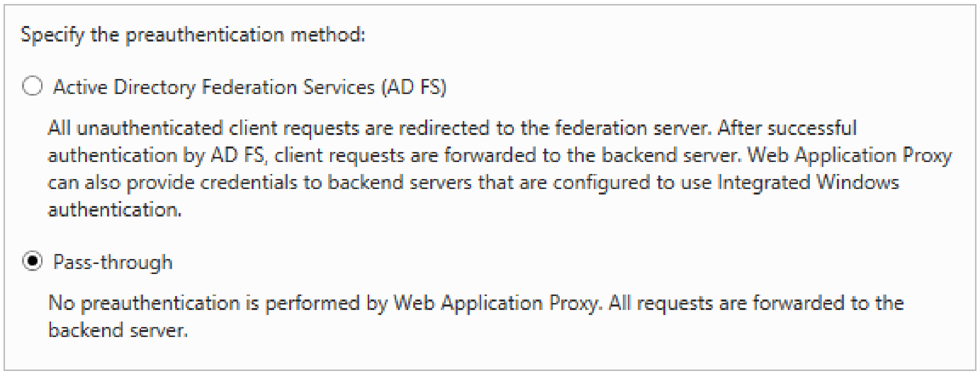


*No web applications are currently published. To publish a web application, click Publish.*

**Figure 9: Viewing the Remote Access Management Console**

To publish OWA to the Internet, we'll need to create two rules. The first rule will be a pass-through authentication rule to the ADFS server. This will allow users to pre-authenticate before being connected to OWA.

14. Under **Tasks**, choose **Publish**.
15. On the Welcome screen, choose **Next**. On the **Preauthentication** tab, choose **Pass-through**.



Specify the preauthentication method:

○ Active Directory Federation Services (AD FS)

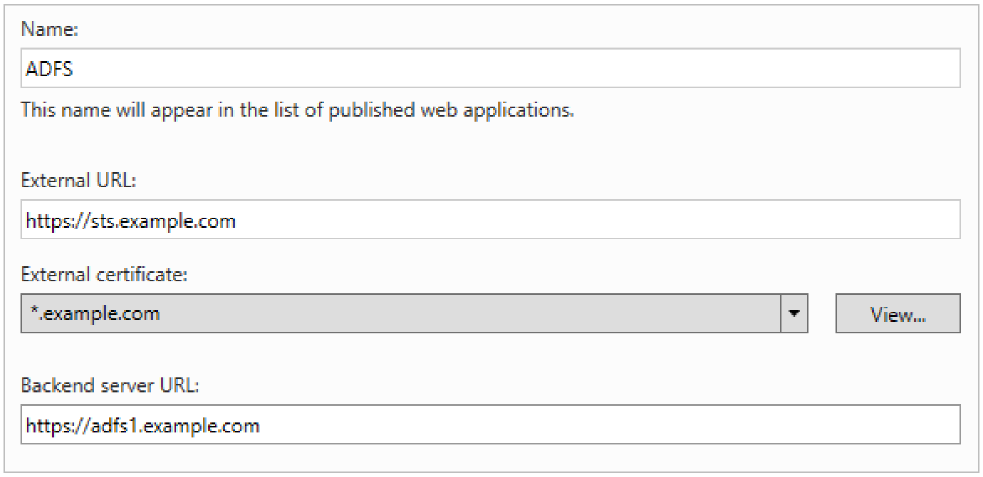
All unauthenticated client requests are redirected to the federation server. After successful authentication by AD FS, client requests are forwarded to the backend server. Web Application Proxy can also provide credentials to backend servers that are configured to use Integrated Windows authentication.

● Pass-through

No preauthentication is performed by Web Application Proxy. All requests are forwarded to the backend server.

**Figure 10: Selecting the Pass-Through Pre-Authentication Method**

16. Provide a name such as ADFS for the rule. Specify the external URL, the external certificate, and the back-end server URL as shown in Figure 11.



Name:

ADFS

This name will appear in the list of published web applications.

External URL:

https://sts.example.com

External certificate:

*.example.com

Backend server URL:

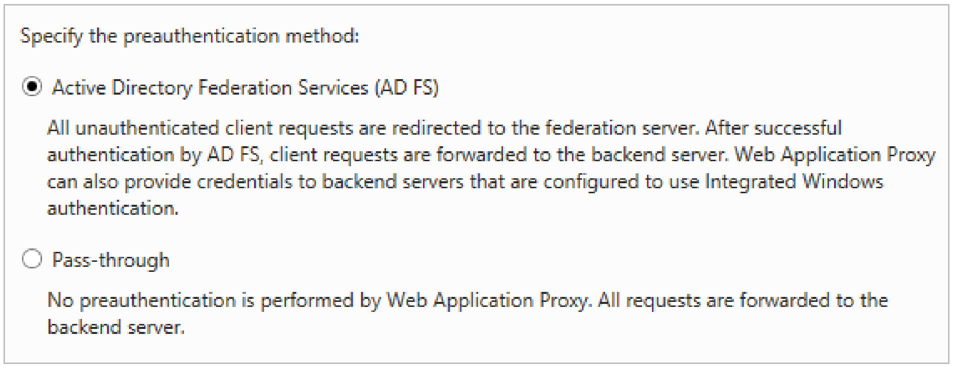
https://adfs1.example.com

**Figure 11: Configuring the Publishing Rule**

**Note** If you've implemented internal load balancing for the AD FS tier, you can set the back-end server URL to a load-balanced endpoint instead of an individual server name.

17. Choose **Publish,** and then **Close** to exit the wizard.

18. Choose **Publish** again to create a new rule for OWA. This time, set the pre-authentication method to **Active Directory Federation Services (AD FS)**, and then choose **Next**.



Specify the preauthentication method:

● Active Directory Federation Services (AD FS)

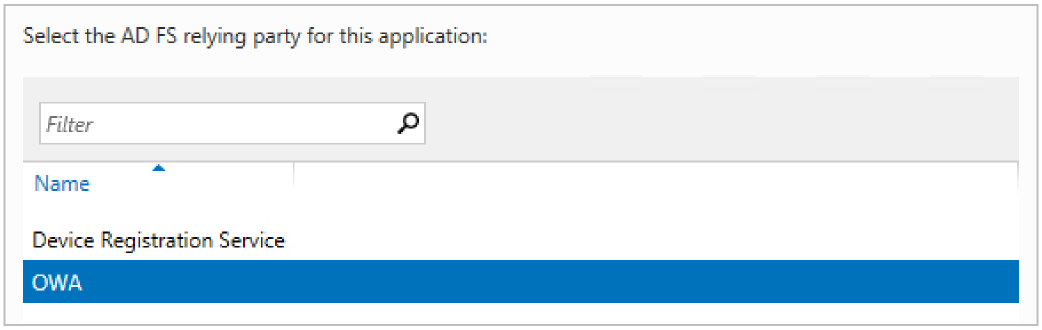
All unauthenticated client requests are redirected to the federation server. After successful authentication by AD FS, client requests are forwarded to the backend server. Web Application Proxy can also provide credentials to backend servers that are configured to use Integrated Windows authentication.

○ Pass-through

No preauthentication is performed by Web Application Proxy. All requests are forwarded to the backend server.

**Figure 12: Selecting the AD FS Pre-Authentication Method**

19. For the relying party for the application, select the relying party trust you created on the AD FS server, and then choose **Next**.

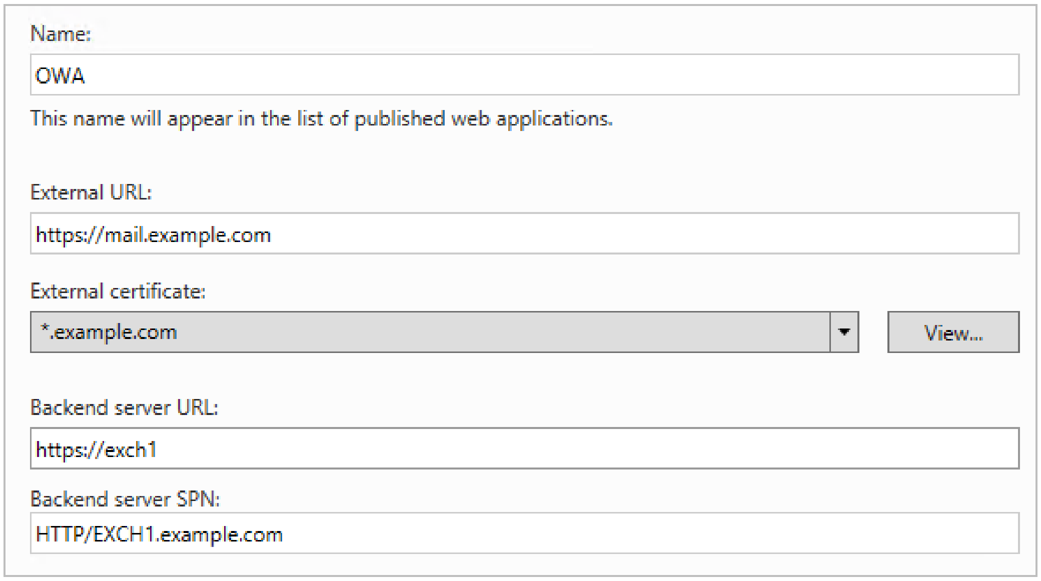


Select the AD FS relying party for this application:

| Name |
|---|
| Device Registration Service |
| OWA |

**Figure 13: Selecting the Relying Party**

20. Provide a name such as OWA for the rule. Specify the external URL, external certificate, back-end URL, and service principal name (SPN) for the back-end server, as shown in Figure 14.



Name:

OWA

This name will appear in the list of published web applications.

External URL:

https://mail.example.com

External certificate:

*.example.com

Backend server URL:

https://exch1

Backend server SPN:

HTTP/EXCH1.example.com

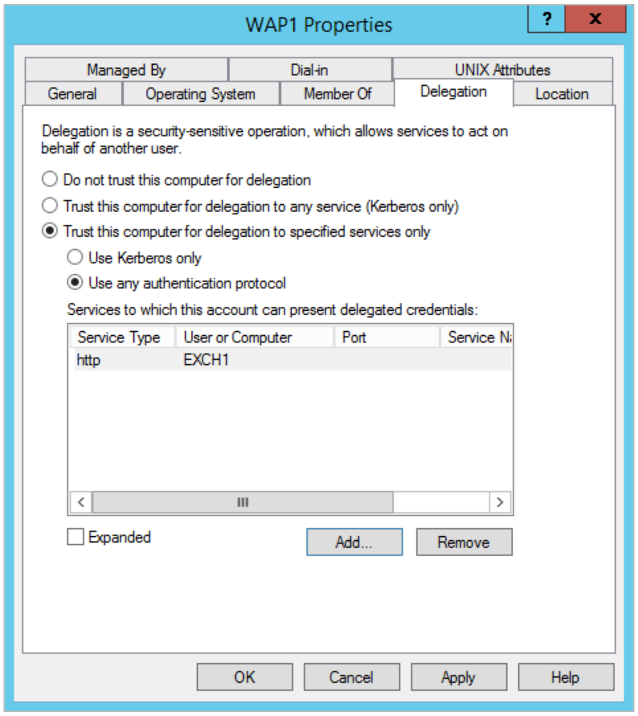
Figure 14: Configuring Rule Details

**Note** If you've implemented internal load balancing for the Exchange client access tier, you can set the back-end server URL and SPN to a load-balanced endpoint instead of an individual server name.

21. Choose **Publish** and close the wizard.

22. Establish an RDP connection to DC1. In **Control Panel**, choose **Administrative Tools**, and then launch the **Active Directory Users and Computers** snap-in.

23. Navigate to the **Computers** container, right-click the WAP1 computer, and then choose **Properties**. On the **Delegation** tab, choose **Trust this computer for delegation to specified services only**. Check the option to use any authentication protocol, and add the HTTP service type on the EXCH1 computer to the list, as shown in Figure 15. Choose **Apply**, and then choose **OK**.



Delegation is a security-sensitive operation, which allows services to act on behalf of another user.

○ Do not trust this computer for delegation

○ Trust this computer for delegation to any service (Kerberos only)

● Trust this computer for delegation to specified services only

- ○ Use Kerberos only
- ● Use any authentication protocol

Services to which this account can present delegated credentials:

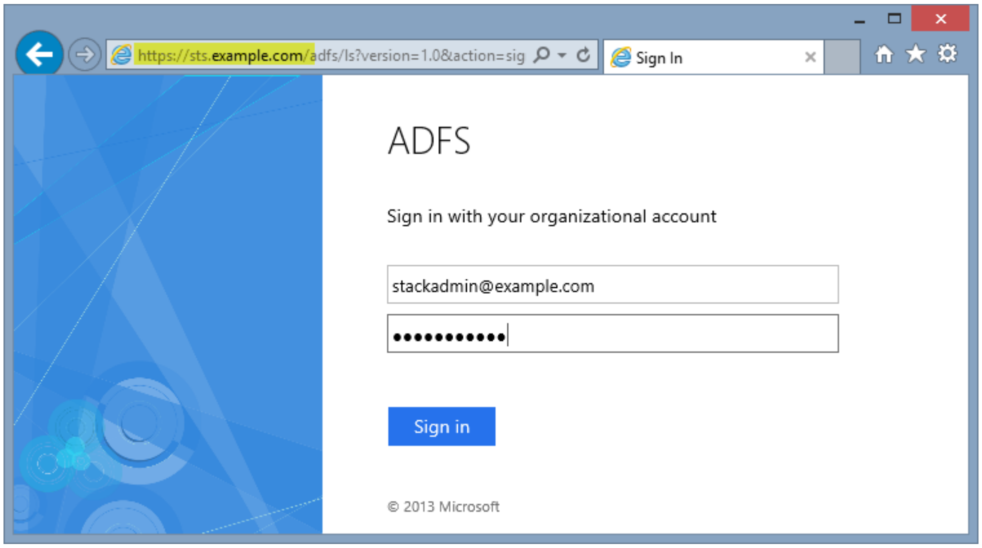
| Service Type | User or Computer | Port | Service N |
|---|---|---|---|
| http | EXCH1 | | |

□ Expanded

Figure 15: Configuring Kerberos Constrained Delegation

Now you are ready to test accessing OWA from an external workstation or server over the Internet.

24. If you did not use your own domain name, you'll need to edit the hosts file on your machine to allow your computer to resolve the endpoints at example.com: Add a mapping for **sts.example.com** and **mail.example.com** to your local hosts file, making sure that both hosts resolve to the public EIP of the WAP1 server.
25. Open a web browser from your external workstation or server. Navigate to **mail.example.com**. You should be redirected to the federation service and prompted for authentication. Provide the stackadmin user name and password, and then choose **Sign in**.



https://sts.example.com/adfs/ls?version=1.0&action=sig

ADFS

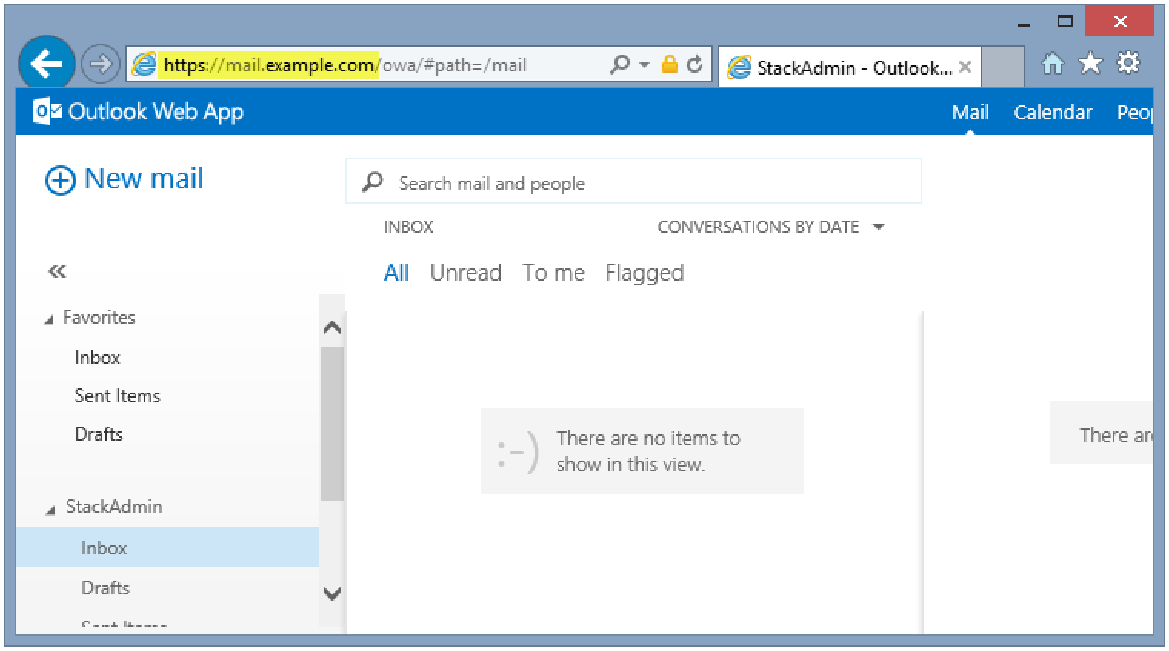
Sign in with your organizational account

stackadmin@example.com

••••••••••••

Figure 16: Pre-Authenticating to AD FS

If the authentication is successful, the connection should be proxied to the EXCH1 server through the Web Application Proxy, as shown in Figure 17.



https://mail.example.com/owa/#path=/mail

There are no items to show in this view.

Figure 17: Connected to the Published Application

# Send Us Feedback

We welcome your questions and comments. Please post your feedback on the AWS Quick Start Discussion Forum.

# Document Revisions

| Date | Change | In sections |
|---|---|---|
| **September 2015** | In the sample templates, changed the default type for Active Directory and RD Gateway instances from m3.xlarge to m4.xlarge for better performance and price. | Step 2 (template customization table) |
| **August 2015** | Initial publication | – |

**Notices**

This document is provided for informational purposes only. It represents AWS's current product offerings and practices as of the date of issue of this document, which are subject to change without notice. Customers are responsible for making their own independent assessment of the information in this document and any use of AWS's products or services, each of which is provided "as is" without warranty of any kind, whether express or implied. This document does not create any warranties, representations, contractual commitments, conditions or assurances from AWS, its affiliates, suppliers or licensors. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.