



# ***Deploy a Microsoft SharePoint 2010 Server Farm in the AWS Cloud in 6 Simple Steps***

*Oct 2012*

*Ulf Schoo*

## Notes

Please consult <http://aws.amazon.com/articles/9982940049271604> for the latest version of this article.

For a complete video walk-through, go to <http://aws.amazon.com/windows/sharepoint/gsg-sharepoint-2010/>.

Have questions about your specific project? Please contact us at <http://aws.amazon.com/sharepoint/contact-us/>.



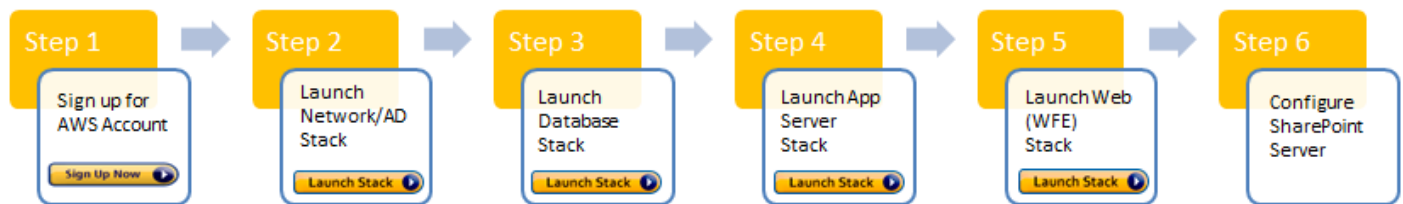
## Abstract

Learn how you can launch a Microsoft SharePoint 2010 server farm for a public facing website on Amazon Web Services (AWS) cloud infrastructure in 6 simple steps. This article provides all the necessary resources, including easy-to-launch AWS CloudFormation sample templates, and instructions on how to create Amazon Machine Images (AMIs) so that you can launch a fully functional highly-available Microsoft SharePoint 2010 server farm on AWS.

## Overview

Whether it's a single server SharePoint Foundation solution or a full featured SharePoint web application running on a multi-tiered SharePoint server farm designed for high availability (HA), Amazon Web Services (AWS) provides a complete set of services and tools for deploying Microsoft Windows based workloads, including Microsoft SharePoint Server, on its reliable, scalable and cost-effective cloud infrastructure.

We recommend that you to read the [Microsoft SharePoint Server on AWS Reference Architecture](#) whitepaper that describes in detail common reference architectures for different SharePoint Server deployment scenarios (primarily, public-facing Internet website and Intranet application). In this article, you will learn how you can launch a Microsoft SharePoint 2010 server farm for the public facing web site scenario. We provide all the necessary resources, including easy-to-launch AWS CloudFormation sample templates, and instructions on how to create Amazon Machine Images so you can launch the fully functional sample stack using the AWS Management Console. If you like to customize the provisioning and configuration steps, we also provide these resources (templates) in download form, and have published an [advanced guide](#) so you can customize the templates as per your needs and deploy them in the cloud repeatedly and reliably using AWS CloudFormation and other tools such as the Windows PowerShell or the command line tools.

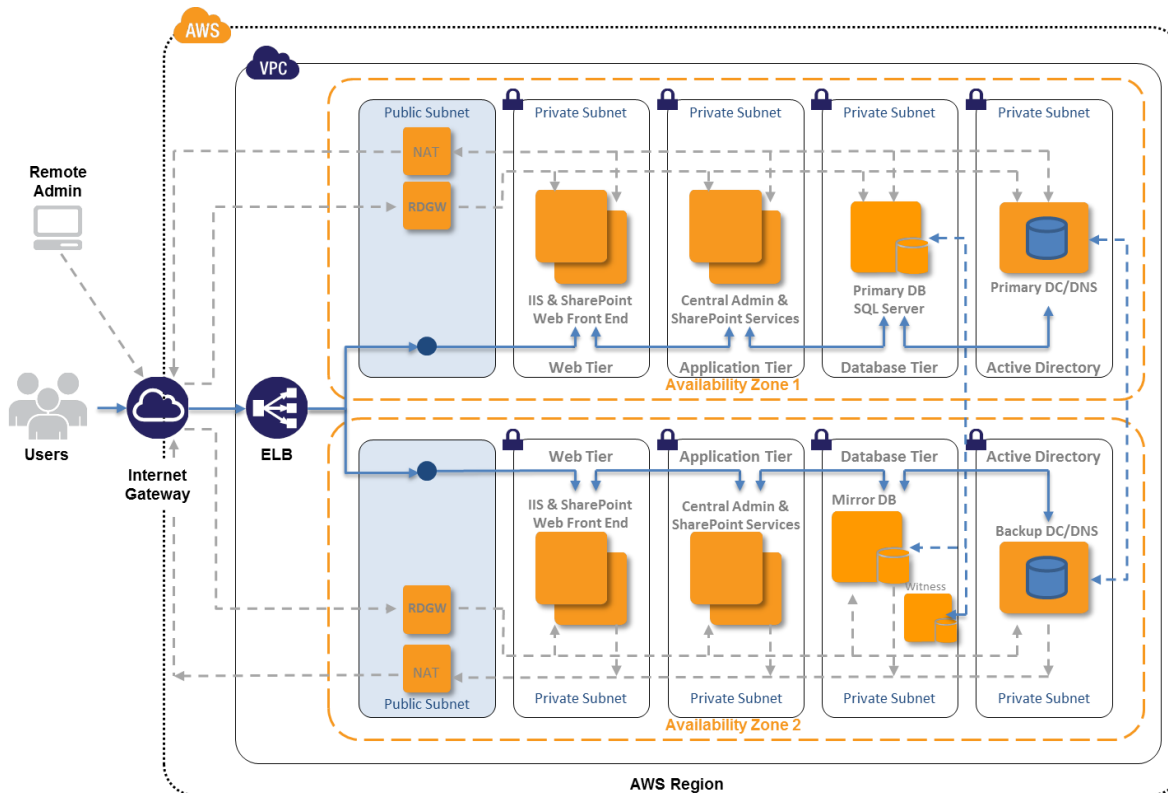


To launch a sample Microsoft SharePoint Server 2010 Farm for a public facing web site in the AWS cloud, all you have to do is the following:

1. **Sign up** for an AWS account (there is no charge for opening an AWS account).
2. Launch the **network and Active Directory stack** using the sample template (Template-1). This creates the necessary network configuration and Active Directory foundational infrastructure.
3. Launch the **database stack** using the sample template (Template-2). This uses a custom AMI and deploys and provisions the database tier.
4. Launch the **app stack** using the sample template (Template-3). This uses a custom AMI and deploys a Windows Server 2008 R2 EC2 instance and installs SharePoint Server 2010 for Internet Sites - Enterprise on your **APP server** using your own license key (BYOL) or, for a trial or test installation, a trial PID key.
5. Launch the **web stack** using the sample template (Template-4). This deploys a Windows Server 2008 R2 EC2 instance and installs SharePoint Server 2010 for Internet Sites - Enterprise on your **WFE servers** using your own license key (BYOL) or, for a trial or test installation, a trial PID Key. It also deploys and configures an Elastic Load Balancer (ELB) which will balance traffic between your WFE servers.

6. **Configure** your SharePoint farm (to get a sample Team Site for a proof of concept or demonstration).
7. You are done!

This enables you to launch relevant and related AWS resources and ultimately build out a sample stack as shown in the diagram below:



**Figure 1: SharePoint server reference architecture for public-facing website scenario**

For the purpose of this article, we will deploy:

1. A VPC spanning 2 Availability Zones (AZ) with 10 subnets (8 private subnets and 2 public subnets)
2. 2 Microsoft Windows Servers 2008 R2 that serve as Active Directory and DNS Servers
3. 1 Microsoft SQL Server Standard 2008 R2 serving as the database tier
4. 1 Microsoft Windows Server 2008 R2 serving as the App server that hosts the SharePoint Central Administration and SharePoint Service
5. 2 Microsoft Windows Server 2008 R2 (1 in each AZ) serving as the Web Front End (WFE) servers
6. 2 Microsoft Windows Server 2008 R2 (1 per DMZ in each AZ) serving as Remote Desktop Gateway (RDGW) servers
7. 2 NAT instances
8. 1 Internet Gateway and 1 Elastic Load Balancer (ELB)

## Step 1. Sign up For an AWS Account

When you create an AWS account, AWS automatically signs up the account for all AWS services, including Amazon EC2. You are charged only for the services that you use. If you already have an AWS account, skip to the next step. If you don't already have an AWS account, use the following procedure to create one.

To create an AWS account, Go to <http://aws.amazon.com>, and click Sign Up Now. Follow the on-screen instructions. Part of the sign-up process involves receiving a phone call and entering a PIN using the phone keypad.

## Step 2. Launch the Network and Active Directory Stack using the Sample Template (Template-1)

Once you have signed up for your account we are ready to provision our networking and Active Directory infrastructure. To do that, click the "Launch Stack" button.

Launch in your preferred region:



[US East \(Northern Virginia\) Region](#)  
[US West \(Northern California\) Region](#)  
[US West \(Oregon\) Region](#)

[EU \(Ireland\) Region](#)  
[Asia Pacific \(Singapore\) Region](#)  
[Asia Pacific \(Tokyo\) Region](#)  
[South America \(Sao Paulo\) Region](#)

This will take you to the AWS Management Console's AWS CloudFormation stack wizard and load the sample Template-1. AWS CloudFormation is a service that will provides you with an easy way to create a collection of related and relevant AWS resources and provision them in a repeatable and predictable fashion using a simple JSON template.

**Create Stack** Cancel X

SELECT TEMPLATE    SPECIFY PARAMETERS    REVIEW

AWS CloudFormation gives you an easier way to create a collection of related AWS resources (a stack) by describing your requirements in a template. To create a stack, fill in the name for your stack and select a template. You may chose one of the sample templates to get started quickly, or one of you own templates stored in S3 or on your local hard drive.

**Stack Name:**

**Stack Template Source:**

Use a sample template

Upload a Template File

Provide a Template URL

Show Advanced Options

**Continue** ▶

Figure 2: Create Stack wizard

The AWS CloudFormation sample Template-1 will create all the necessary AWS resources, network configuration and Active Directory configuration for your SharePoint server farm deployment. Since no two SharePoint server deployments are the same and you may want to configure, for example, your network address schema to more closely align with an existing on-premises deployment or your domain administrator account and password, you will need to input or modify a few parameters before you can launch the stack. Alternatively, if you create this stack for proof of concept or demonstration purposes you can just simply accept the defaults.

Creating a stack from our sample Template-1 will require, amongst other things, to input an AWS Key Pair name, which is a public/private key pair that allows you to securely connect to your instance after it launches. For a short tutorial on how to create a new Key Pair Name, see this [YouTube Video](#).

Also, our sample Template-1 allows for modifying the instance types for your Active Directory and Remote Desktop Gateway instances. For more information on visit the [Amazon EC2 instance types](#).

**Create Stack** Cancel

SELECT TEMPLATE    **SPECIFY PARAMETERS**    REVIEW

**Template Description:** **\*\*BETA Version 0.4\*\*** STEP-1 This template creates a VPC infrastructure for a multi-AZ, multi-tier deployment of a Windows based Application infrastructure. It installs 2 Active Directory Domain Controllers into subnets in separate AZs inside a VPC, as well as Remote Desktop Gateway and NAT instances into the DMZ for each Availability Zone. The default Domain Administrator password will be the one retrieved from the instance. For adding members to the domain, ensure that they are launched into the domain member security group created by this template and then configure them to use the AD instances fixed private IP addresses as the DNS server. **\*\*WARNING\*\*** This template creates Amazon EC2 Windows instance and related resources. You will be billed for the AWS resources used if you create a stack from this template.

**Specify Parameters**

Below are the parameters associated with your CloudFormation template. You may review and proceed with the default parameters or make customizations as needed below.

**SPSAdminUser**   
User name for the SharePoint Server Admin Account. This Account is a Domain User and will also be added to the SQL DB as a member of the dbcreator role.

**PrivSub6CIDR**   
CIDR Block for Private Subnet 6 located in AZ2

**SPSAdminPassword**   
Password for the SPS admin user. Must be at least 8 characters containing letters, numbers and symbols

**PrivSub7CIDR**   
CIDR Block for Private Subnet 7 located in AZ2

**PrivSub3CIDR**   
CIDR Block for Private Subnet 3 located in AZ1

**UserCSVSourceLocation**   
An accessible source location of a Comma Delimited Values (CSV) file containing any

**I acknowledge that this template may create IAM resources**

[< Back](#) Check this to acknowledge that this template can create IAM (Identity and Access Management) resources. If you do not check this box, templates that require creating IAM resources will fail. Check this box only on templates that you trust.

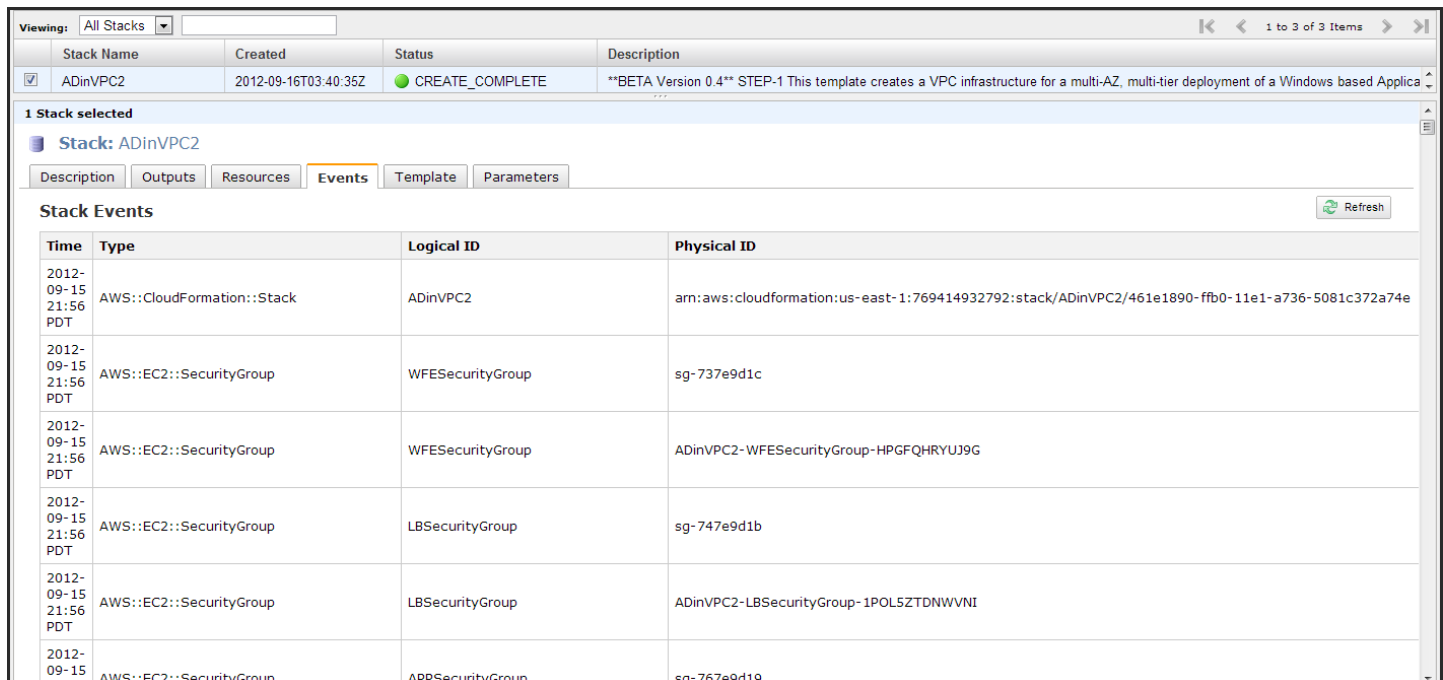
**Figure 3: Create Stack Wizard, review and modify parameters**

The sample template will launch and provision all the defined cloud resources (such as creating the VPC, subnets, configuring security groups) but it also performs fine-grained configuration tasks on the Windows Instances (such as creating the physical representation of the deployed Active Directory infrastructure by creating AD sites, subnets and site links). These configuration tasks on the instances themselves are performed using Windows PowerShell scripts. Those PowerShell scripts provide limited functionality and are not meant to represent a production ready solution. Instead, they are intentionally built as mesh-ups of samples freely available on the usual PowerShell community sites and meant as examples of how you can use AWS CloudFormation and PowerShell to reach deep into your instances at provisioning time and perform the necessary configuration steps. In all likelihood, you will want to (and can) replace the scripts with your own.

The architecture diagram (Figure 6) below shows all the infrastructure and resources that will be launched behind the scenes by our AWS CloudFormation sample template.

NOTE: You can use the sample templates discussed in this article as-is, modify them or use them as a starting point for your own template.

If you want to follow along and see what's happening behind the scenes you can go to the AWS CloudFormation console (<https://console.aws.amazon.com/cloudformation>), select the stack you just launched and click the Events tab. You can hit the refresh button frequently to follow along in the creation process. (Figure 4)



Time	Type	Logical ID	Physical ID
2012-09-15 21:56 PDT	AWS::CloudFormation::Stack	ADinVPC2	arn:aws:cloudformation:us-east-1:769414932792:stack/ADinVPC2/461e1890-ffb0-11e1-a736-5081c372a74e
2012-09-15 21:56 PDT	AWS::EC2::SecurityGroup	WFESecurityGroup	sg-737e9d1c
2012-09-15 21:56 PDT	AWS::EC2::SecurityGroup	WFESecurityGroup	ADinVPC2-WFESecurityGroup-HPGFQHRUJ9G
2012-09-15 21:56 PDT	AWS::EC2::SecurityGroup	LBSecurityGroup	sg-747e9d1b
2012-09-15 21:56 PDT	AWS::EC2::SecurityGroup	LBSecurityGroup	ADinVPC2-LBSecurityGroup-1POL5ZTDNWWNI
2012-09-15	AWS::EC2::SecurityGroup	APPSecurityGroup	sg-767e9d19

**Figure 4: AWS CloudFormation Manager Console, View Events Tab**

After the stack is launched, it will provide you with the Elastic IP Addresses (EIP) of the 2 Remote Desktop Gateways (RDGW1 + RDGW2) when you select the Outputs Tab. You will use this IP address to connect to your RDGW instance and then to your Domain Controller to see if all configurations have been performed correctly or to perform additional Active Directory or DNS configuration tasks. (Figure 5)

Stack Name	Created	Status	Description
<input checked="" type="checkbox"/> ADinVPC2	2012-09-16T03:40:35Z	CREATE_COMPLETE	**BETA Version 0.4** STEP-1 This template creates a VPC infrastructure for a multi-AZ, multi-tier deployment of a Windows based Applica...

1 Stack selected

Stack: ADinVPC2

Description | **Outputs** | Resources | Events | Template | Parameters

**Stack Outputs**  
Output values may have been specified by the template author and will be available when stack creation is complete. Refresh

Key	Description	Value
RDGW1ElasticIP	Elastic IP address of the first Remote Desktop Gateway (RDGW1) in AZ1	107.23.66.217
RDGW2ElasticIP	Elastic IP address of the second Remote Desktop Gateway (RDGW2) in AZ2	107.23.60.94
DomainAdmin	Default domain administrator account	contoso\Administrator
DomainAdminPassword		Please retrieve Administrator password of the instance

Figure 5: AWS CloudFormation Manager Console, View Outputs Tab

At the end of this step, you will have the following resources of our architecture launched:

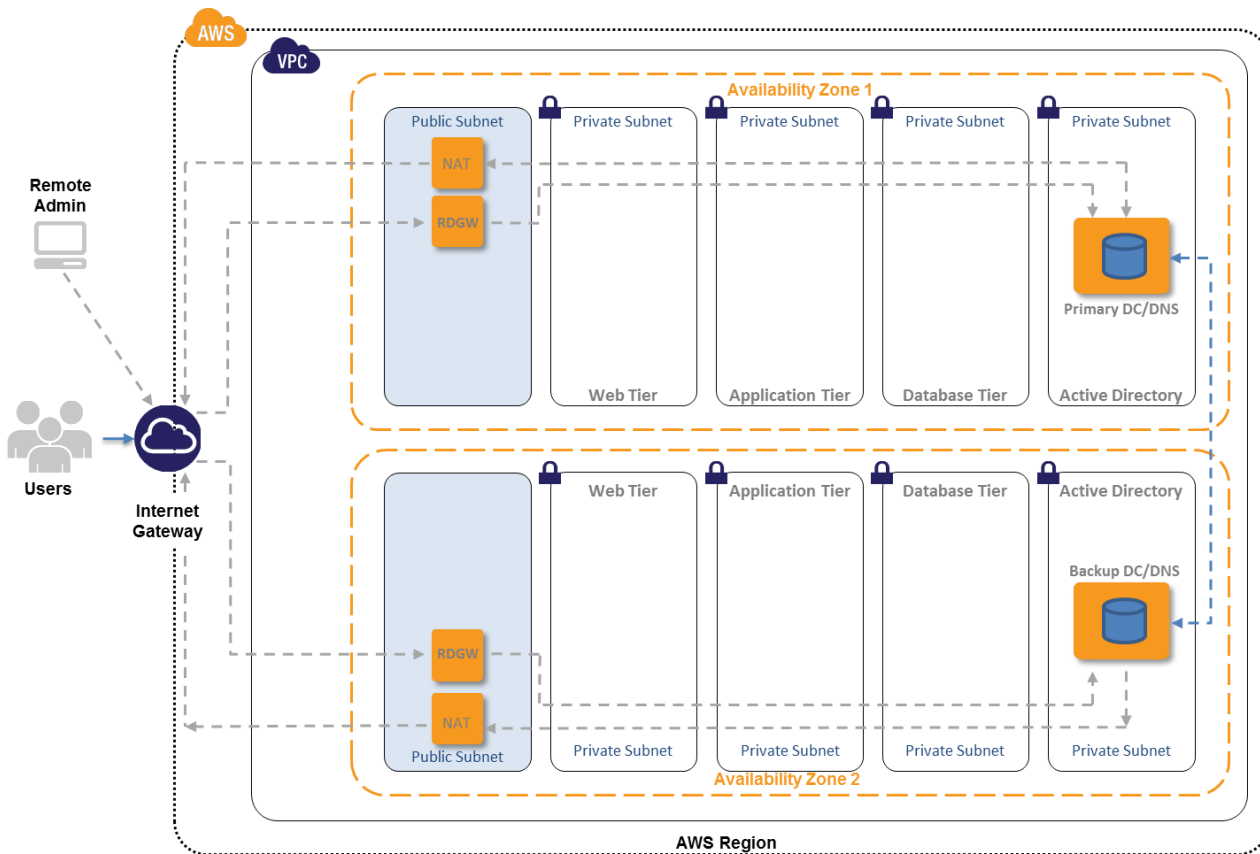


Figure 6 : Architecture implemented at the completion of step 2

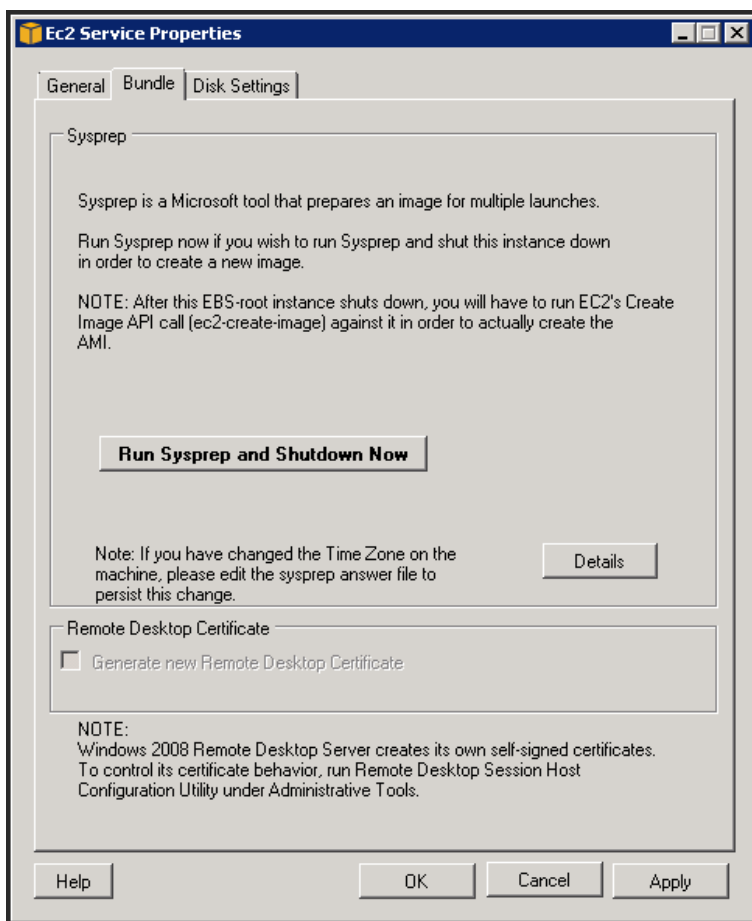
### Step 3. Launch the Database Stack using the Sample Template (Template-2)

Before we can move on and launch the next stack, which will deploy SQL Server 2008 R2 Standard Edition as the database for our database tier, we have to perform a handful of manual steps to create our own CloudFormation-

enabled Microsoft SQL Server Standard 2008 R2 AMI. This will enable us to use AWS CloudFormation and Windows PowerShell to configure the database for our SharePoint Server farm.

The current SQL AMI's published by AWS are not CloudFormation-enabled. In order to use a standard SQL AMI in our scripted infrastructure deployment, we will have to create a SQL AMI that runs the EC2Config Service and the CFN Helper scripts. The steps to create such an AMI are as follows:

1. Start with the latest AWS published SQL 2008 R2 AMI
2. Launch up the instance as a standalone instance (any instance type larger than t1.micro works).
3. Install the latest **IronPythonized CFN tools** MSI package from [here](#)
4. Manually reset the UserData flag in C:\Program Files\Amazon\Ec2ConfigService\Settings\Config.xml
  - <Name>Ec2HandleUserData</Name>
  - <State>Enabled</State>
5. Manually create the PATH variable in Computer/Properties/Advanced System Settings/Advanced/Environment Variables. Under **System variables (\*NOT\* User Variables)** add
  - Path=C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0\;C:\Program Files (x86)\Amazon\cfn-bootstrap



**Figure 7: EC2 Service Properties, Run Sysprep**

6. Use Sysprep from within the EC2Config app to bundle everything up. (Figure 7) Wait until the instance shows up as \*STOPPED\* in the AWS Management Console

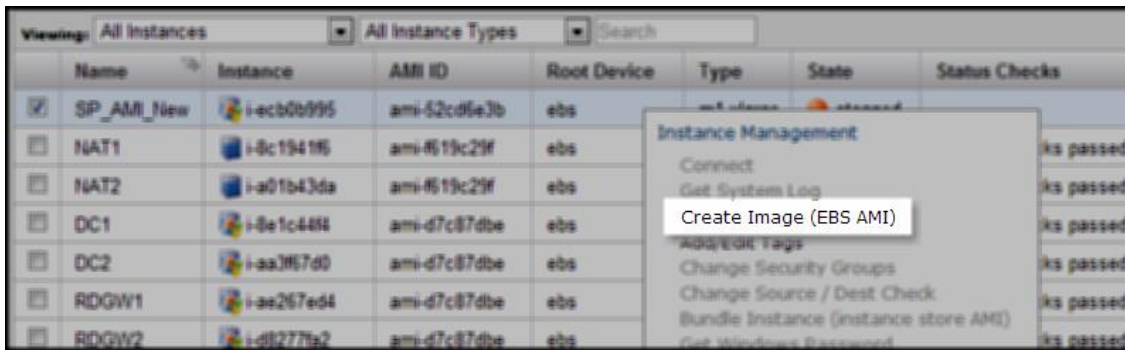


Figure 8: Amazon EC2 Console - Create Image (EBS AMI)

7. Create the AMI image using the AWS EC2 Management Console (Figure 8)
8. If you haven't done so already, now you should quickly look up and write down the ID of the CloudFormation-enabled SQL AMI. We will need this ID in our next step below. To look up the AMI ID we navigate to the EC2 Console and then AMIs and take note of the AMI ID.

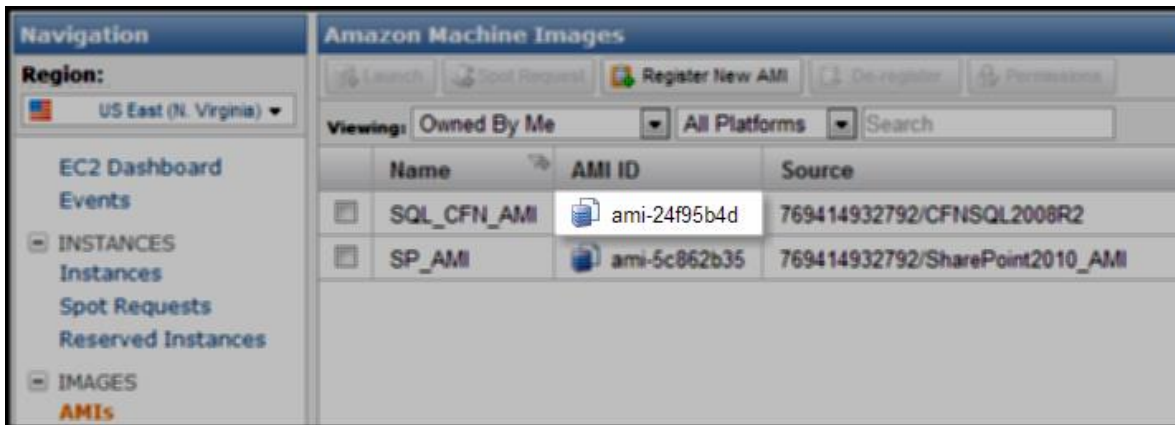


Figure 9: Amazon EC2 Console: Amazon Machine Images

Now that you have created your CloudFormation-enabled SQL Server 2008 R2 Standard Edition AMI you are ready to move on to the next step and click the "Launch Stack" button to deploy and provision the database tier.

Launch in your preferred region:



[US East \(Northern Virginia\) Region](#)  
[US West \(Northern California\) Region](#)  
[US West \(Oregon\) Region](#)

[EU \(Ireland\) Region](#)  
[Asia Pacific \(Singapore\) Region](#)  
[Asia Pacific \(Tokyo\) Region](#)  
[South America \(Sao Paulo\) Region](#)

This will take you to the AWS Management Console's AWS CloudFormation Stack Wizard and load the sample Template-2.

The AWS CloudFormation sample Template-2 will launch the database stack into the previously deployed networking and Active Directory infrastructure and join the Database Server to the domain. In addition to the previously recorded SQL AMI ID, you will need to look up the following IDs in the VPC console and input them into the template.

- VPC ID

- SQL Server Security Group ID
- Domain Member Security Group ID
- SQL Server Subnet ID

**Create Stack** Cancel X

SELECT TEMPLATE    **SPECIFY PARAMETERS**    REVIEW

**Template Description:** **BETA Version 0.4** STEP-2 This template creates and installs a CloudFormation enabled SQL Server 2008 R2 into a subnet inside a VPC. It attached two additional striped volume sets, Volume D and E, to hold the SharePoint Content and Configuration databases and the Log files. Default Domain Administrator password will be the one retrieved from the instance. For adding members to the domain, ensure that they are launched in the domain member security group created by the Step-1 Infrastructure template and then configure them to use the AD instances private IP addresses as the DNS server. **WARNING** This template creates Amazon EC2 Windows instance and related resources. You will be billed for the AWS resources used if you create a stack from this template.

**Specify Parameters**  
Below are the parameters associated with your CloudFormation template. You may review and proceed with the default parameters or make customizations as needed below.

**DomainDNSName**   
Fully qualified domain name (FQDN) of the forest root domain e.g. corp.example.com

**SQLSvrSubnet**   
ID of the Subnet you want to provision the SQL Server into

**DomainMemberSGID**   
ID of the Domain Member Security Group

**DomainAdminPassword**   
Password for the domain admin user. Must be at least 8 characters containing letters, numbers and symbols

**KeyPairName**   
Public/private key pairs allow you to securely connect to your instance after it launches

**DomainAdminUser**   
User name for the account that will be added as Domain Administrator. This is separate

**I acknowledge that this template may create IAM resources**

< Back    **Continue** >

**Figure 10: Create Stack wizard: review and modify parameters**

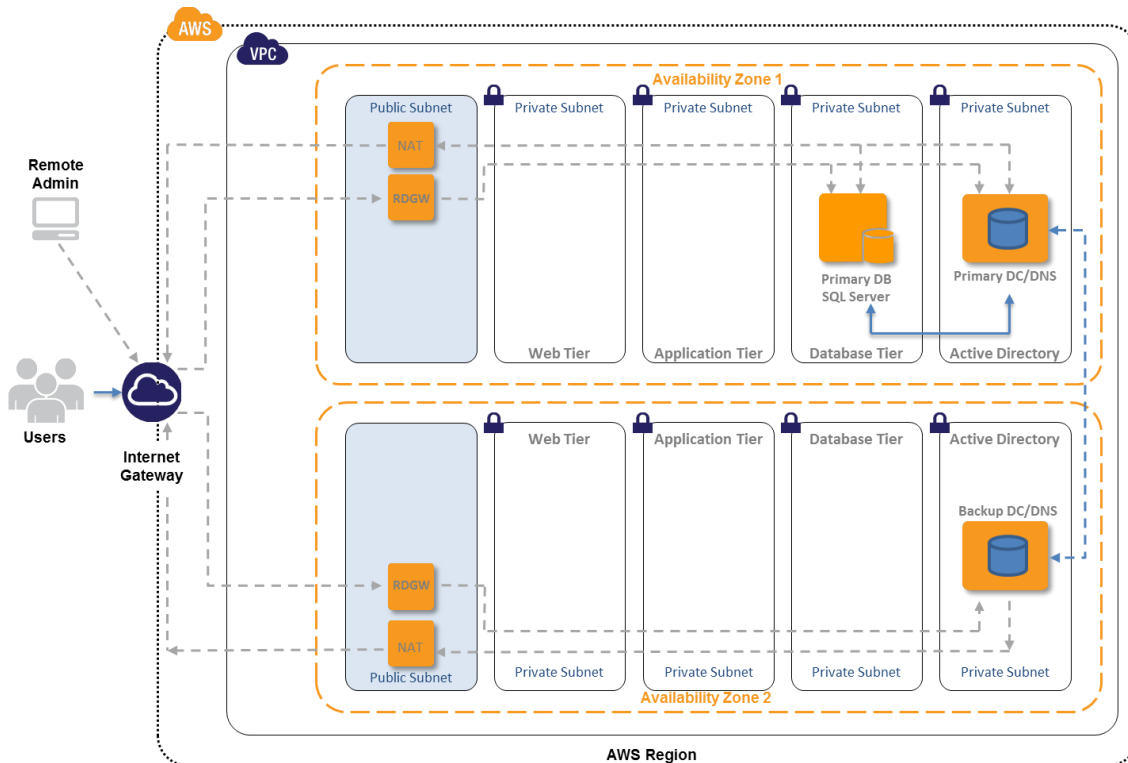
In addition to the volume that holds the operating system, your SQL instance will have, multiple volumes attached, one for the SharePoint database files and one for the log files. This follows in part Microsoft's recommendations for mitigating any potential IO contention. (NOTE: The drive letters for the database (D:\) and log file (E:\) volume are fixed per the sample template but you can modify the directory name.) You also want to ensure consistent SQL Server IO performance that is in line with the desired performance profile of your SharePoint Application. Therefore, sample Template-2 creates volumes provisioned with a specific number of I/O operations per second (IOPS). These Provisioned IOPS volumes can be attached to specially "EBS-optimized" instance types. Currently, the following instance types can be launched as EBS-optimized instances:

- Large (m1.large)
- Extra Large (m1.xlarge)
- High-Memory Quadruple Extra Large (m2.4xlarge)

Sample **Template-2** also executes a PowerShell script that creates a SQL login for the SharePoint farm administrator account and adds this login to the *dbcreator* and *securityadmin* role, as required for a successful SharePoint configuration.

After the stack is launched (CREATE\_COMPLETE), it will provide you with the SQL Server NetBIOS name. You will use this NetBIOS name to connect from your RDGW instance to your SQL server to see if all configurations have been performed correctly or to perform additional SQL server configuration tasks. In Step 6 you will also use the SQL Server NetBIOS name when you configure your SharePoint Server Farm and connect from your APP and WFE servers to the SQL server.

At the end of this step you will have the following resources of our architecture launched:

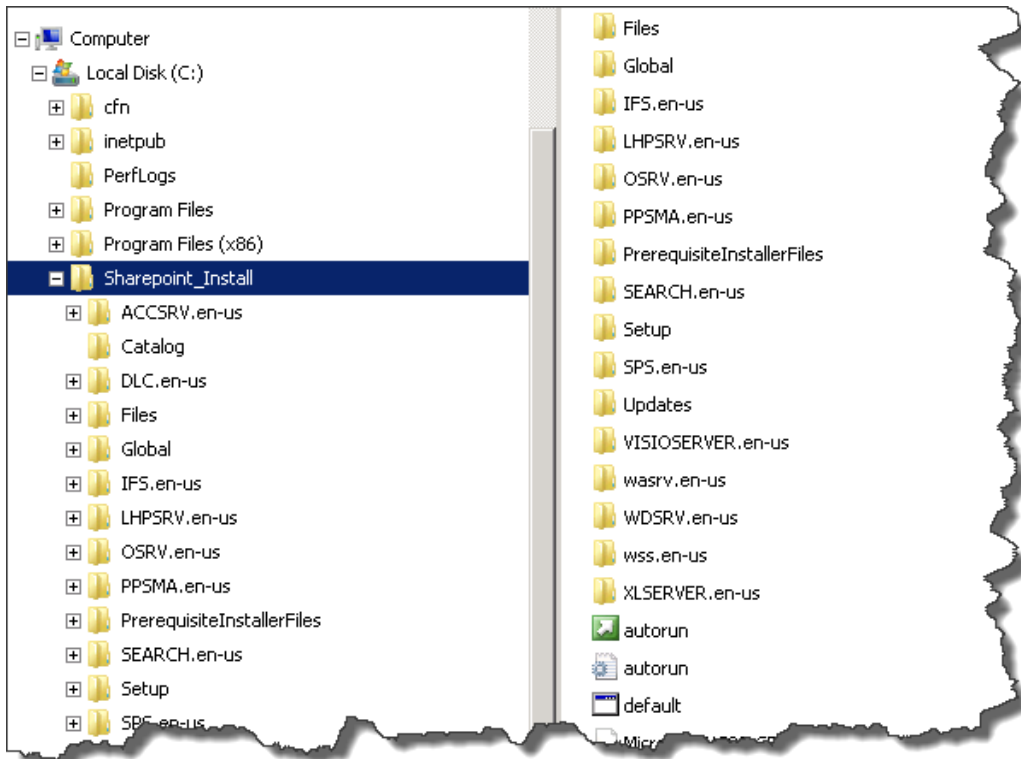


**Figure 11: Architecture implemented at the completion of step 3**

**Step 4. Launch the Application Stack using the Sample Template (Template-3)**

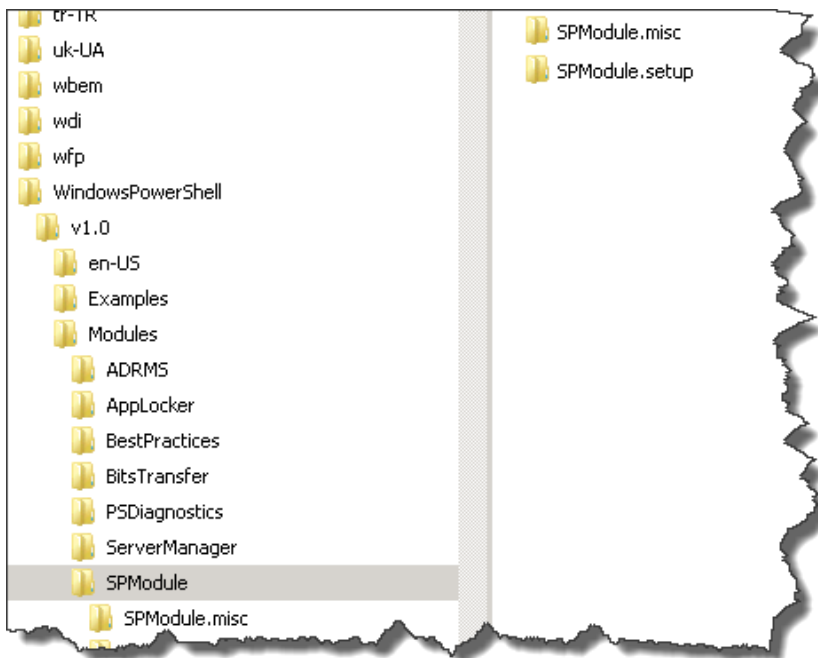
For both, the App Server and the Web Front End Servers, we need to create our own Windows Server AMI that holds the raw (uninstalled) SharePoint bits and the SPModule. The steps for creating such an AMI are as follows:

1. Start with the latest AWS published [Windows Server 2008 R2 English 64-bit - Base for CloudFormation](#) AMI.
2. Launch the Instance as a standalone Instance (any instance type larger than t1.micro works).
3. On the launched instance, download SharePoint Server 2010 for Internet Sites, Enterprise from [here](#) and copy the raw (uninstalled) bits to a directory c:\SharePoint\_install.



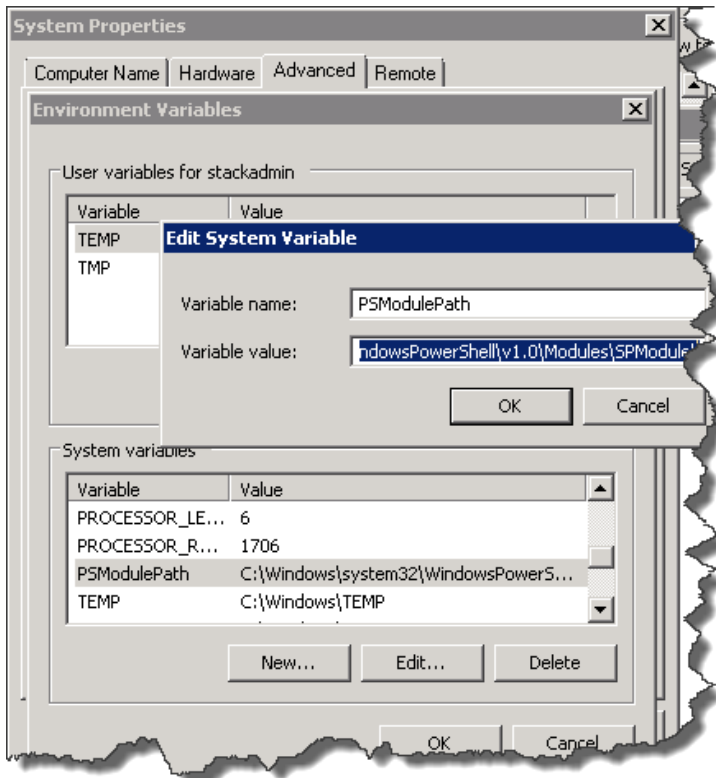
**Figure 12: Download SharePoint Server 2012**

4. On the launched instance, download the SPMModule.zip file from [here](#) and extract it to C:\Windows\System32\WindowsPowerShell\v1.0\Modules\SPModule.



**Figure 13: Extracted SPMModule location**

5. Create/Edit the PSModulePath in the System Variables to point to the location where you copied the files in Step 4.



**Figure 14: Configure System Variables**

6. After that manually reset the UserData flag in C:\Program Files\Amazon\Ec2ConfigService\Settings\Config.xml
  - <Name>Ec2HandleUserData</Name>
  - <State>**Enabled**</State>
7. Use Sysprep from within the EC2Config app to bundle everything up. Wait until the instance shows up as **\*STOPPED\*** in the management console.
8. Create the AMI image using the AWS Management Console.

At this point in the process you have launched the resources that make up your network & security infrastructure and your Active Directory and database. Besides small, individual configurations like adding a SharePoint Farm Administrator account or creating a login for that account on your SQL Server database the infrastructure and resources we have launched so far are not specific to SharePoint. They could easily be the backbone for a Microsoft Windows based 3-tier application of any scale.

After the completion of this step you have taken the first of the final 3 steps that make this a SharePoint Server Farm running on the AWS cloud. To proceed, click the "Launch Stack" button.

Launch in your preferred region:



[US East \(Northern Virginia\) Region](#)  
[US West \(Northern California\) Region](#)  
[US West \(Oregon\) Region](#)

[EU \(Ireland\) Region](#)  
[Asia Pacific \(Singapore\) Region](#)  
[Asia Pacific \(Tokyo\) Region](#)  
[South America \(Sao Paulo\) Region](#)

This will take you to the AWS Management Console's AWS CloudFormation stack wizard and load the sample Template-3.

The AWS CloudFormation sample Template-3 will launch the APP server stack into the previously deployed networking, Active Directory and database infrastructure and join the APP server to the domain. Just as you looked up IDs when launching the SQL Server stack, you need to look up the following IDs in the VPC console and input them into the template.

- VPC ID
- APP Server Security Group ID
- Domain Member Security Group ID
- APP Server Subnet ID

This deploys a Windows Server 2008 R2 Amazon EC2 instance and installs SharePoint Server 2010 for Internet Sites - Enterprise on your APP server. You will have to provide your own license key (BYOL) or, for a trial or test installation, get the Trial PID Key directly from Microsoft SharePoint Trial Site at <http://www.microsoft.com/en-us/download/details.aspx?id=16631>. Please note that you are responsible for complying with Microsoft's requirements for your use of the SharePoint 2010 trial. Microsoft may discontinue or change the requirements around the SharePoint 2010 trial at any time. Refer to Microsoft's [License Mobility through Software Assurance](#) program for further details on how you can provide your mobilized SharePoint license into the deployment process.

**Specify Parameters**  
Below are the parameters associated with your CloudFormation template. You may review and proceed with the default parameters or make customizations as needed below.

**AppServerSecurityGroup**   
ID of the App Server Security Group

**SPSAdminPassword**   
Password for the SPS admin user. Must be at least 8 characters containing letters, numbers and symbols

**SPLicensekey**   
A valid SharePoint PID in the format VK7BD-VBKWR-6FHD9-Q3HM9-6PKMX. You can download a SharePoint 2010 trial and trial PID from <http://www.microsoft.com/en-us/download/details.aspx?id=16631>

**AppSVR1InstanceType**

**Figure 15: Providing your own license key (BYOL)**

After the stack is launched (CREATE\_COMPLETE), it will provide you with the APP Server NetBIOS name. You will use this NetBIOS name to connect from your RDGW instance to your APP server to see if all configurations have been performed correctly. In Step 6 you will log into the APP server using its NetBIOS name when you create and configure your SharePoint server farm Central Administration site.

If your deployment and desired farm capacity requires it you can repeat step 5 and launch additional APP servers to create App Server groups either by launching additional App server stacks into the same subnet or, taking advantage of the provided high availability infrastructure, into a subnet in the second Availability Zone (AZ). To accomplish this, all you have to do is look up the APP server subnet ID in the VPC console and provide it as input to the template.

At the end of this step you will have the following resources of our architecture launched:

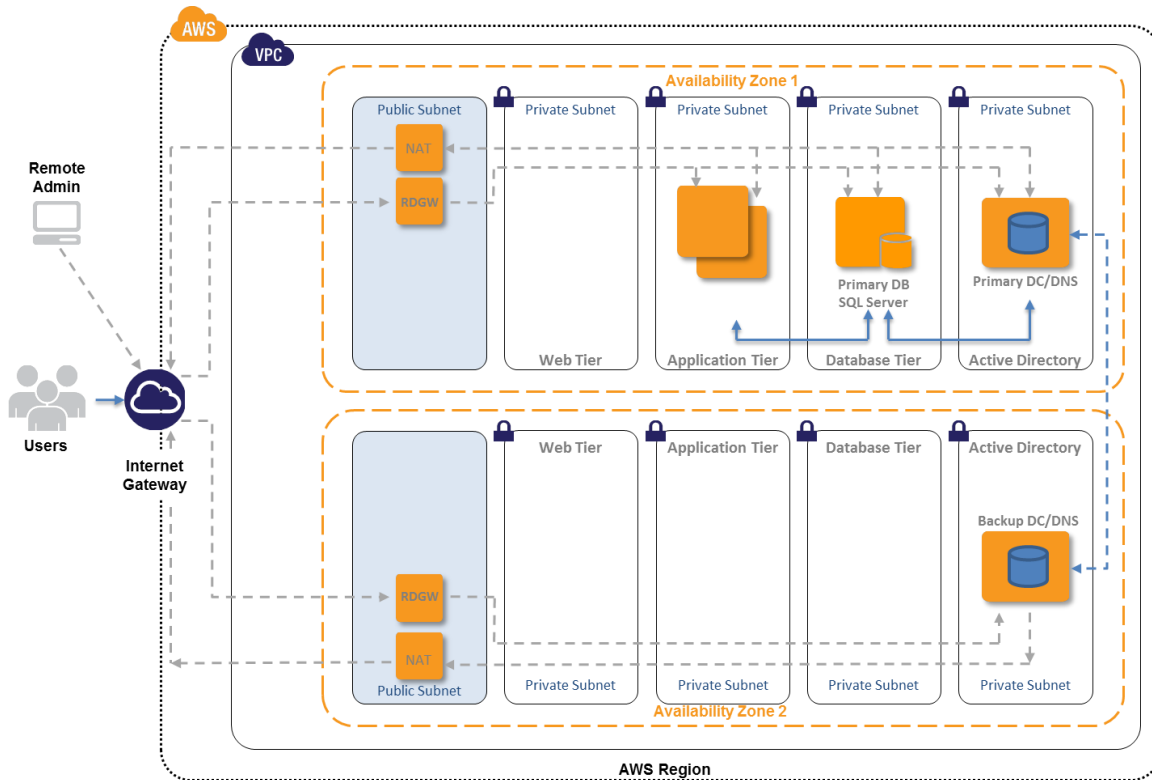


Figure 16: Architecture implemented at the completion of step 4

## Step 5. Launch the Web (WFE) Stack Using the Sample Template (Template-4)

Now that you have built your stack from the bottom up, launched the resources that make up your network and security infrastructure, your Active Directory and database plus your SharePoint APP server, you will complete the installation of the SharePoint Server Farm by launching 2 Web Front End (WFE) servers (1 per Web tier in each Availability Zone).

To proceed, click the "Launch Stack" button.

Launch in your preferred region:



[US East \(Northern Virginia\) Region](#)  
[US West \(Northern California\) Region](#)  
[US West \(Oregon\) Region](#)

[EU \(Ireland\) Region](#)  
[Asia Pacific \(Singapore\) Region](#)  
[Asia Pacific \(Tokyo\) Region](#)  
[South America \(Sao Paulo\) Region](#)

This will take you to the AWS Management Console's AWS CloudFormation stack wizard and load the sample Template-4.

The AWS CloudFormation sample Template-4 will launch the WFE server stack into the previously deployed networking, Active Directory, database and APP server infrastructure and join the two WFE Servers to the domain. Just as you looked up IDs when you launched the SQL and APP server stacks this will require you to look up the following IDs in the VPC console and input them into the template.

- VPC ID
- WFE Server Security Group ID
- Domain Member Security Group ID
- WFE1 Server Subnet ID
- WFE2 Server Subnet ID
- DMZ1 Subnet ID (Required to launch the Elastic Load Balancer)
- DMZ2 Subnet ID (Required to launch the Elastic Load Balancer)
- Load Balancer Security Group ID

This deploys 2 Windows Server 2008 R2 EC2 instances and installs SharePoint Server 2010 for Internet Sites - Enterprise on your WFE servers using your own license key (BYOL) or, for a trail or test installation, a Trail PID Key. (Please refer to Microsoft's [License Mobility through Software Assurance](#) program for further details on how you can provide your mobilized SharePoint license into the deployment process). It also deploys and configures an Elastic Load Balancer (ELB) which will balance traffic between your WFE servers.

At the end of this step you will have the following resources of our architecture launched:

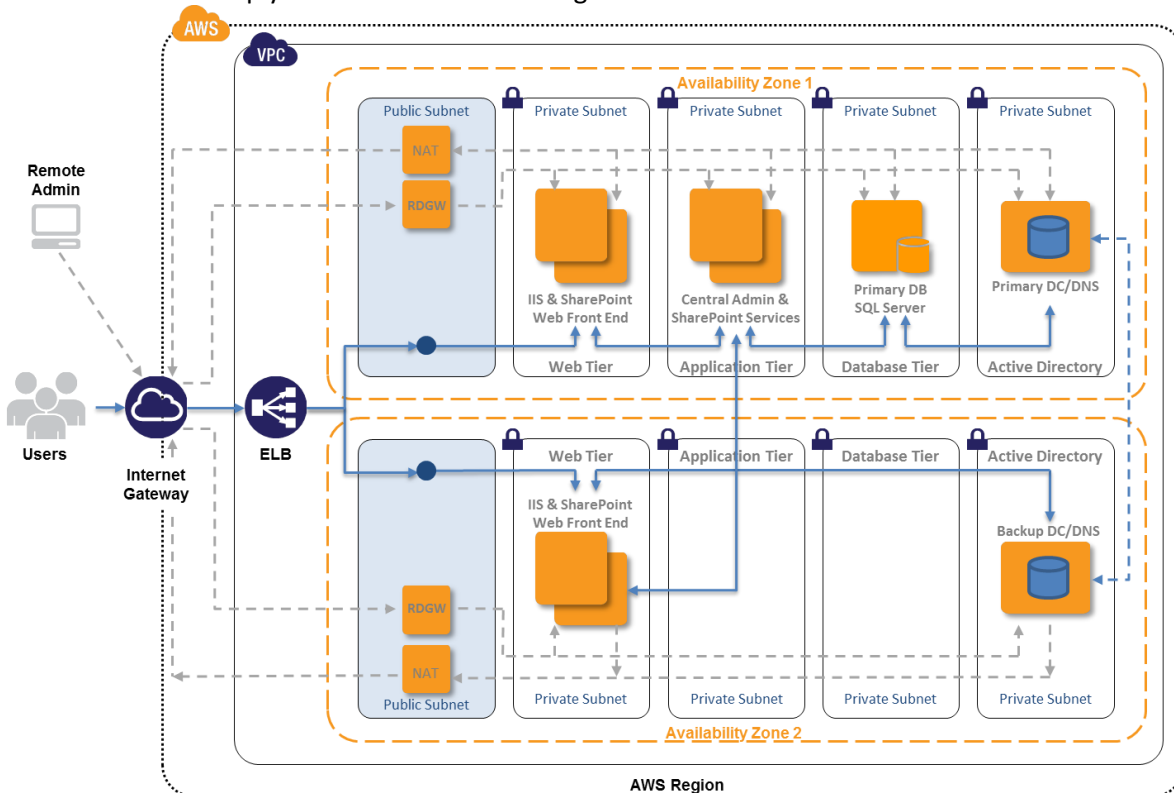


Figure 17: Architecture implemented at the completion of step 5

## Step 6. Configure the SharePoint Server Farm

---

After you have launched the complete stack, you need to configure your SharePoint Farm. It is outside the scope of this article to discuss the many configuration options that SharePoint supports. However, for the purpose of creating a simple proof of concept (POC) or demonstration setup, you will need to perform the following configuration steps:

On the APP Server:

- Run the SharePoint Products and Technologies Configuration Wizard.
- Create a new server farm.

On the WFE1 Server:

- Run the SharePoint Products and Technologies Configuration Wizard.
- Connect to an existing farm.
- Launch the Farm Configuration Wizard.
- Create a Site Collection.

On the WFE2 Server:

- Run the SharePoint Products and Technologies Configuration Wizard.
- Connect to an existing farm.

You will find in the in Step 6 of the [advanced guide](#) more detailed, step-by-step instruction that will guide you through the minimum setup steps required to configure a functional SharePoint Team Site. After that you are ready to connect to your SharePoint site using the IP address of the Elastic Load Balancer (ELB).

## How Much Is It Going to Cost to Run This Sample Site on AWS?

AWS offers significant savings over hosting the same application on-premises. It provides you with the flexibility to choose between different instance type configurations (such as Small, Medium, Large...), plus you can choose numerous combinations of On-Demand and Reserved Instances that match your usage projections. It also offers a wide range of Reserved Instance types (Light, Medium and Heavy Utilization) that allow you to save more money as you become more certain of individual instance utilization.

You can to use the [AWS Simple Monthly Calculator](#) to input your capacity requirements and estimate your costs. We have made a few assumptions and have provided a [sample calculation](#) for the deployment configuration discussed earlier, which assumes 3-Year Heavy Utilization Reserved Instances.

Click to view and/or customize the configuration: [Saved Sample Calculation](#)

Under those assumptions, the total cost of running this sample site (see the Resources section for the deployment configuration and setup) on AWS will be approximately **\$3,250/month**, amortized over 3-year period (includes Windows and SQL Server License). If the architecture can support approximately 500 concurrent users, the cost will be approximately **\$6.5 per User per Month**. We recommend that you adjust these assumptions based on your own

business requirements.

## How Can I Modify the Sample AWS CloudFormation Templates?

With Amazon EC2, you have complete control of your compute environment. For further details and additional information on how to modify the four AWS CloudFormation templates provided with this article, please read the [advanced guide](#). The advanced guide describes the contents of the AWS CloudFormation templates and PowerShell scripts so you can configure the various infrastructure details, such as compute instance types, provisioned storage, security, and networking controls.

## Resources

1. [Whitepaper: Microsoft SharePoint Server on AWS Reference Architecture](#) (PDF)
2. [Advanced Implementation Guide: Microsoft SharePoint 2010 on AWS](#) (PDF)
3. [Sample Template-1 \(Template 1 Infrastructure with AD.json\)](#). This will deploy following AWS resources:
  1. A VPC spanning 2 Availability Zones (AZ) with 10 subnets (8 private subnets and 2 public subnets) and 1 Internet Gateway
  2. 2 m1.xlarge Amazon EC2 Instances running Microsoft Windows Servers 2008 R2 that serve as Active Directory and DNS Servers
  3. 2 m1.small Amazon EC2 Instances that serve as NAT instances
  4. 2 m1.large Amazon EC2 Instances running Microsoft Windows Server 2008 R2 (1 per DMZ in each AZ) serving as Remote Desktop Gateway (RDGW) servers
  5. Standard EBS volumes
4. [Sample Template-2 \(Template 2 SQL Server.json\)](#). This will deploy following AWS resources:
  1. 1 m2.4xlarge (EBS optimized) Amazon EC2 Instance running Microsoft SQL Server Standard 2008 R2 serving as the database tier
    1. 6 x 50 GB EBS Volumes with 500 PIOPS/disk
5. [Sample Template-3 \(Template 3 APP Server.json\)](#). This will deploy following AWS resources:
  1. 1 m1.xlarge Amazon EC2 Instances running Microsoft Windows Server 2008 R2 serving as the App server that hosts the SharePoint Central Administration and SharePoint Service
    1. 4 x 100 GB Standard EBS volumes
6. [Sample Template-4 \(Template 4 WFE Server.json\)](#). This will deploy following AWS resources:
  1. 2 m1.xlarge Amazon EC2 Instances running Microsoft Windows Server 2008 R2 (1 in each AZ) serving as the Web Front End (WFE) servers
    1. each with 4 x 100 GB Standard EBS volumes
  2. 1 Elastic Load Balancer