Microsoft SharePoint Server on AWS: Reference Architecture

February 2012

(Please consult http://aws.amazon.com/whitepapers for the latest version of this paper.)

Abstract

Amazon Web Services (AWS) provides a complete set of services and tools for deploying Windows® workloads, including Microsoft® SharePoint® Server, on its highly reliable and secure cloud infrastructure platform. This white paper discusses general concepts regarding how to use these services and provides detailed technical guidance on how to configure, deploy, and run a SharePoint Server farm on AWS. It illustrates reference architecture for common SharePoint Server deployment scenarios and discusses their network, security, and deployment configurations so you can run SharePoint Server workloads in the cloud with confidence.

This white paper is targeted to IT infrastructure decision-makers and administrators. After reading it, you should have a good idea of how to set up and deploy the components of a typical SharePoint Server farm on AWS. You learn which artifacts to use and how to configure the various infrastructure details, such as compute instances, storage, security, and networking.

Introduction

Enterprises need to grow and manage their global computing infrastructures rapidly and efficiently while simultaneously optimizing and managing capital costs and expenses. AWS's computing and storage services meet this need by providing a global computing infrastructure. The AWS infrastructure enables companies to rapidly spin up compute capacity or quickly and flexibly extend their existing on-premise infrastructure into the cloud. AWS provides a rich set of services and robust, enterprise-grade mechanisms for security, networking, computation, and storage.

SharePoint Server is a widely deployed application platform, common in many organizations as the main portal for team–corporate collaboration, content management, workflow, and access to corporate applications. One key benefit of SharePoint Server is that it enables organizations to rapidly respond to changing business needs. AWS is a perfect complement to SharePoint Server, because it enables organizations to rapidly provision the necessary computing infrastructure to power SharePoint Server solutions.

AWS and Microsoft have partnered to enable customers to deploy enterprise-class workloads involving Windows Server® and Microsoft SQL Server® on a pay-as-you-go, on-demand elastic infrastructure, thereby eliminating the capital cost for server hardware and greatly reducing the provisioning time required to create or extend a SharePoint Server farm. This partnership has resulted in the ability to license and run SharePoint Server on AWS under provisions in Microsoft's License Mobility through Software Assurance program.

As a relevant data point and case study, the Amazon Corporate IT team hosts Amazon's own corporate intranet running SharePoint Server on AWS. They have published a white paper detailing its evaluation, security requirements, architecture, benefits, and lessons learned from the deployment. Note that at the time the Amazon Corporate IT team deployed their SharePoint Server environment and wrote the white paper, a number of the AWS services discussed herein were either not in place or limited in their availability. This current paper provides an up-to-date and more high-level description of how to support SharePoint Server on AWS.

SharePoint Server Reference Architecture and Scenarios

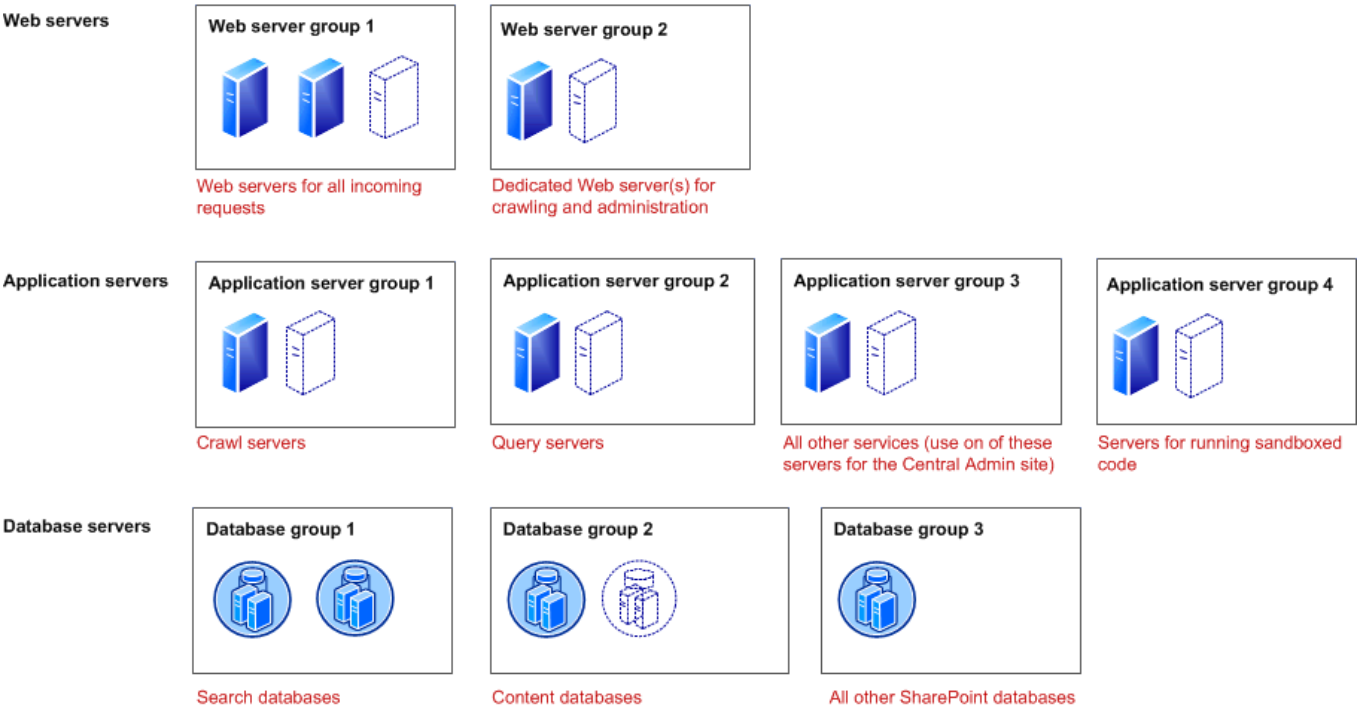
To understand how SharePoint Server and associated components can be hosted on AWS, let's first review the architecture and components of a typical SharePoint Server farm and explore the common scenarios and topologies.

SharePoint Server Farm Reference Architecture

Microsoft provides considerable guidance for architecting SharePoint Server farm topologies for many scenarios and scales. This section reviews the typical SharePoint Server farm architecture as recommended by Microsoft and identifies a couple of common deployment scenarios and associated topologies that you will map onto AWS later in this paper.

SharePoint Server has evolved over several versions to provide a rich set of capabilities and services. SharePoint Server architecture has also evolved to support a service-based architecture, enabling specific services to be scaled out to individual servers and server groups. In addition, SharePoint Server reference architecture defines distinct roles and server groups that you can create and scale out independently. This model fits nicely within AWS's scale-out approach.

The SharePoint Server reference architecture tiers and services are illustrated in Figure 1.



Source: http://technet.microsoft.com/en-us/library/ff758647.aspx

Figure 1: The SharePoint Server reference architecture

Additional infrastructure components are required or recommended to support SharePoint Server farms:

- **Active Directory® Domain Services (AD DS).** SharePoint Server requires AD DS to serve as the authoritative identity store and authentication mechanism. AD DS (with one or more domain controllers) must reside within the same network as the SharePoint Server farm and be accessible to SharePoint Server farm instances.

- **Threat management and intrusion protection.** This component may be an additional element for SharePoint Server scenarios that include external or public-facing sites. In a Windows-based infrastructure, this component would typically be provided by products such as Microsoft Forefront® Threat Management Gateway 2010.

Common SharePoint Server Deployment Scenarios

SharePoint Server can support a variety of content and collaboration goals. This paper discusses two of the most common scenarios: intranet hosting of a corporate SharePoint Server farm and hosting of an Internet site based on SharePoint Server.

Intranet SharePoint Server Farm

In this scenario, a company wants to run SharePoint Server within its enterprise to support internal users. The company may deploy its entire SharePoint Server farm in the cloud and scale all the components to get additional capacity or extend its on-premise deployment to the cloud to increase capacity, improve performance, or scale the resource-intensive components in the cloud, when needed. Specific resource-intensive services such as Microsoft Office Excel® or Word may be hosted individually to support specialized workloads. Figure 2 illustrates this scenario.

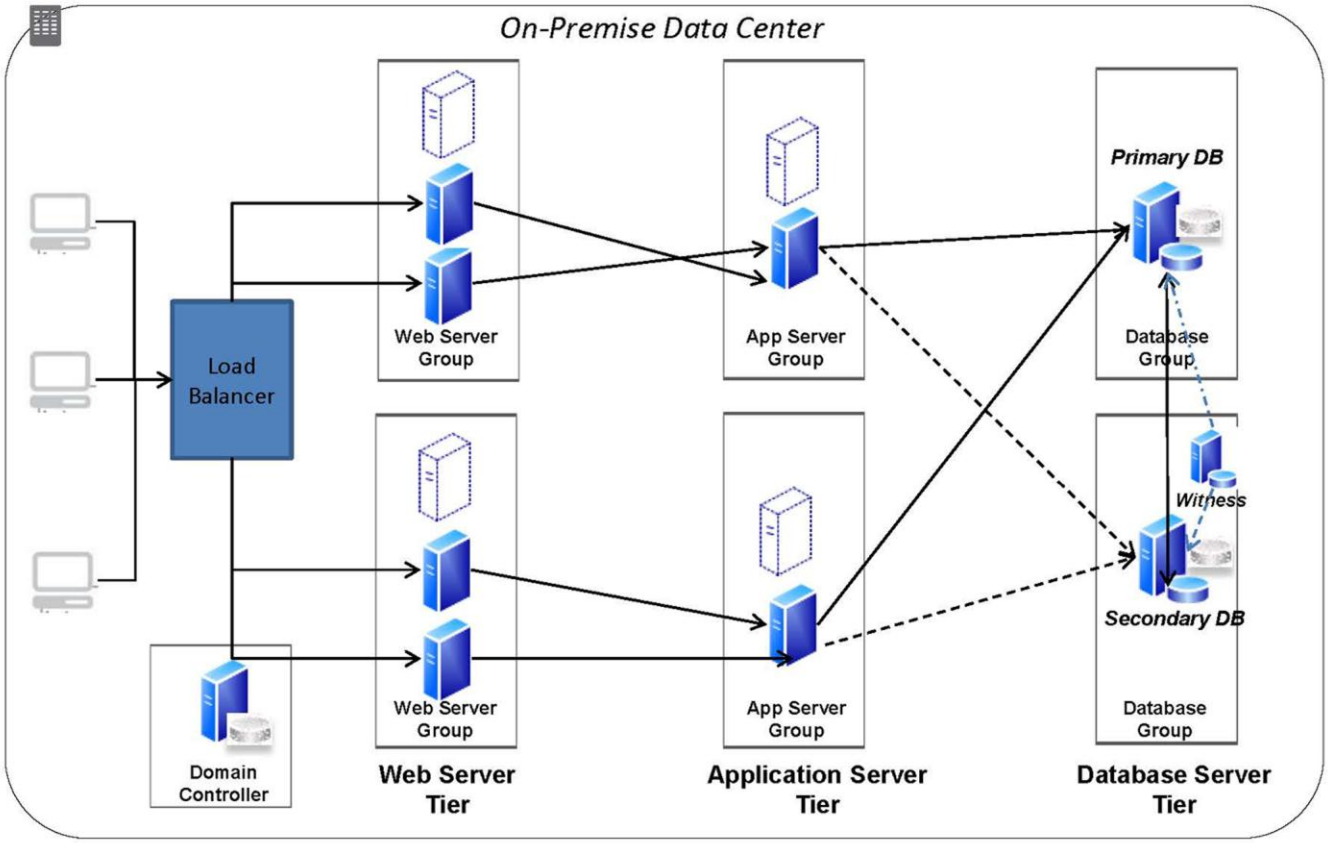


Figure 2: Typical intranet SharePoint Server farm topology

Internet Website or Service Based on SharePoint Server

In this scenario, SharePoint Server is used as the basis for hosting a website, public web application, or Software as a Service (SaaS) site. This scenario is different from the intranet scenario in that public-facing servers have been added. These servers require enhanced security and threat management as well as AD DS domain controllers to support user authentication and authorization. Figure 3 depicts this scenario.

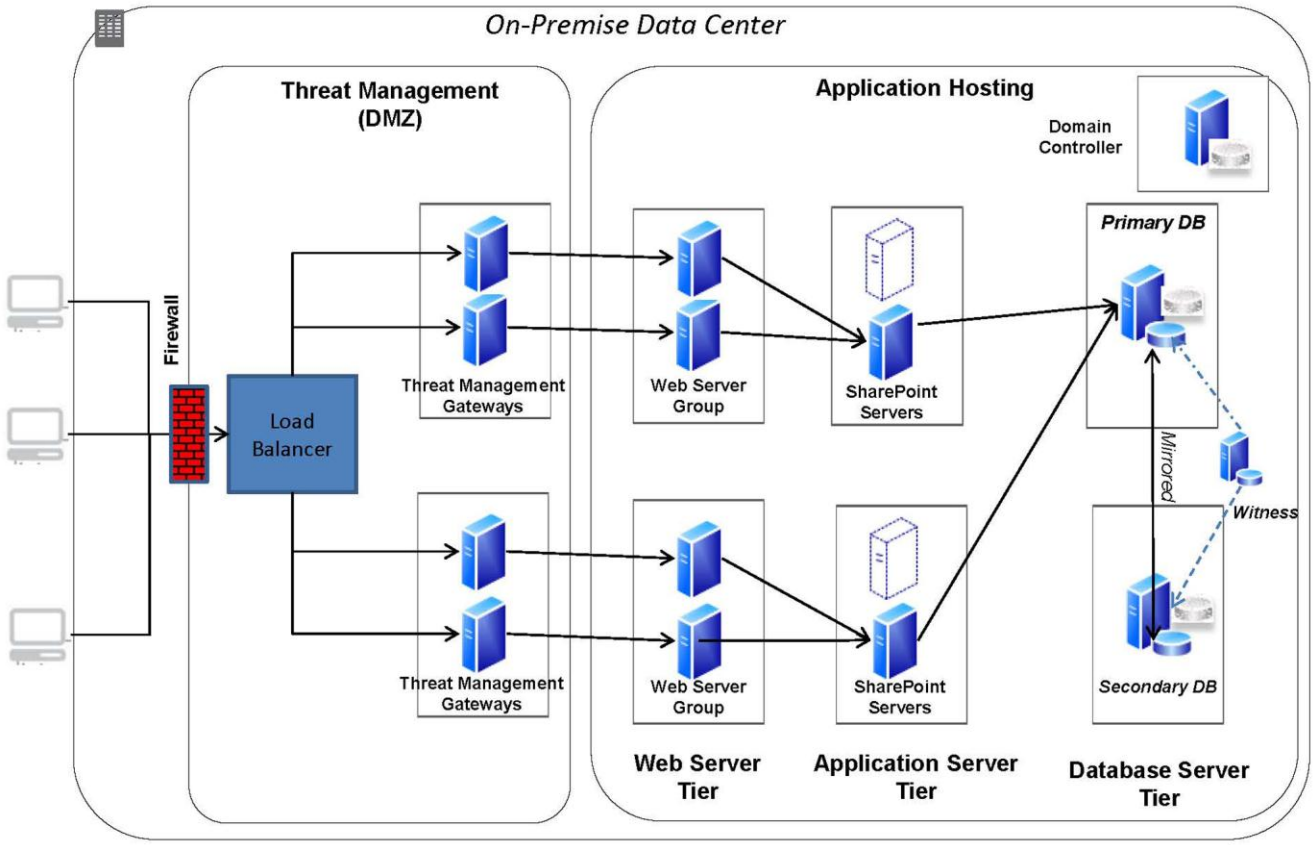


Figure 3: Typical SharePoint Server farm topology for an Internet-facing public website

Key elements that distinguish this scenario from the previous intranet scenario are:

- A demilitarized zone (DMZ) to provide firewall and threat management at the front-line access points
- Active Directory domain controllers resident within the farm (not associated with the user environment)

Implementing SharePoint Server Architecture Scenarios in AWS

The remainder of this white paper provides step-by-step mapping for each SharePoint Server farm scenario discussed earlier to an equivalent setup in AWS, including similar resources, network and security setup, and configuration. To implement the SharePoint Server scenarios in AWS, the following elements are discussed:

- **Network setup and configuration.** This section covers the setup of the network for the SharePoint Server farm within AWS, including subnets to support the logical server groups for different tiers and roles within the SharePoint Server reference architecture.

- **Server setup and configuration.** This section covers the services and artifacts involved in the setup of the various servers for each tier and role in the SharePoint Server farm. It also includes setting up and configuring SQL Server and supporting high availability.

- **Security.** This section discusses security mechanisms in AWS, including how to configure instance and network security to enable authorized access to the overall SharePoint Server farm as well as between tiers and instances within the farm. It also covers areas such as data privacy (encryption) and threat management (in the case of the public-facing scenario).

- **Deployment and management.** This section provides details on packaging, deployment, monitoring, and management of the SharePoint Server farm components.

Network Setup

Let's start with the network setup to provide the environment in which you instantiate and configure your servers and database.

The Microsoft reference architecture is organized around a multi-tiered (web, application, and database) approach, allowing you to independently scale and configure each tier. Your first task is to define a network environment that supports this type of tiered structure and enables you to deploy the various server roles in each tier with suitable security configuration.

Amazon Virtual Private Cloud

A key component of AWS networking is the Amazon Virtual Private Cloud (Amazon VPC). Amazon VPC provides the ability to reserve an isolated portion of the AWS cloud in which to deploy and manage a SharePoint Server farm. Amazon VPC supports the creation of public and private subnets within the virtual network, allowing you to host the different tiers and roles within the SharePoint Server architecture.

Amazon VPC also supports the ability to establish a hardware virtual private network (VPN) connection between a VPC and an external location, such as a corporate data center. Customers use a hardware or software VPN appliance (the customer gateway) and connect that gateway to the VPC (the virtual private gateway) to provide seamless integration between on-premise compute infrastructure and resources within the VPC. Leveraging this VPN–VPC connectivity extends the corporate network data center to the cloud. Corporate users can interact with cloud instances and applications in a relatively transparent way, effectively supporting the notion of an "extended enterprise" in the cloud.

To map your SharePoint Server reference architectures and scenarios to AWS, you must first structure your VPC and subnets to mirror the same organizational tiers, server groups, and access requirements defined there. VPC subnets that need to be accessible from the Internet through the VPC Internet gateway need to be public; otherwise, you can

designate them as private, and they will not be accessible from outside the VPC. In the case of a VPN-connected VPC, connections through the VPN occur through the virtual private gateway; therefore, instances can be in private subnets but still reachable (as long as the security configuration allows it). **Thus, VPN-only scenarios do not require public subnets (e.g., for web server front ends).** However, the public-facing SharePoint Server scenario does need to be accessible from outside of AWS, so each front-end instance must be in a public subnet to be reached via the Internet gateway.

Fault tolerance and scalability for our SharePoint Server farm scenarios is critical to ensure they can provide sufficient performance through changes in load, and be resilient to any unforeseen issues within the farm infrastructure. The Elastic Load Balancing (ELB) web service can be used to be used to distribute internet-based requests to internal web servers, and so this is a suitable choice for our internet website scenario. However, since ELB at this point only handles traffic coming from outside the VPC, we can't use that for our intranet scenario (in which user requests come in via the private VPN connection). For the intranet scenario, we need to utilize a 3rd party software load balancer (such as the Riverbed Stingray Traffic Manager or HAProxy) to achieve similar functionality.

You also want to distribute multiple instances to each Availability Zone to provide redundancy and failover in the case of an Availability Zone failure. VPC subnets do not span Availability Zones, so you must set up a separate but similar subnet structure within each zone. Likewise, set up load balancing to distribute requests to servers in multiple Availability Zones. Therefore, you should set up load balancers in each Availability Zone used to provide high availability there, as well.

NOTE: The IP address ranges for the VPC and subnets are defined using a single Classless Inter-domain Routing (CIDR) IP address block, such as 10.0.0.0/16, providing an internal IP address space of 65,536 unique IP addresses. Subnets can then be created with their own unique CIDR block ranges within the overall VPC address range.

VPC Setup for the Intranet Scenario

Let's look at the specific steps for setting up a VPC instance for the intranet scenario.

The AWS Management Console provides a wizard-based approach to setting up Amazon VPC environments for a few typical Amazon VPC configurations. For your SharePoint Server intranet scenario, the goal is to set up the AWS environment to enable corporate users to use SharePoint Server via VPN access; but you do not need to allow access from the public Internet. The VPC Creation Wizard option **VPC with a Private Subnet Only and Hardware VPN Access** initiates the setup you are looking for.

NOTE: Servers within the farm may need to exit of AWS for things like software updates. Such actions can be accomplished either by adding a network address translation (NAT) instance in the VPC and configuring it to be public or by having the servers traverse the VPN tunnel to use the corporate data center Internet access. Amazon VPC includes a default route table that guides communications to and from instances, and the VPC Creation Wizard enables the route tables to allow instances to communicate with each other (using the internal VPC IP addresses) and externally out of the VPC (for all other IP addresses) through the NAT instance.

Based on the specifics of your SharePoint Server intranet scenario, you must add several components into the results of the basic "Scenario 4" setup that the VPC Creation Wizard provides:

- **One VPC created within a specific AWS region that has components spanning multiple Availability Zones.** Your SharePoint Server infrastructure will be deployed across multiple Availability Zones to provide high availability.

- **Private subnets in each Availability Zone to hold your load balancers.** A VPC can have multiple subnets in which each subnet resides in a separate Availability Zone. Each subnet must reside entirely within one Availability Zone.

- **Software Load Balancers in each Availability Zone.** This setup establishes primary and secondary load balancers within each of the Availability Zones, where the primary distributes traffic to any of the healthy instances in either of the Availability Zones. In the event of a failure of the primary load balancer (or the Availability Zone overall), the secondary load balancer takes over and continues to distribute traffic to remaining healthy instances.

- **Private subnets in each Availability Zone to hold web, application, and database servers as well as AD DS domain controllers.** These subnets are not directly accessed by users (everything goes through the load balancers) and hence do not need to be accessible outside of the VPC.

- **One virtual private gateway and one customer gateway.** These provide VPN connectivity between the corporate data center and the VPC.

Putting together everything discussed thus far, Figure 4 shows the network configuration defined for the intranet scenario.

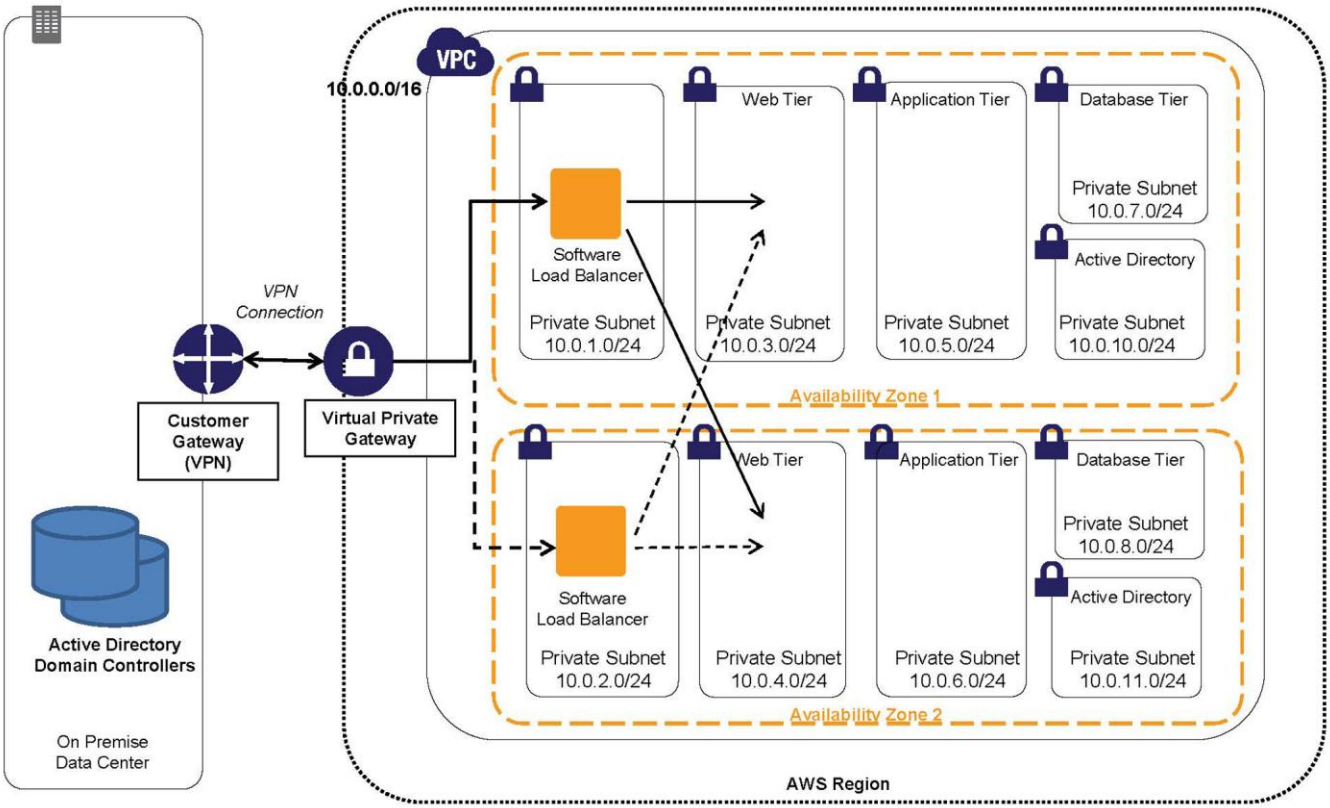


Figure 4: Network configuration for the intranet scenario

VPC Setup for the Public Website Scenario

For the public website scenario, there are different requirements and setup configurations.

The public website scenario most resembles the VPC Creation Wizard scenario "VPC with Public and Private Subnets." The differences between the public website scenario and the intranet scenario are:

- In the public website scenario, you do not have a corporate data center, so there is no need to set up a VPN connection.
- With a public website, there is no need for a virtual private gateway (because you are not connecting to a VPN).
- In this scenario, AWS Elastic Load Balancers are employed
- In a public-facing website, the load balancers need to be in public subnets so that users can access them over the Internet.
- You still want to put the web, application, and database tiers in private subnets; users only need to get at the load balancers.
- The public website scenario requires additional components at the front end for firewall and threat management (more on this topic later).
- The public website scenario adds NAT instances in each Availability Zone to facilitate servers in private subnets communicating out to the Internet (to get operating system software updates, for example).

Given these differences, Figure 5 shows the network setup for the public website scenario.

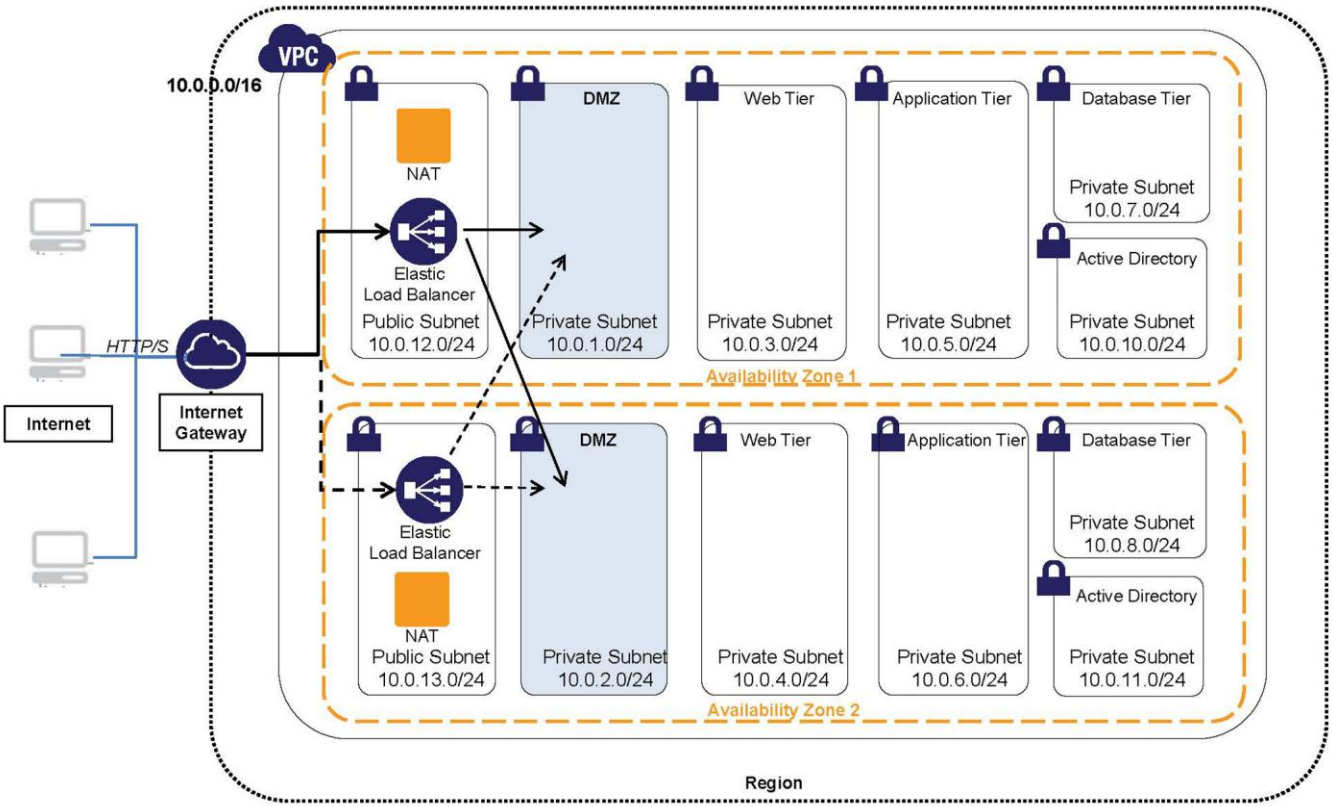


Figure 5: Network configuration for the Internet-facing public website scenario

AD DS Setup and DNS Configuration

SharePoint Server requires AD DS for user authentication. However, you also want to leverage AD DS to provide Domain Name System (DNS) functionality within the VPC among the various server instances.

For your SharePoint Server farm to operate, you need connectivity to one or more domain controllers to facilitate user authentication and DNS resolution across servers within the farm. In the intranet scenario, you want the SharePoint Server instances to authenticate to users' corporate credentials (effectively an extension of their corporate network). There are two different ways to support this behavior:

- SharePoint Server instances could traverse the VPN–VPC connection back to the corporate data center and authenticate to on-premise domain controllers.
- Domain controllers could be hosted in AWS and replicated from on-premise domain controllers via the VPN–VPC connection. This action allows the servers to authenticate to local (within AWS) domain controllers but still authenticate to corporate user identities and credentials.

Amazon recommends the second option for better performance and reliability. The domain controllers can be replicated across Availability Zones (as with your other resources) to provide high availability. Microsoft provides guidance on Active Directory Replication Over Firewalls.

NOTE: It is also possible to support this scenario for corporate environments that do not use AD AD but rather another Lightweight Directory Access Protocol (LDAP)–based directory service. You can use Active Directory Federation Services (AD FS) with SharePoint Server and other (non-AD DS) authentication providers to facilitate federated authentication. AWS provides a detailed white paper on how to set up and configure AD FS in AWS to support federated authentication. Figure 6 depicts the additions to the hosting infrastructure and AD DS replication details.

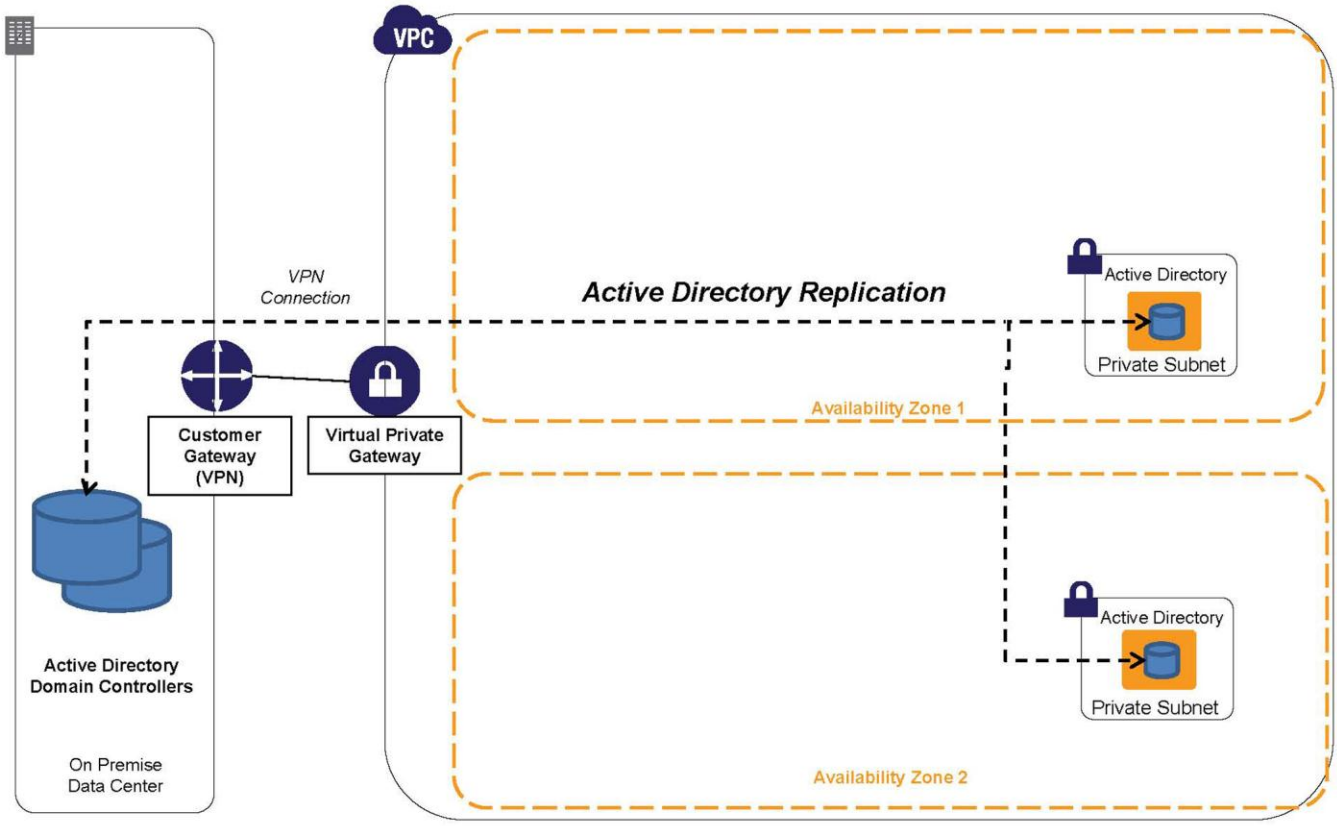


Figure 6: Additions to the hosting infrastructure and AD DS replication details for the intranet scenario

In your public-facing scenario, the SharePoint Server farm is not connected to a corporate infrastructure via VPN. Instead, it requires AD DS to be instantiated within the AWS environment to facilitate user registration and authentication for the SharePoint Server instances running there. As in the intranet scenario, Amazon suggests hosting domain controllers in multiple Availability Zones to provide redundancy and high availability, as illustrated in Figure 7.

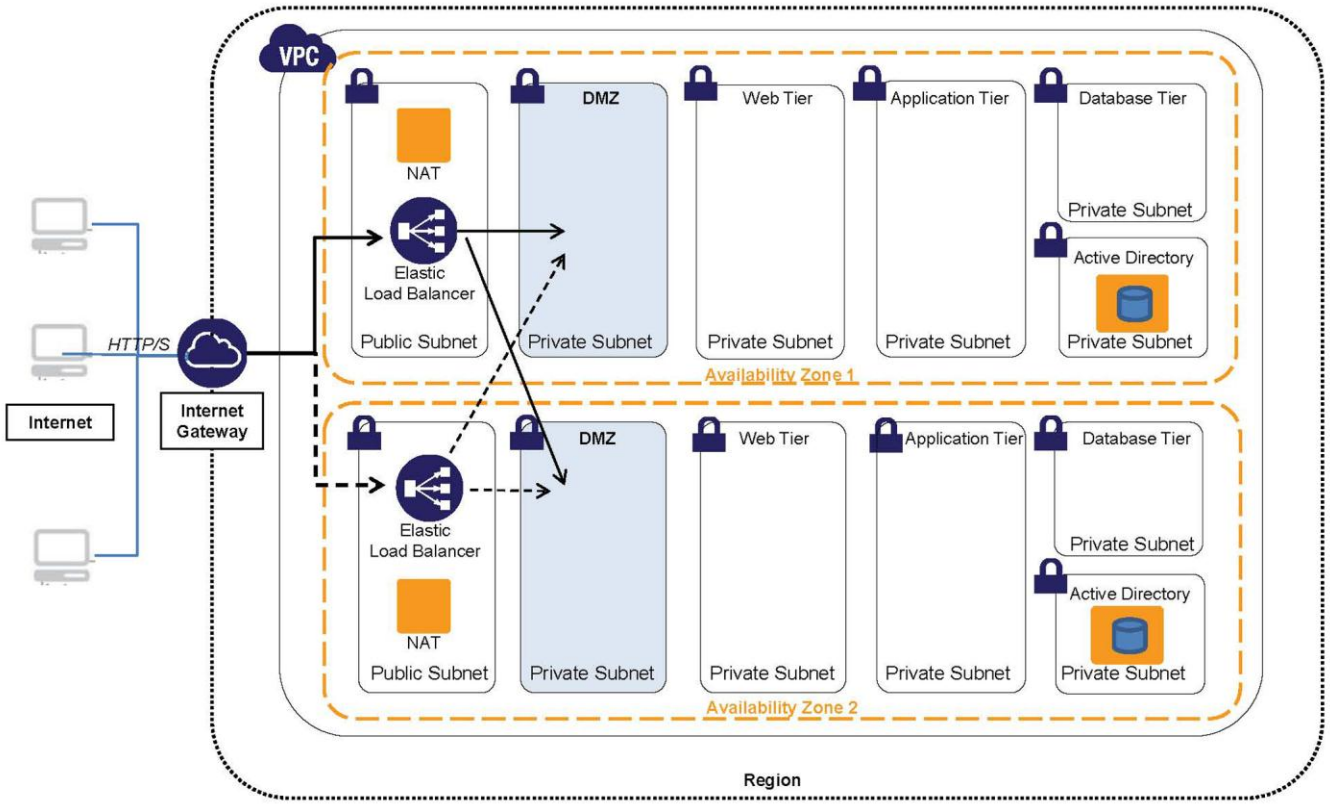


Figure 7: Hosting domain controllers in multiple Availability Zones to provide redundancy and high availability

AD DS is typically run in on-premise, static environments, and there are certain typical configuration details and assumptions that are different when AD DS runs in AWS. For AD DS domain controllers to be used for DNS in AWS and across Availability Zones, each needs to be in a security group that opens User Datagram Protocol (UDP) ports 0–65,535. (Security groups are discussed in detail in a later section.)

Server Setup and Configuration

Now that your network is set up in the structure you need, let's tackle the task of setting up and instantiating the various server instances within the VPC to support your SharePoint Server reference architectures.

At the heart of AWS is the Amazon Elastic Compute Cloud (Amazon EC2) web service, a cloud computing infrastructure that supports a variety of operating systems and machine configurations (e.g., CPU, RAM). AWS provides preconfigured virtual machine (VM) images (Amazon Machine Images, or AMIs) with guest operating systems (Linux®, Windows, etc.) and may have additional software (e.g., SQL Server) used as the basis for virtualized instances running in AWS. You can use these AMIs as starting points to instantiate and install or configure additional software, data, and more to create application- or workload-specific AMIs.

To implement the various tiers and roles in the SharePoint Server reference architecture, start out with AMIs that are based on Windows Server 2008 R2, and look at the software running each one to determine which AMIs are applicable to web, application, or database tier servers. At this time, several AMIs support some version of Windows Server. Some AMIs include components like Microsoft Internet Information Services (IIS) for the web tier roles; others include SQL Server Standard (for the database tier).

SharePoint Server is not preinstalled in any of the Windows-based AMIs because of licensing model restrictions. The only supported approach to licensing SharePoint Server on AWS is through Microsoft's License Mobility through Software Assurance program. Customers covered by active Microsoft Software Assurance contracts may move current on-premise Windows Server application workloads (such as SharePoint Server) to AWS without additional Microsoft software license fees.

AWS provides a comprehensive collection of information, tools, and resources for running Windows-based applications and workloads on AWS. Also, there is detailed information about how Windows is supported and used on Amazon EC2. Finally, you can find details on the specific AMIs that include Windows, SQL Server, etc., within the Amazon EC2 AMI catalog.

Mapping SharePoint Server Roles and Servers to Amazon EC2 AMIs and Instance Types

A key aspect of implementing your AWS solution is choosing the appropriate AMI and instance type for each role within the farm. Each role in the SharePoint Server reference architecture has distinct requirements for software and infrastructure resources, such as CPU, RAM, and disk storage. Microsoft and AWS have partnered to publish a number of Windows-based AMIs that include additional software components for supporting typical roles (e.g. IIS for web server, SQL Server for database server, Windows core for domain controller) that run on a variety of Amazon EC2 instance types.

In terms of machine capacity and sizing, Microsoft provides detailed guidance for various components within a SharePoint Server farm, so that topic is not be covered in this paper. However, the basic details of typical system requirement minimums for various components within a SharePoint Server farm are summarized in the tables that follow.

Table 1 presents the minimum system requirements Microsoft recommends for the different tiers and roles within a SharePoint Server farm.

Table 1: Minimum system requirements for SharePoint Server roles and tiers

Tier/role	Scenario	Processor	RAM	Hard disk
Web/Application Tier	All	64-bit, 4 core	8 GB	80 GB
Database server	Small deployment	64-bit, 4 core	8 GB	80 GB
Database server	Medium deployment	64-bit, 8 core	16 GB	80 GB
Domain controller	All	64-bit, 4 core	8 GB	80 GB

Table 2 shows how to map these requirements to Amazon EC2 AMIs and Windows instance types.

Table 2: Mapping minimum system requirements to AMIs and Windows instance types

Tier	Applicable Amazon EC2 instance type and range	AMI to use
Web front end	Extra Large (m1.xl)	Windows Server 2008 R2 + IIS
Application server	Extra Large: High Memory Quad Extra Large (m2.xl–m2.4xl)	Windows Server 2008 R2
Database server	High Memory Quadruple Extra Large (m2.4xl)	Optimized SQL Server 2008 R2 AMIs from Microsoft
Domain controller	Extra Large (m1.xl)	Windows Server (in the role of a domain controller)

The AMIs listed in Table 2 include the default configuration for Amazon EBS volumes (formatted as Windows file systems) for boot drive and associated data storage applicable to the role. The SQL Server 2008 R2 AMIs indicated have been configured with multiple EBS volumes to support distinct SQL Server storage components (data, logs, temp files), optimizing for storage requirements and I/O patterns of each component. Amazon EC2 also supports the ability to customize an instance, allowing you to attach additional Amazon EBS volumes or resize an existing Amazon EBS volume by taking a snapshot, and then creating a new, larger volume from the snapshot. You can then use this customized instance as the basis for a new, customized AMI.

SharePoint Server Configuration

As mentioned earlier, SharePoint Server is not pre-installed in any publically available AMI, so you must obtain sufficient licensing for deploying SharePoint Server in AWS (through Microsoft License Mobility) and then install SharePoint Server into your instances. Typically, you will create your own private SharePoint Server AMI, by creating a Windows Server-based instance, installing and configuring SharePoint Server, and then turning that instance into an AMI as described here. This private AMI will be the basis of the various SharePoint Server instances in your farm.

SQL Server Configuration

The versions of SQL Server that are included and licensed for use with the Windows Server AMIs are SQL Server Express and SQL Server Standard. SQL Server Enterprise can be installed in Windows AMIs and used in AWS as well but must be licensed for use in the same way as SharePoint Server, through provisions in the Microsoft License Mobility through Software Assurance program.

As in on-premise deployments, the data tier for SharePoint Server in AWS needs to be architected and configured to support sufficient performance, high availability, and reliability to provide a good user experience and quickly respond to a database failure with minimal transaction loss. For SQL Server instances, Amazon recommends the High Memory Quadruple Extra Large Amazon EC2 instance type. This type provides higher-performance network I/O (high). This higher performance, combined with the other metrics such as CPU, yields a good performance profile for SQL Server running on AWS.

Recommended Amazon EBS Disk Configuration for SQL Server

Amazon EBS volumes can be configured in a variety of ways (redundant array of independent disks [RAID] striping, different volume sizes, etc.) to yield different performance characteristics. The optimized SQL Server Standard AMI mentioned earlier is published jointly between Microsoft and AWS and is configured with separate Amazon EBS volumes, each storing key SQL Server data components as recommended by Microsoft for optimal performance.

For high-I/O scenarios, it is possible to create and attach additional Amazon EBS volumes and to stripe using software RAID to increase the total number of I/O operations per second (IOPS). Each Amazon EBS volume is protected from physical drive failure through drive mirroring, so using a RAID level higher than RAID-0 is unnecessary.

For SharePoint Server instances, it is common to use Remote BLOB Storage (RBS) in conjunction with SQL Server for storage of file-based content. This file-based content will reside in SQL Server instances, and the existing Amazon EBS configuration should be sufficient for most uses. However, it may be desirable or necessary to extend the size or add more Amazon EBS disks (or other associated storage) for supporting large RBS stores. For further details regarding Amazon EBS setup, configurations, and tuning options, see the Amazon Elastic Compute Cloud User Guide.

High Availability for SQL Server

You can achieve high availability for SQL Server in AWS by implementing SQL Server mirroring across multiple Availability Zones. In this configuration, SQL Server instances are launched in two different Availability Zones (within a Region), with a smaller “witness” SQL Server instance to monitor and facilitate the failover, if needed. Figure 8 illustrates this configuration.

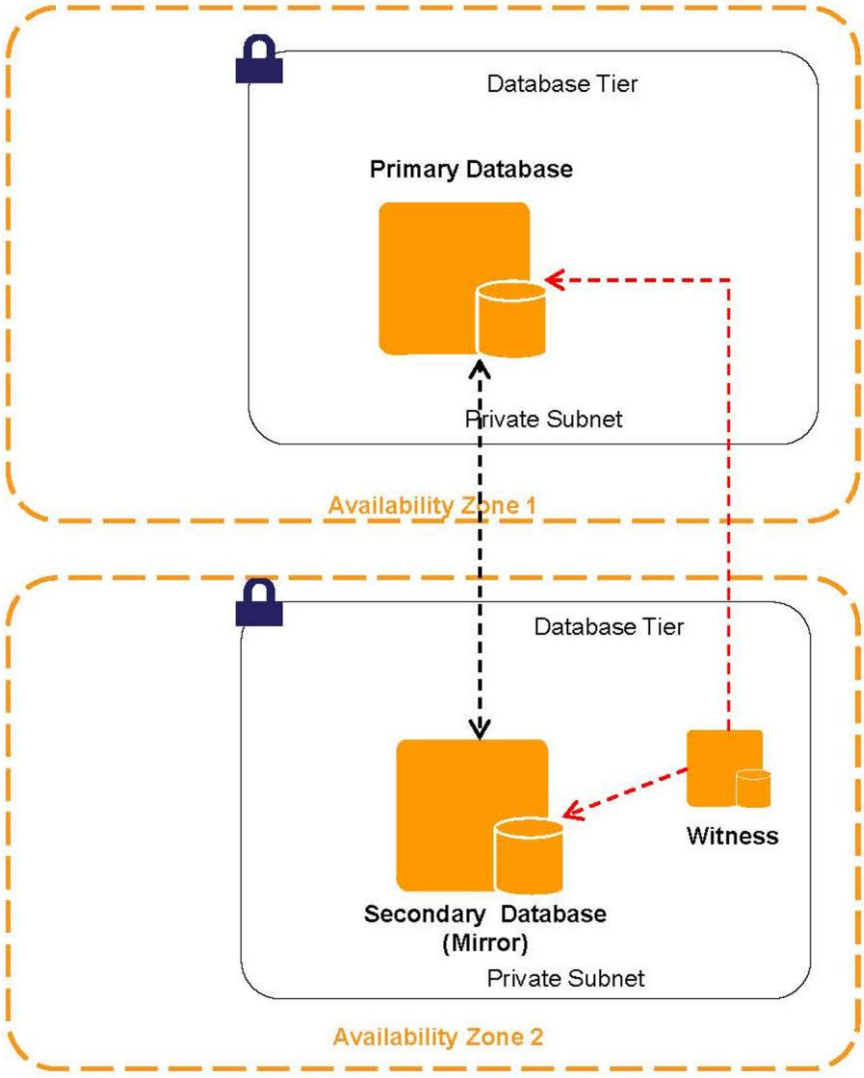


Figure 8: SQL Server mirroring across multiple Availability Zones

AWS recently published RDBMS in the Cloud: Microsoft SQL Server 2008 R2, a comprehensive resource that provides a detailed discussion of considerations, approaches, and options for optimizing the use of SQL Server in AWS. With the addition of your Amazon EC2 instances and SQL Server mirroring, your intranet scenario looks like Figure 9.

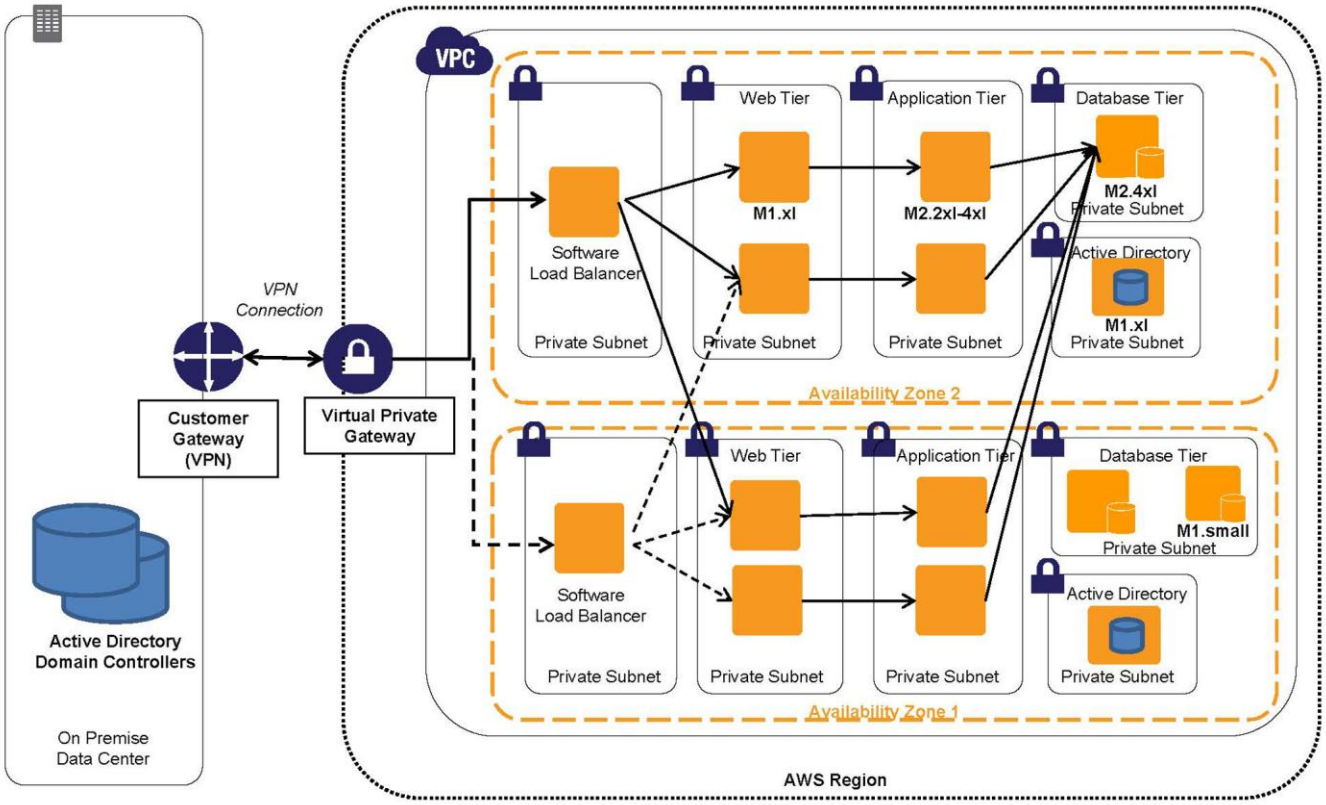


Figure 9: Intranet scenario with the addition of Amazon EC2 instances and SQL Server mirroring

With the addition of your Amazon EC2 instances and SQL Server mirroring, your public site scenario looks like Figure 10.

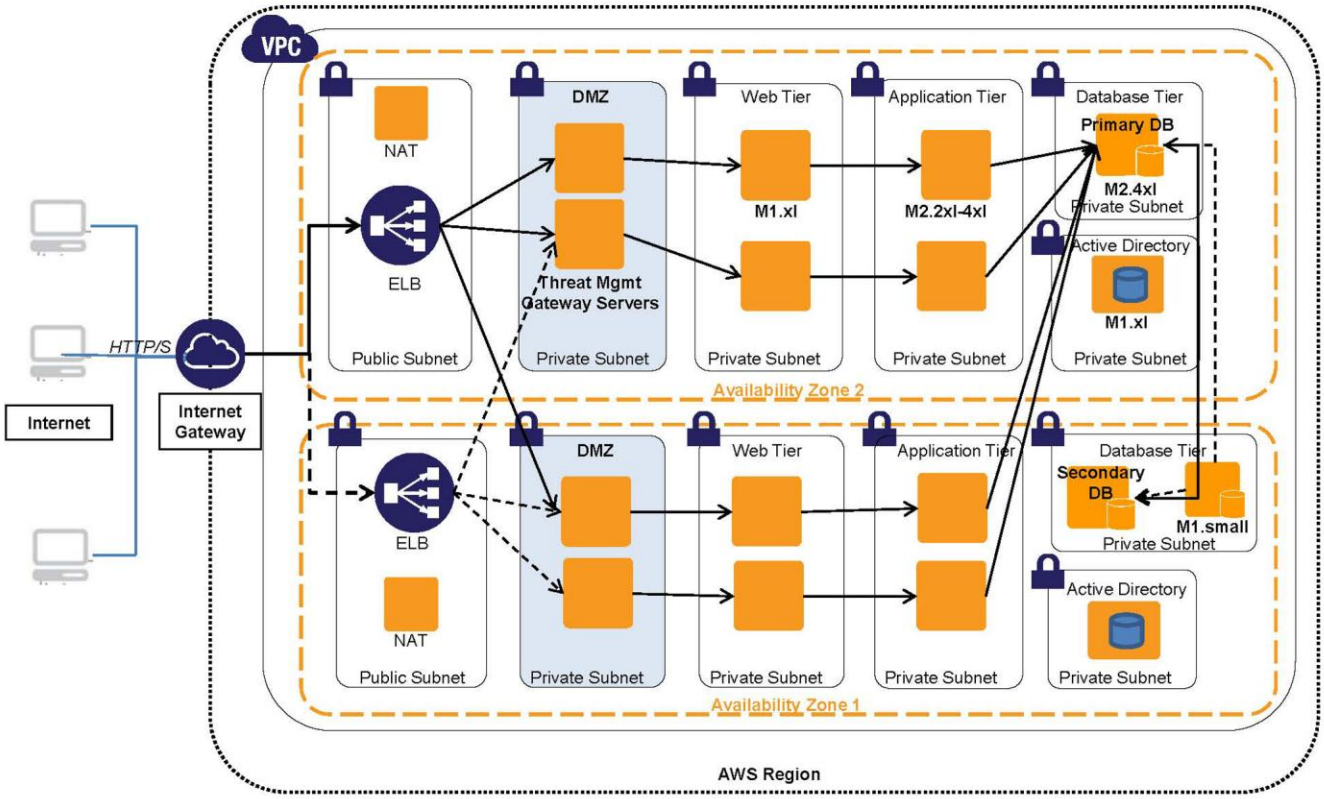


Figure 10: Public site scenario with the addition of Amazon EC2 instances and SQL Server mirroring

Security

Security setup is critical in the implementation of your SharePoint Server farm to enable proper network access (in and out of the VPC, specific subnets, and the instances running each subnet) to facilitate user authentication and appropriate authorization, data privacy, and threat management (in the case of public-facing sites). These and other key elements have to be set up correctly to provide the necessary security measures and enable users to access their SharePoint Server content and applications with the correct identity and authorization.

A cornerstone of your scenarios is the use of Amazon VPC for providing the overall isolation of the farm and segmenting parts of the farm (i.e., the server groups) to support the desired management and control. Within Amazon VPC and subnet isolation, there are security details that you must set up to enable proper access (and restrictions). The two main approaches at your disposal are:

- **Security groups.** A security group acts as a firewall that controls the traffic allowed in and out of a group of instances. When you launch an instance in a VPC, you can assign the instance to up to five VPC security groups. **Security groups act at the instance level, not the subnet level**.
    - In general, it is a good idea to define distinct security groups for each tier. Doing so allows you to define the settings for each tier (and vary them independently) as well as restrict access to the "calling" tier (e.g., allowing the database tier to be called only from the application tier).
- **Network access control lists (ACLs).** A network ACL is an optional layer of security that acts as a firewall for controlling traffic in and out of a subnet. You might set up ACLs with rules similar to your security groups to add a layer of security to your VPC. **Network ACLs act at the subnet level, not the instance level.**

Security Groups

Here are the two approaches discussed in greater detail:

- Elastic Load Balancing:
    - Elastic Load Balancing is the point of contact for users, so the Elastic Load Balancing security group should be configured to support inbound client connection types of HTTP or HTTPS (port 80 and port 443, respectively). You can configure the Elastic Load Balancing in any combination, but Amazon recommends using HTTPS for both inbound client connection types. You should create an outbound security rule that lists the web tier security group as the target, restricting the load balancer to sending requests out to the web tier instances only.
- Web tier:
    - In the scenario, the web tier instances are not directly exposed but receive requests via the elastic load balancer. You can (and should) configure the web instances to accept requests only from the load balancer. Fortunately, the load balancer includes a special source security group. Create a security rule for your web tier that restricts inbound access to this special security group, ensuring that only the load balancers are allowed to send to and receive from the web front-end instances. You can also set up an outbound rule to limit outgoing requests to the application tier instances.
- Application tier:
    - As in the web tier case, your application tier security group should be configured with an inbound rule listing the web tier security group as an allowed sender and an outbound rule listing the database security group for outgoing messages.
- Database tier:
    - As in the other cases, you should require Secure Sockets Layer (SSL) for connections to and from SQL Server. Doing so requires the use of a security group with a rule that allows SSL (port 443) to be used only for the database instances.
    - You also want to restrict inbound access to the application tier instances, so create a security rule that restricts inbound access to the application tier security group.

The Appendix includes a chart detailing the various recommended security groups and settings for your SharePoint Server farm scenarios.

Network ACLs

Network ACLs mirror the rules specified in security groups and add an extra layer of security to allow general access rules to be honored regardless of which instances are sending or receiving. Because network ACLs act at the network level (not the instance level), you can set up additional rules to handle certain networks, IP addresses, and address ranges in a specific way. For instance, you can set up a network ACL that defines a rule to deny ingress to a range of source IP addresses (blacklisted IP addresses). For detailed guidance on setting up Amazon VPC network ACLs, see the Amazon Virtual Private Cloud User Guide.

Windows Instance Security

You can configure Windows instances within the VPC through Group Policy objects (GPOs) to require IP Security (IPsec) connections, further ensuring secure connectivity to the instances.

Administrator Access

In your architecture, the middle tier and database tier instances are placed in private subnets, restricting access from outside the VPC. This placement reduces exposure and enhances security. However, it is still necessary to provide access to those instances for administrative purposes, such as configuration updates and troubleshooting.

To help manage the instances in the private subnet, an indirect (and secure) method is to set up one or more bastion servers in a public subnet to act as proxies, and then set up SSH port forwarders or Remote Desktop Protocol (RDP) gateways to proxy access to the application or database tier instances. After bastion servers are set up, administrators can use RDP to gain access to the bastion host; they can then access other instances using SSH at their VPC private IP addresses. Figure 11 illustrates this arrangement.

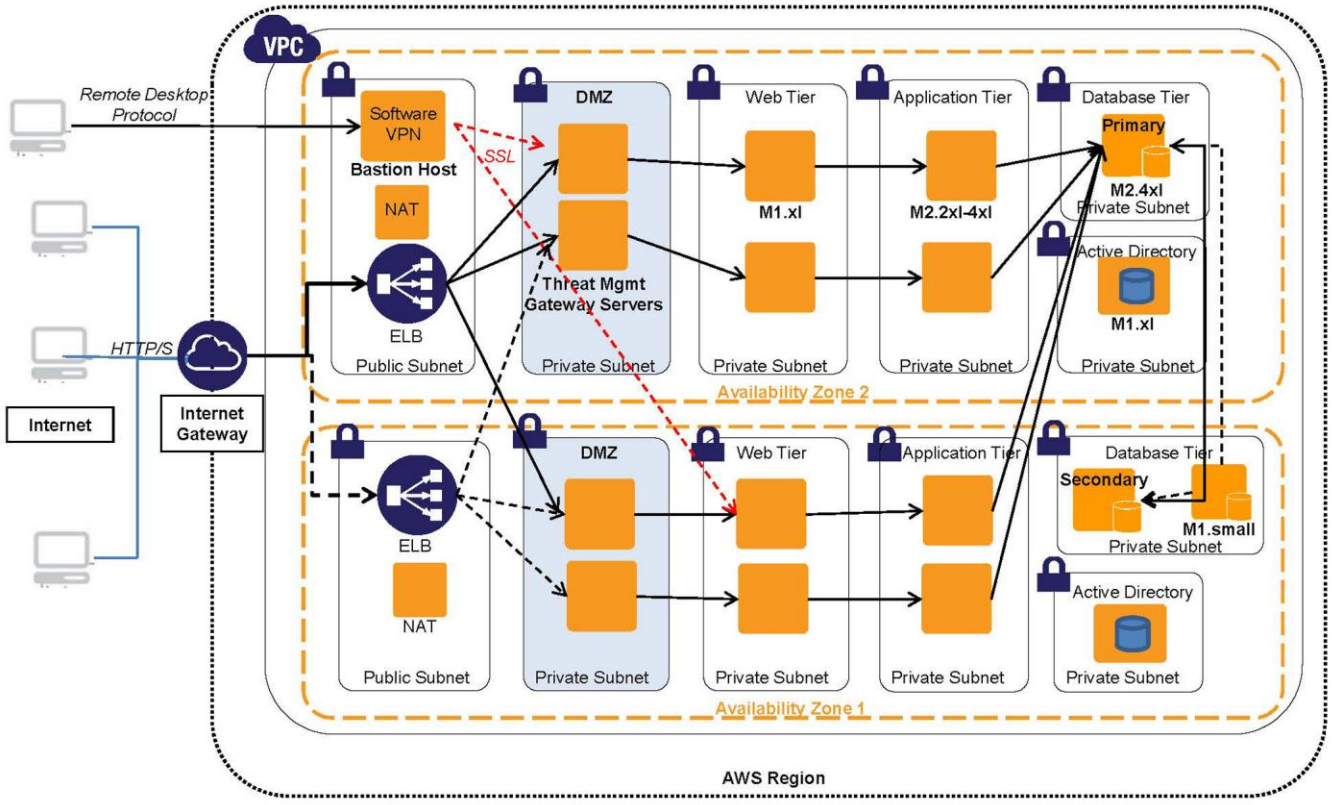


Figure 11: Using RDP to gain access to the bastion host

Data Privacy

Because sensitive content and data can be stored within the SharePoint Server farm, some organizations may require that the content be encrypted. To successfully support encryption of data within the AWS environment, a few key requirements must be considered and supported:

- **Encryption technology.** The Amazon EBS volumes contain the data at rest, in the form of SQL Server database data and files. Amazon EBS volume encryption is not supported in AWS; however, there are options for encryption that can be considered:
    - **Encrypting File System (EFS).** Windows includes EFS, which supports the ability to encrypt individual files or folders.
    - **BitLocker® Drive Encryption.** Windows Server 2008 R2 supports BitLocker, which provides the ability to encrypt a disk file system attached to the server instance.
    - **SQL Server Transparent Data Encryption (TDE).** SQL Server Enterprise provides native encryption support through TDE.
    - **Third-party Amazon EBS volume encryption.** Third-party commercial options are available for encryption of Amazon EBS volumes.
- **Encryption key management.** Implementing encryption requires secure management and authorized use of the encryption keys. In the case of Amazon EC2, instances can be stopped and started as well as recovered from Amazon EBS snapshots. In all these cases, the Amazon EBS volumes will be encrypted, and the Amazon EC2 subsystem must access and use the encryption key to be able to attach and use it on subsequent restarts.

The AWS Solution Provider site lists several third-party software vendors that provide security infrastructure that supports Amazon EBS encryption and key management.

Deployment

To set up your SharePoint Server farm in AWS, you must establish and configure several complex and interrelated details to enable proper functions and the correct security settings. Furthermore, you will inevitably need to change the configuration over time to perform such actions as adding instances for scale out or updating instance configurations.

AWS provides a number of tools and approaches for facilitating deployment in AWS:

- **AWS Management Console.** The AWS Management Console is an interactive tool that is good for starting out or smaller deployments. However, for more complex scenarios or automated deployment sequences, consider one of the other options described below.
- **AWS application programming interface (API) tools.** AWS provides several command-line interface (CLI) commands and programmatic web service APIs that are typically built into scripts; these commands allow a set of actions to occur in a coordinated way.
- **AWS sample code and libraries.** AWS provides a Sample Code & Libraries Catalog to support application-based setup and configuration. Several programming languages are supported through software development kits (SDKs) that AWS provides.
- **AWS CloudFormation.** AWS provides an easy way to create and manage a collection of related AWS resources, provisioning and updating them in an orderly and predictable fashion. With AWS CloudFormation, you do not need to figure out the order in which AWS services need to be provisioned or the subtleties of how to make those dependencies work:

  - You can use a tool called AWS CloudFormer to reverse-engineer an existing set of resources or settings running in an AWS account into an AWS CloudFormation template. So, a typical approach for a complex setup is to manually deploy or configure components of the SharePoint Server farm, and then use this tool to generate an appropriate AWS CloudFormation script.

    NOTE: AWS CloudFormation does not support the creation of VPCs at this time; however, it does support the creation of the resources within a VPC (e.g., Amazon EC2 instances, security groups).

- **Windows and .NET Developer Center.** These Windows and Microsoft .NET tools include the AWS SDK for .NET and the AWS Toolkit for Visual Studio.

A key approach to automating deployment of components within an AWS solution is to create custom AMIs for distinct roles that have additional software dependencies and configuration requirements. For the SharePoint Server reference architecture, distinct roles are defined (web front end, application server, database server, and others) for which you can create custom AMIs. Custom AMIs for the SharePoint Server farm architecture can be based on public Windows-based AMIs (as indicated earlier) or Windows-based AMIs that you create as a starting point.

Monitoring and Management

You must be able to monitor a number of core dimensions within a SharePoint Server farm to enable corrections and updates when issues occur or performance suffers. Amazon CloudWatch is an AWS service that monitors various health metrics associated with AWS resources. You can use it to collect, analyze, and view system and application metrics so that you can make operational and business decisions more quickly and with greater confidence. Amazon CloudWatch sets several predefined metrics, such as CPUUtilization and disk I/O performance, that AWS measures and that you can view and act upon. You can also publish your own metrics directly to Amazon CloudWatch to allow statistical viewing in the AWS Management Console and to issue (and react on) custom alarms.

Microsoft System Center Operations Manager is the typical tool used to monitor and manage a Microsoft-based infrastructure. Fortunately, Operations Manager can be used in AWS, too. The Windows-based infrastructure on AWS includes the standard Operations Manager agents for Windows Server, SharePoint Server, and SQL Server.

In the intranet scenario, Operations Manager works as it does in an on-premises case, because your VPN–VPC arrangement effectively extends the enterprise network into the AWS cloud. In the public site scenario, Operations Manager can be hosted in an instance and accessed over RDP (through the bastion host method described earlier) and provide monitoring and management against the other components of the SharePoint Server farm.

Backup and Recovery

Business continuity is a key requirement in the SharePoint Server farm scenarios discussed here. Downtime means core content and collaboration cannot occur or your website is down. As discussed earlier, you can improve availability by hosting multiple instances in different tiers distributed across AWS Availability Zones. However, there still may be situations in which system failures (e.g., because of software or hardware issues, disasters) occur, or there is a need to roll back or recover some or all of the farm data to a previous point in time. Thus, you must still have a backup and recovery strategy to support recovery of one or more data components or servers—or the entire farm.

Typically, recovery requirements are expressed in terms of two metrics:

- **Recovery time objective (RTO).** RTO is the time objective in which to restore a process, service, or data item to required functional level or accessibility. For example, an RTO of 4 hours means that a full recovery is required to be up and operational within 4 hours after a failure in the system.

- **Recover point objective (RPO).** The RPO is the maximum acceptable amount of data loss, expressed in time. For example, an RPO of 1 hour means recovered data may be at most 1 hour out of date from the most recent changes.

In terms of supporting backup and recovery of SharePoint Server farms on AWS, there are essentially two approaches to consider:

- Use the built-in back-up and recovery mechanisms in SharePoint Server and SQL Server, with Microsoft tools to back up to (and recover from) Windows file-based storage locations.

- Use AWS backup and recovery mechanisms that operate against AWS resources such as Amazon EBS volumes.

SharePoint Server and SQL Server provide their own built-in capabilities for backing up content, application data, metadata, and configuration settings. In addition, you can use tools such as Microsoft System Center Data Protection Manager (DPM) to back up configuration settings and metadata stored within SQL Server. Microsoft provides significant guidance around SharePoint Server backup and recovery that can and should be used to provide back-up and recovery capabilities, both at the farm level and at the granular server or service level. In this case, Amazon Simple Storage Service (Amazon S3) provides the most natural location in which to store and retrieve this data. Amazon S3 does not natively provide a Windows file system interface, but open source and commercial tools are available that do provide the ability to interact with Amazon S3 in this manner.

Amazon EC2 provides the ability to take point-in-time snapshots of Amazon EBS volumes and save them to Amazon S3 for durable storage and recovery. Amazon EBS snapshots are *incremental backups,* meaning that only the blocks on the device that have changed since the last snapshot will be saved. Also, when you delete a snapshot, only the data not needed for any other snapshot is removed. So, regardless of which prior snapshots have been deleted, all active snapshots will contain all the information needed to restore the volume. In addition, the time to restore the volume is the same for all snapshots, offering the restore time of full backups with the space savings of incremental backups.

Snapshots can also be used to instantiate multiple new volumes, expand the size of a volume, or move volumes across Availability Zones. In the case of your SharePoint Server farm, the SQL Server instances within the data tier will hold the persistent state, so taking regular snapshots of the primary SQL Server data tier Amazon EBS volumes provides backup of the database itself and any associated files (e.g., RBS files, metadata files).

AWS recently published AWS Disaster Recovery, a whitepaper that provides extensive details on the various considerations and options available within AWS to support disaster recovery.

Putting It All Together

With all the key topics covered, let's see how your SharePoint Server deployment scenarios are ultimately set up in an AWS environment.

Intranet SharePoint Server Farm

The key components of the intranet SharePoint Server farm in an AWS environment scenario are as follows:

- Amazon VPC, with VPN connection to the corporate data center
- Private subnets only, connected to the corporate network via VPN
- At least two Availability Zones used to survive the low probability of an Availability Zone failure
- Elastic load balancers across web front-end servers
- SQL Server in mirrored configuration across Availability Zones
- Database (Amazon EBS volume) snapshots

Figure 12 illustrates this scenario.

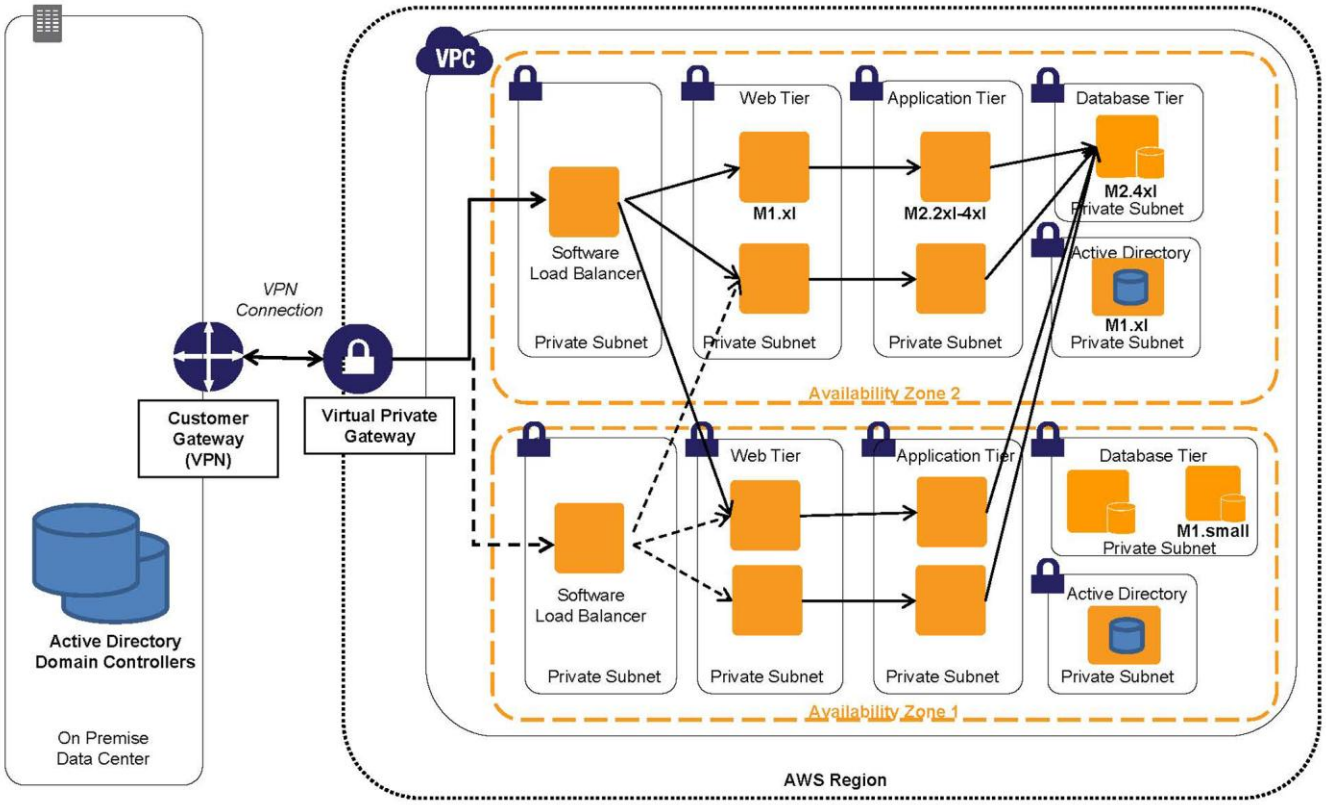


Figure 12: Intranet SharePoint Server farm in AWS

Internet-facing Public Website on SharePoint Server

The key components for the internet website hosted on SharePoint Servers in an AWS environment scenario are as follows:

- Amazon VPC, with public and private subnets
- Threat management gateway servers in the public subnet
- Elastic Load Balancing across the threat management gateway servers
- Bastion host in a public subnet, hosting a software VPN to provide administrative access to internal instances
- At least two Availability Zones used to survive the low probability of an Availability Zone failure
- Multiple web front-end servers behind threat management gateway servers within each Availability Zone in a private subnet
- SQL Server in mirrored configuration across Availability Zone private subnets
- AD DS domain controllers in AWS for user registration and authentication

Figure 13 illustrates this scenario.

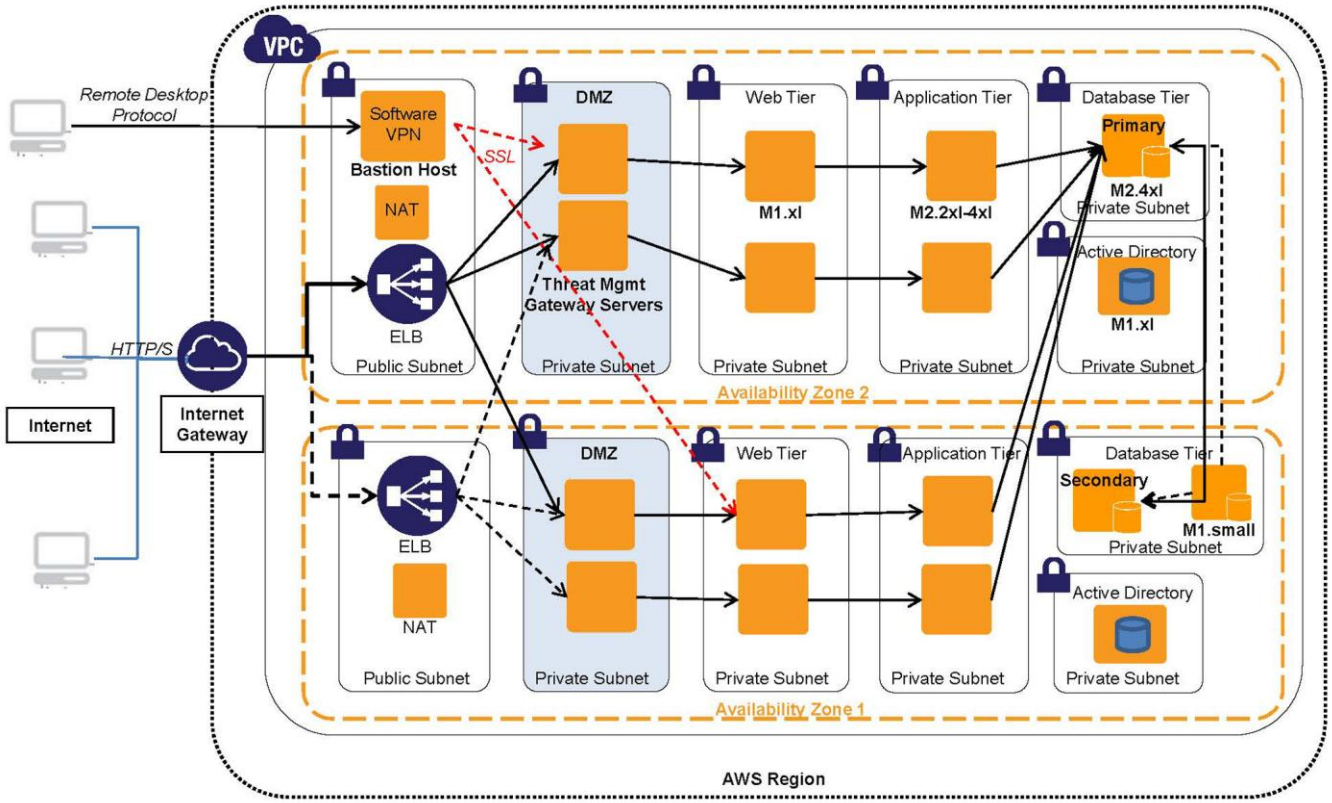


Figure 13: Public-facing Internet website on SharePoint Server in AWS

Although you can use SharePoint Server to support a variety of content and collaboration goals, these scenarios are two of the most common. See the next section for information about other scenarios and additional resources.

Conclusion

This paper discusses two common deployment scenarios for SharePoint Server—intranet and public website—and how to run them in an AWS cloud environment. It discusses how you can leverage different services that AWS provides (network setup, server setup, security, and deployment) and configure them specifically to run enterprise-class software like SharePoint Server at scale in a secure fashion that is easier to maintain.

Further Reading

- Microsoft on AWS:
  - http://www.awsmicrosite.com
- Amazon EC2 Windows Guide:
  - http://docs.amazonwebservices.com/AWSEC2/latest/WindowsGuide/Welcome.html?r=7870
- Microsoft AMIs for Windows and SQL Server:
  - http://aws.amazon.com/windows
  - http://aws.amazon.com/amis/Microsoft?browse=1
  - http://aws.amazon.com/amis/6258880392999312 (SQL Server)
- AWS Windows and .NET Developer Center:
  - http://aws.amazon.com/net
- Microsoft License Mobility:
  - http://aws.amazon.com/windows/mslicensemobility
- White papers:
  - "Amazon's Corporate IT Deploys SharePoint 2010 to the Amazon Web Services Cloud" at http://media.amazonwebservices.com/AWS_Amazon_SharePoint_Deployment.pdf
  - "Relational Database Management Systems in the Cloud: Microsoft SQL Server 2008 R2" at http://aws.amazon.com/whitepapers/rdbms-in-the-cloud
  - "Providing SSO to Amazon EC2 Apps from an On-premises Windows Domain" at http://download.microsoft.com/download/6/C/2/6C2DBA25-C4D3-474B-8977-E7D296FBFE71/EC2-Windows%20SSO%20v1%200--Chappell.pdf

Appendix

Security Group Settings for a SharePoint Server Farm

The following chart provides an example of the typical security group settings recommended for the SharePoint Server reference architecture.

Intranet SharePoint Server Farm

Tier/security group			Protocol	Port range	Comments
Elastic Load Balancing					
	Inbound	**Source**			
		IP address range of the corporate network	TCP	80	Allow inbound HTTP access from corporate IP sources
		IP address range of the corporate network	TCP	443	Allow inbound HTTPS access from corporate IP sources
	Outbound	**Destination**			
		WebTierSG	TCP	80	Allow outbound access to web tier servers
Web Tier					
	Inbound	**Source**			
		Elastic Load Balancing Source Security Group	TCP	80	Allow inbound HTTP from Elastic Load Balancing only
		Elastic Load Balancing Source Security Group	TCP	443	Allow inbound HTTPS access from Elastic Load Balancing only
		IP address range of corporate administrators	TCP	3389	RDP access for corporate administrators
		ActiveDirSG	TCP	49152–65535	AD DS
	Outbound	**Destination**			
		AppTierSG	TCP	0–65535	Allow only web front-end servers to access the application tier
		AppTierSG	UDP	0–65535	Allow only web front-end servers to access the application tier

Tier/security group			Protocol	Port range	Comments
		0.0.0.0/0	TCP	80	Allow outbound HTTP access to servers on the Internet (e.g., for software updates)
		0.0.0.0/0	TCP	443	Allow outbound HTTPS access to servers on the Internet (e.g., for software updates)
AppTier					
	Inbound	**Source**			
		WebTierSG	UDP	0–65535	Allow only web front-end servers to access the application tier
		IP address range of corporate administrators	TCP	3389	RDP access for corporate administrators
		ActiveDirSG	TCP	49152–65535	AD DS
	Outbound	**Destination**			
		DBTierSG	TCP	1433	Allow outbound SQL Server access to database tier instances
		0.0.0.0/0	TCP	80	Allow outbound HTTP access to servers on the Internet (e.g., for software updates)
		0.0.0.0/0	TCP	443	Allow outbound HTTPS access to servers on the Internet (e.g., for software updates)
		ActiveDirSG	TCP	49152–65535	AD DS
DBTier					Database primary, mirror, and witness
	Inbound	**Source**			
		App TierSG	TCP	1433	Allow only web front-end servers to access the application tier
		DBTierSG	–	–	Allow database mirror and witness
		IP address range of corporate administrators	TCP	3389	RDP access for corporate administrators
		ActiveDirSG	TCP	49152–65535	AD DS
	Outbound	**Destination**			

Tier/security group			Protocol	Port range	Comments
		ActiveDirSG	TCP	49152–65535	AD DS
		0.0.0.0/0	TCP	80	Allow outbound HTTP access to servers on the Internet (e.g., for software updates)
		0.0.0.0/0	TCP	443	Allow outbound HTTPS access to servers on the Internet (e.g., for software updates)
ActiveDirSG					
	Inbound	**Source**			
		ActiveDirSG	TCP	1–65535	Allow AD DS domains to talk to each other
		ActiveDirSG	UDP	1–65535	Allow AD DS domains to talk to each other
		0.0.0.0/0	TCP	53	DNS for VPC instance
		0.0.0.0/0	UDP	53	DNS for VPC instances
		0.0.0.0/0	TCP	88	Kerberos authentication
		0.0.0.0/0	UDP	88	Kerberos authentication
		0.0.0.0/0	UDP	123	Network News Transfer Protocol (NNTP)
		0.0.0.0/0	TCP	135–139	Remote Procedure Call (RPC), NetBIOS
		0.0.0.0/0	UDP	135–139	RPC, NetBIOS
		0.0.0.0/0	TCP	389	LDAP to directory service
		0.0.0.0/0	UDP	389	LDAP to directory service
		0.0.0.0/0	TCP	445	Server Message Block (SMB)
		0.0.0.0/0	UDP	500	IPsec Internet Security Association and Key Management Protocol (ISAKMP)
		0.0.0.0/0	TCP	636	LDAP Secure Sockets Layer (SSL)
		0.0.0.0/0	UDP	636	LDAP SSL
		0.0.0.0/0	TCP	3268–3269	LDAP to global catalog server
		0.0.0.0/0	UDP	4500	NAT traversal (NAT-T)
		0.0.0.0/0	TCP	49152–65535	Dynamic ports
		IP address range of corporate administrators	TCP	3389	RDP access for corporate administrators
	Outbound	**Destination**			

Tier/security group			Protocol	Port range	Comments
		0.0.0.0/0	TCP	80	Allow outbound HTTP access to servers on the Internet (e.g., for software updates)
		0.0.0.0/0	TCP	443	Allow outbound HTTPS access to servers on the Internet (e.g., for software updates)

Internet-facing Public Website on SharePoint Server

Tier/security group			Protocol	Port range	Comments
Elastic load balancer					
	Inbound	**Source**			
		0.0.0.0/0	TCP	80	Allow inbound HTTP access from corporate IP sources
		0.0.0.0/0	TCP	443	Allow inbound HTTPS access from corporate IP sources
	Outbound	**Destination**			
		WebTierSG	TCP	80	Allow outbound access to web tier servers
BastionSG					Security group for (public) bastion host
	Inbound	**Source**			
		IP address range of corporate administrators	TCP	3389	RDP access for corporate administrators
Web Tier					
	Inbound	**Source**			
		Elastic Load Balancing source security group	TCP	80	Allow inbound HTTP from Elastic Load Balancing only
		Elastic Load Balancing source security group	TCP	443	Allow inbound HTTPS access from Elastic Load Balancing only
		BastionSG	TCP	22	SSH access for corporate administrators
		ActiveDirSG	TCP	49152–65535	AD DS
	Outbound	**Destination**			

Tier/security group			Protocol	Port range	Comments
		AppTierSG	TCP	0–65535	Allow only web front-end servers to access the application tier
		AppTierSG	UDP	0–65535	Allow only web front-end servers to access the application tier
		0.0.0.0/0	TCP	80	Allow outbound HTTP access to servers on the Internet (e.g., for software updates)
		0.0.0.0/0	TCP	443	Allow outbound HTTPS access to servers on the Internet (e.g., for software updates)
AppTier					
	Inbound	**Source**			
		WebTierSG	UDP	0–65535	Allow only web front-end servers to access the application tier
		BastionSG	TCP	22	SSH access for corporate administrators
		ActiveDirSG	TCP	49152–65535	AD DS
	Outbound	**Destination**			
		DBTierSG	TCP	1433	Allow outbound SQL Server access to database tier instances
		0.0.0.0/0	TCP	80	Allow outbound HTTP access to servers on the Internet (e.g., for software updates)
		0.0.0.0/0	TCP	443	Allow outbound HTTPS access to servers on the Internet (e.g., for software updates)
		ActiveDirSG	TCP	49152–65535	AD DS
DBTier					DB primary, mirror, and witness
	Inbound	**Source**			
		App TierSG	TCP	1433	Allow only web front-end servers to access the application tier
		DBTierSG	–	–	Allow database mirror, witness

Tier/security group			Protocol	Port range	Comments
		BastionSG	TCP	22	SSH access for corporate administrators
		ActiveDirSG	TCP	49152–65535	AD DS
	Outbound	**Destination**			
		ActiveDirSG	TCP	49152–65535	AD DS
		0.0.0.0/0	TCP	80	Allow outbound HTTP access to servers on the Internet (e.g., for software updates)
		0.0.0.0/0	TCP	443	Allow outbound HTTPS access to servers on the Internet (e.g., for software updates)
ActiveDirSG					
	Inbound	**Source**			
		ActiveDirSG	TCP	1–65535	Allow AD DS domains to talk to each other
		ActiveDirSG	UDP	1–65535	Allow AD DS domains to talk to each other
		0.0.0.0/0	TCP	53	DNS for VPC instances
		0.0.0.0/0	UDP	53	DNS for VPC instances
		0.0.0.0/0	TCP	88	Kerberos authentication
		0.0.0.0/0	UDP	88	Kerberos authentication
		0.0.0.0/0	UDP	123	NNTP
		0.0.0.0/0	TCP	135–139	RPC, NetBIOS
		0.0.0.0/0	UDP	135–139	RPC, NetBIOS
		0.0.0.0/0	TCP	389	LDAP to directory service
		0.0.0.0/0	UDP	389	LDAP to directory service
		0.0.0.0/0	TCP	445	SMB
		0.0.0.0/0	UDP	500	IPsec ISAKMP
		0.0.0.0/0	TCP	636	LDAP SSL
		0.0.0.0/0	UDP	636	LDAP SSL
		0.0.0.0/0	TCP	3268–3269	LDAP to global catalog server
		0.0.0.0/0	UDP	4500	NAT-T
		0.0.0.0/0	TCP	49152–65535	Dynamic ports
		BastionSG	TCP	3389	RDP access for corporate administrators through a bastion host
	Outbound	**Destination**			

Tier/security group			Protocol	Port range	Comments
		0.0.0.0/0	TCP	80	Allow outbound HTTP access to servers on the Internet (e.g., for software updates)
		0.0.0.0/0	TCP	443	Allow outbound HTTPS access to servers on the Internet (e.g., for software updates)

For detailed guidance on setting up VPC security groups, see the Amazon Virtual Private Cloud User Guide.