Securing the Microsoft Platform on Amazon Web Services

Tom Stickle
August 2012

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

Abstract

Deploying Microsoft products on Amazon Web Services (AWS) is fast, easy, and cost-effective. Before deploying these applications to production, it is helpful to have guidance on approaches for securing them. The paper outlines the principles for protecting the runtime environment of applications running on AWS with a focus on risk assessment, reducing attack surface, adhering to the principle of least privilege, and protecting your data

Introduction to Amazon Web Services and the Microsoft Platform

Amazon Web Services (AWS) appeals to IT professionals in the Microsoft community because the AWS platform provides an alternative to pre-allocating compute, storage, and networks that require capital investment and lead time. This allows customers to offset large investments in data center infrastructure by using AWS for production, development, and testing of new or existing applications. AWS infrastructure also enables companies to rapidly spin up computing capacity that allows them to pay for what they use. When companies successfully complete their project, they can scale up quickly to meet demand, and if the project is put on hold or canceled altogether, infrastructure can be shut down at any time and they stop paying for it. Additionally, the broader AWS platform provides a rich set of services that, when combined with Microsoft products, can further enhance the experience of your users.

These Microsoft products together make up the Microsoft platform, a well-integrated set of operating systems, server roles, applications, and development tools that provide flexibility and business agility. The deep integration between applications and infrastructure makes it simple to manage and share enterprise data. The Microsoft platform has a strong focus on user experience, which reduces friction for users, managers, administrators, and developers. AWS is a perfect complement to the Microsoft platform because it allows administrators to rapidly provision “pay as you go” infrastructure that powers the Microsoft platform and applications designed to run on it.

AWS and Microsoft have worked together 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, for example, a SharePoint Server farm. This joint effort has further resulted in the ability to license and run SharePoint Server and other Microsoft Server products on AWS under provisions in the Microsoft License Mobility through Software Assurance program.

The Importance of System Security

We live in a world with an evolving threat landscape. Cyber-attacks are becoming much more sophisticated, targeted, and serious in nature. The individuals and groups responsible for these exploits have a wide variety of attack techniques at their disposal. The damage these attacks cause can range from minor vandalism to service disruptions and even to leakage of confidential data.

This paper addresses these security threats from the viewpoint of an IT Professional who is involved in the architecture, development, quality assurance, and ongoing operation of the Microsoft platform on AWS. The paper outlines the principles for protecting the runtime environment of applications running on AWS with a focus on risk assessment, reducing attack surface, adhering to the principle of least privilege, and protecting your data. The paper also focuses on techniques for protecting the confidentiality, integrity, and availability of data in the system.

Familiarizing yourself with the services available in AWS and how they interact with each other is an important step in understanding how to protect your system resources and data. The AWS Security Whitepaper is a great starting point for understanding the various services and how they are secured or isolated in their default state. This will help you to understand the protections that are in place and to determine whether you want to take additional steps to protect your resources and data.

Risk Assessment

Customers who choose to deploy the Microsoft platform on AWS are doing so because they want to provide innovative capabilities for their target audience. Whether that means disseminating information in a timely manner or aggregating it from groups of users, today's systems are highly connected. This paper acknowledges that there must be a balance between existing information security controls and the functional capabilities and economic cost of the system. Risk assessment is a technique for identifying and understanding risk, and more specifically the process of studying, analyzing, and describing the set of outcomes for particular threats.

The first step in assessing the risk of your application infrastructure is to make sure you are deeply familiar with the logical and physical architecture of the system. This includes the application tiers or subsystems and the network communication between the tiers and subsystems. It includes required resources such as SQL Server and the underlying configuration of AWS resources such as Elastic Block Storage (EBS) volumes. It should also include an understanding of who will be interacting with various system interfaces and from where.

Another important step in assessing risk is to do threat modeling to understand the potential threats to the system, determine the risk, and then develop the appropriate mitigations for the risk. A simple way to do this is to draw a diagram of your applications' subsystems and network entry points. You can then identify the threats for each interface and interaction, and then address each threat individually until your system is covered. This process is useful because it forces you to think about potential threats upfront, and it provides an opportunity to place controls into the infrastructure upfront.

When considering controls, it is important to adhere to the principle of "least privilege." This principle refers to users of the system having the least possible privilege necessary to perform their job function and no more. Least privilege also refers to the processes in the system having the least possible authority necessary to do their job as well. This helps reduce the *attack surface* of the system, making it much harder for an adversary to exploit. Attack surface can be defined as the set of exploitable vulnerabilities that we have; these would include the network, software, and humans that are involved in the ongoing operation of the system. Following the principle of least privilege, always look to reduce the attack surface of the system by exposing the absolute minimal set of ports to the network.

Another important mitigation is providing the controls necessary to protect against the accidental or deliberate misuse of confidential data. This refers to protecting the confidentiality, integrity, and availability of data in the system. Consequences for not adequately protecting data can range from the embarrassing (someone vandalizing website content) to the severe (leakage of sensitive data).

The following topics address a set of controls that you can implement to adhere to the principle of least privilege using AWS capabilities alongside the functionality that exists in the Microsoft platform. We also cover strategies for using controls to protect sensitive data in the system whether it's in transit or at rest.

Controls Available in AWS

AWS provides a set of building blocks that customers can use to build infrastructure for their applications. In this model, some security capabilities such as physical security are the responsibility of AWS and are highlighted in the AWS security whitepaper. Other areas such as controlling access to your application fall squarely on the application developer and the tools in the Microsoft platform. There is a gray area in between, though, where AWS has security capabilities that exist in the platform that you can configure to mitigate additional risks in their environment. We address these capabilities in this section.

Amazon Virtual Private Cloud

Amazon Virtual Private Cloud (Amazon VPC) is virtual networking technology that is a natural fit for running Microsoft platform components on AWS. Amazon VPC isolates the network and allows you to create your own network subnets and divide application layers into network subnets for a greater level of control. In addition, VPC allows you to specify your own static IP addresses or DHCP from a range of known IP addresses in a way that is friendly to both Microsoft Active Directory and the many applications that rely on it. You can easily customize the network configuration for your VPC and you can select your own private IP address ranges, create subnets, and configure route tables and gateways.

In addition to providing isolation and familiar multi-layer networks, Amazon VPC also supports multiple connectivity options. With these options, you can:

- Decide to allow the instances on your subnet to just send, or to send and receive, traffic from the Internet.
- Connect securely to your own corporate data center using an industry standard virtual private network (VPN) appliance or software VPN.
- Connect multiple office locations via site-to-site VPN tunnels and use the cloud as a router.
- Cross-connect your corporate network with your VPC using a dedicated 1 Gb or 10 Gb connection using AWS Direct Connect.

These options enable seamless integration between on-premise compute infrastructure and resources within the VPC. Leveraging VPN-VPC connectivity extends the corporate 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.

In Figure 1, we use multiple subnets to isolate various tiers of the infrastructure. We have one “public subnet” that is capable of receiving Internet traffic, and that’s where we decided to place our AWS Elastic Load Balancer. We then use network routes to allow communication between subnets, but we do so using our principle of least privilege. For example, it is a requirement that the application tier can send traffic to the database tier, but there is no reason for the web tier to ever directly access the database tier.

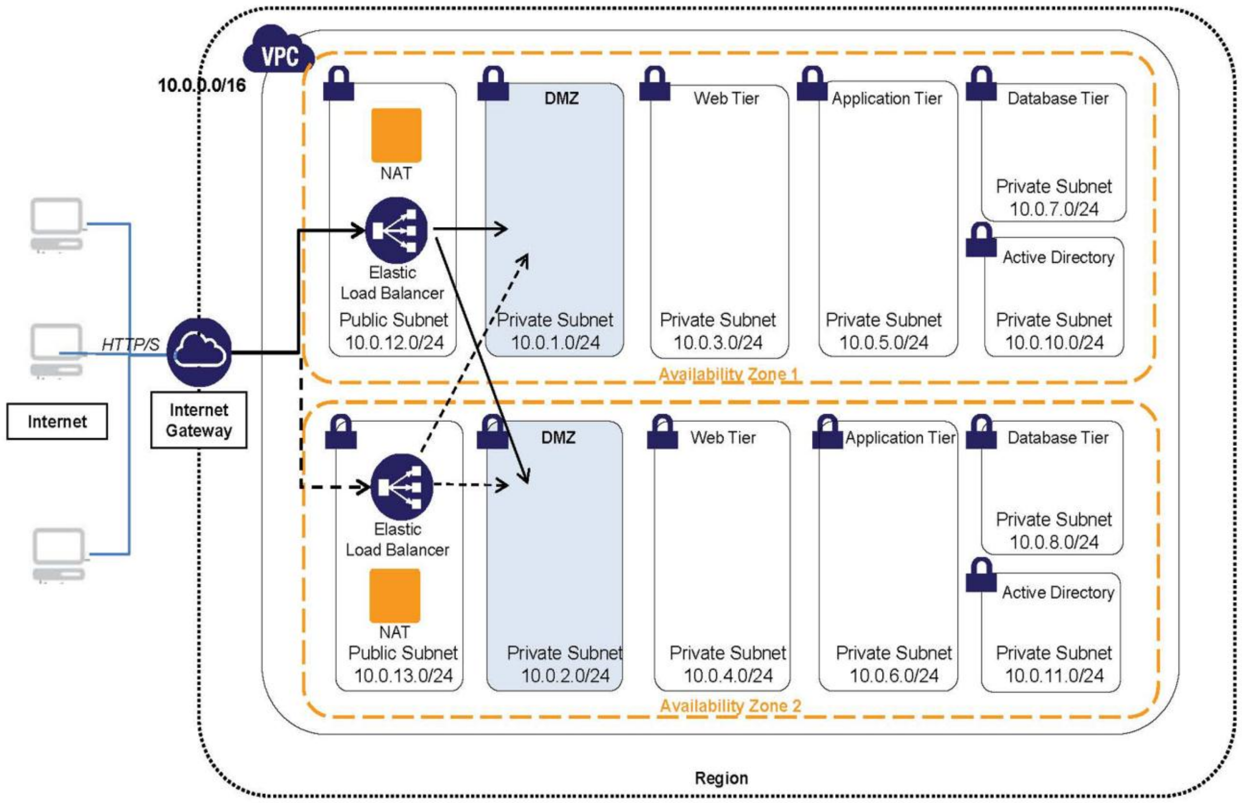
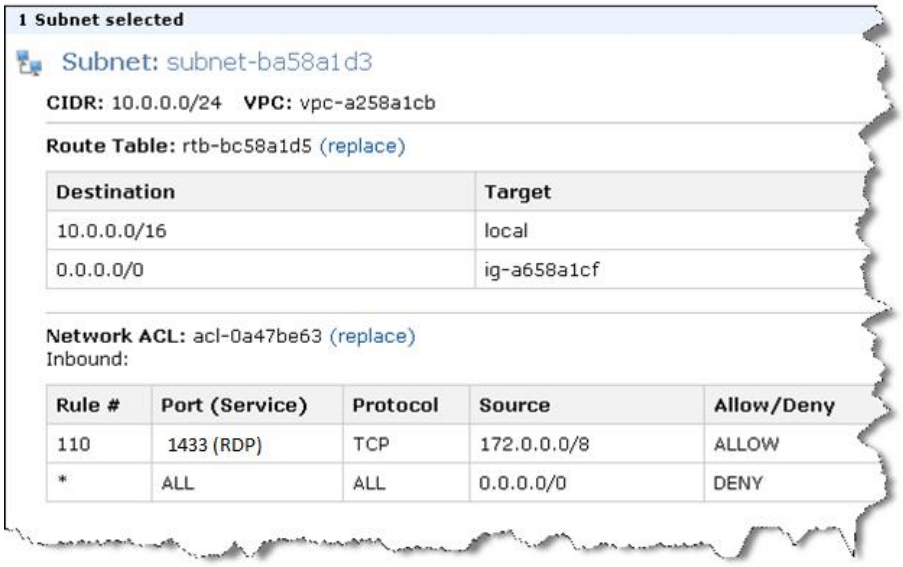

VPC
10.0.0.0/16
NAT
Elastic Load Balancer
Public Subnet 10.0.12.0/24
DMZ
Private Subnet 10.0.1.0/24
Web Tier
Private Subnet 10.0.3.0/24
Application Tier
Private Subnet 10.0.5.0/24
Database Tier
Private Subnet 10.0.7.0/24
Active Directory
Private Subnet 10.0.10.0/24
Availability Zone 1
HTTP/S
Internet
Internet Gateway
Elastic Load Balancer
NAT
Public Subnet 10.0.13.0/24
DMZ
Private Subnet 10.0.2.0/24
Web Tier
Private Subnet 10.0.4.0/24
Application Tier
Private Subnet 10.0.6.0/24
Database Tier
Private Subnet 10.0.8.0/24
Active Directory
Private Subnet 10.0.11.0/24
Availability Zone 2
Region

Figure 1 – Isolated environment with full Internet access

In a more sophisticated network topology, we may allow connectivity from some of the private subnets directly to our data center network. That technique can be used to integrate with our existing internal systems. Likewise, it would be common to allow for administrative access to subnets using VPN-VPC connectivity and tightly controlled routes. All of these topologies can be created with Amazon VPC in a manual or automated fashion, and serve as the foundation for any Microsoft platform application built on AWS.

Network Access Control Lists

Network access control lists (ACLs) can be attached to any network subnet in a VPC and provide a way for you to do stateless filtering of traffic. Network ACLs can be used for inbound or outbound traffic and provide an effective way to blacklist a CIDR block or individual IP addresses. These ACLs can contain ordered rules to allow or deny traffic based upon IP protocol, service port, or source/destination IP address. Figure 2 shows how you could set a rule that would allow administrative traffic to come inbound on port 1433 from a specific set of IP addresses.



1 Subnet selected

Subnet: subnet-ba58a1d3

CIDR: 10.0.0.0/24 **VPC:** vpc-a258a1cb

Route Table: rtb-bc58a1d5 (replace)

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	ig-a658a1cf

Network ACL: acl-0a47be63 (replace)
Inbound:

Rule #	Port (Service)	Protocol	Source	Allow/Deny
110	1433 (RDP)	TCP	172.0.0.0/8	ALLOW
*	ALL	ALL	0.0.0.0/0	DENY

Figure 2 – Network ACL to allow range of IP addresses to SSH inbound

Security Groups

Security groups are one of the most critical tools we have to isolate our infrastructure on Amazon EC2. All EC2 instances are required to belong to one or more security groups. Security groups enable the AWS administrator to set policy for controlling open ports, and to set policy for providing isolation between application tiers. In Amazon VPC, every instance runs over a stateful firewall that runs on the host with all ports closed by default. The security group is responsible for opening up ingress and egress ports on that firewall. For example, you could have a security group called “webtier” that has rules to open port 80 and 443. You could then run 10 webservers that are all part of the “webtier” security group. If you later decide that you just want to support HTTPS traffic from the web server, you can simply close port 80 in the “webtier” security group. All 10 instances will immediately respect this change and start blocking traffic from surfacing on port 80.

Security groups provide much more than firewall policy, though. You can use them to lock one tier of your application to another for much better control over the isolation of the system. For example, suppose you create a security group to run your SQL Servers in. In that security group, you can specify that you will allow traffic on port 1433, but only from members of the security group containing your SharePoint servers. This provides an additional layer of protection in addition to using VPC subnets and network routes to provide network isolation. It allows for more granular control, which allows you to further reduce the attack surface. Later in this paper, we highlight some specific usage scenarios for security groups when we discuss how to use them to protect your application.

AWS Identity and Access Management (IAM) and IAM Roles

When running application infrastructure on AWS, you have two different administrative functions. One class of administrators is responsible for interacting with AWS to start and stop resources, attach volumes, rotate credentials, and configure Auto Scaling. Another class of administrators is responsible for connecting to Windows servers and setting up Active Directory, installing patches, and joining domains. IAM was designed to provide control over the former class of administrators; those who are responsible for the AWS environment. IAM enables classes of administrators, allowing you to specify policy for access control.

Let's use the example of an IT organization with an infrastructure team and an application team. The infrastructure team controls the network topology, and the application team is not allowed to modify network resources as a matter of policy. A separate IAM group could then be used for the application administrator that does not allow members to create, delete, or modify any VPC resources (subnets, VPN Gateways, VPCs). Adhering to the principle of least privilege, you might then decide that you want to limit the infrastructure team and prevent them from starting or stopping Amazon EC2 instances. You could also take away their ability to create Elastic Block Store (EBS) volumes, attach them, and perhaps also prevent modification of Auto Scaling groups. Lastly, you may want to ensure that none of these administrators has access to the rolled-up billing information for the account. All of this can be done with AWS IAM.

Since AWS offers a wide range of services that applications can consume, applications are often required to possess credentials for those services. Rather than embedding IAM users' credentials on the Amazon EC2 instance to access a particular service, users can use IAM Roles. IAM Roles allow you to launch an Amazon EC2 instance with a predetermined set of IAM authorizations whose credentials will be made available to the Amazon EC2 instance. The credentials will automatically be rotated several times a day, and they can be retrieved programmatically via the EC2 Instance Metadata Service. For example, your application may need to grab mapping tiles from Amazon Simple Storage Service (Amazon S3) so they can be processed. You could create a new IAM role called "S3Access" that has read-only access to Amazon S3 and no other service access. You can then launch your instance with this IAM role assigned and access the appropriate credentials from the instance without the need to store the credentials in persistent storage.

Increasing Availability

A common approach to high availability of enterprise applications is to eliminate single points of failure. For example, rather than run a single SharePoint Web Front End, you can run two or more in an active-active configuration. This practice helps protect against an application failure or an isolated hardware failure of one of the servers. With AWS, you have an additional layer of protection by employing Availability Zones. Availability Zones are fault-separated locations within a region that are physically separated from each other. Availability Zones have their own utility power and sufficient network diversity to help protect against the failure of a particular network provider. By placing redundant subsystems of your application into different Availability Zones, you can help protect against macro issues such as a utility power failure or a core router failure.

Most traditional data center environments have only the network bandwidth to support their operations at standard peak volumes. Unlike these traditional environments, AWS has a great deal of network diversity and bandwidth, which provides customers with an additional layer of protection from external network threats that attack available bandwidth.

Deploying and Protecting Security Infrastructure

In order to reduce threats to your application, you need to combine some AWS controls with controls that are available in the Microsoft platform. In this section, we look at specific pieces of security infrastructure that can be set up generically for your applications. We then focus on applying the principle of least privilege and locking down that infrastructure. Every Microsoft application deployed in AWS will have a requirement for the following:

- Remote Server Administration
- Patches and Updates
- Active Directory for Authentication, Access Control, and Policy Management

Remote Server Administration

Rather than connect directly to administer each machine, we can proxy administrative connections using the Remote Desktop Gateway (RD Gateway) role service in Windows Server 2008 R2. This service provides an additional layer of security and offers security controls that can mitigate threats related to administrative access to servers. At a minimum, RD Gateway consolidates the audit logs for administrative access, but it also offers strong authentication mechanisms based on SSL mutual authentication and the ability to apply authorization policies. For example, you can specify that clients need to be members of Active Directory groups and then specify which network resources those groups can connect to. You can also specify whether device redirection is involved.

Remote Desktop Gateways have another interesting capability in that they can be placed in a public subnet and then configured to allow remote administration over an Internet connection, through a VPN-VPC connection, or through AWS Direct Connect, depending on your needs. Communications to RD Gateway are done via HTTPS (through port 443), which allows you to configure the gateway at the network edge if you want to. The RD Gateway will then proxy communications to the backend servers via the typical RDP protocol, which is transmitted by default to port 3389.

The AWS Management Console provides a wizard-based approach to setting up Amazon VPC environments for a few typical Amazon VPC configurations. For the initial scenario, the goal is to set up the AWS environment to enable corporate users to access EC2 instances via VPN access; but you don't 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. For an additional layer of security, we divided the application tiers into a set of subnets. Given this extra layer, we can create two private subnets to demonstrate the use of the gateway to proxy RDP connections.

In order to install the Remote Desktop Gateway, simply start a Windows Server 2008 R2 server in one of your private subnets. The subnet must be routable from the Virtual Private Gateway, and the wizard in the console will ensure that your initial subnets are routable. Connect to the Windows Server and in the server manager add the "Remote Desktop Services" role, and then the "Remote Desktop Gateway" role service. You will need to add a valid certificate and configure the proper authorization policies as part of the configuration.

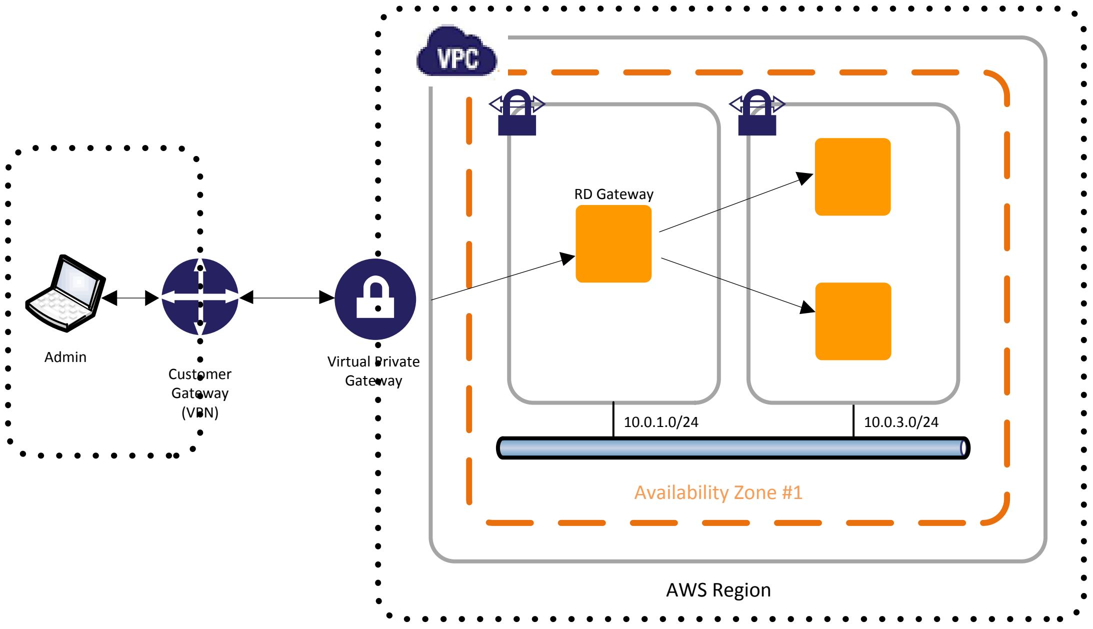


Figure 3 – Minimal Configuration of Remote Desktop Gateway on AWS

Reducing the Attack Surface of the Remote Desktop Gateway Infrastructure

In the case of the Remote Desktop (RD) Gateway, AWS security groups are the key component in stopping administrators from bypassing the RD Gateway and connecting directly with instances. By placing the RD Gateway into its own unique security group, we can instruct other security groups to accept traffic only on port 3389 (RDP) from the RD Gateway's security group. This technique can be used as an additional layer of protection between application layers in addition to the use of routing rules between network subnets. Figure 4 shows the addition of the security groups.

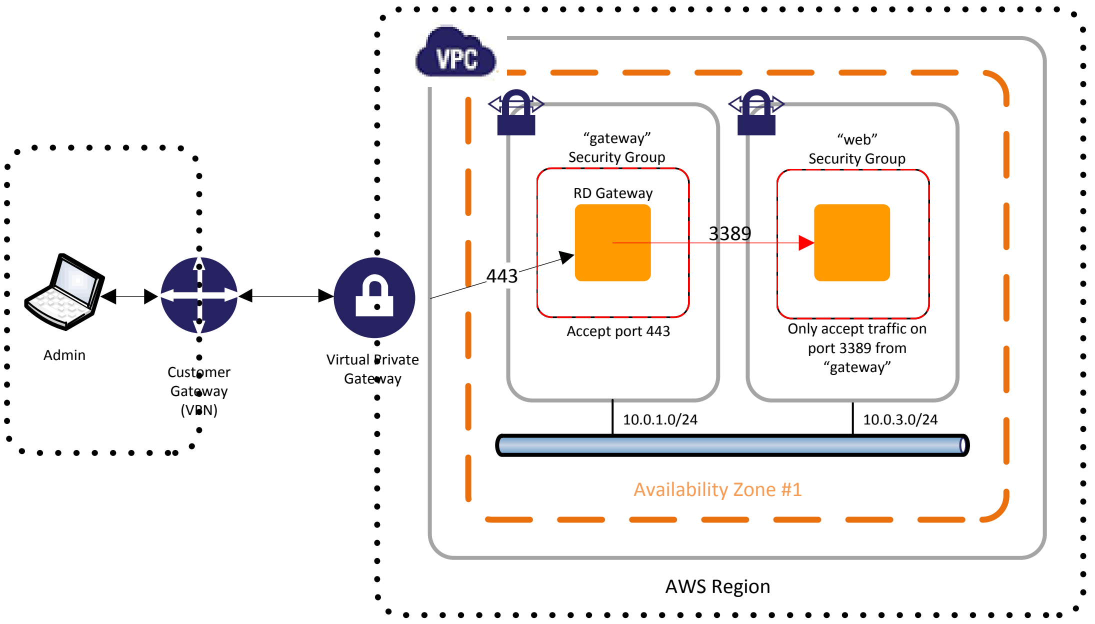


Figure 4 – Remote Desktop Gateway with Security Groups

As a security best practice, you want to always look to reduce the attack surface on your interfaces. You can do this with security group rules, which should be as specific and granular as possible to always keep open the least number of ports possible. Administrators on the network will access the RD Gateway, so we recommend restricting access to the specific machines used for administration rather than allowing connections from any machine on the network. Figure 5 shows a more precise look at the security group settings for the RD Gateway and how other security groups should be tied to it.

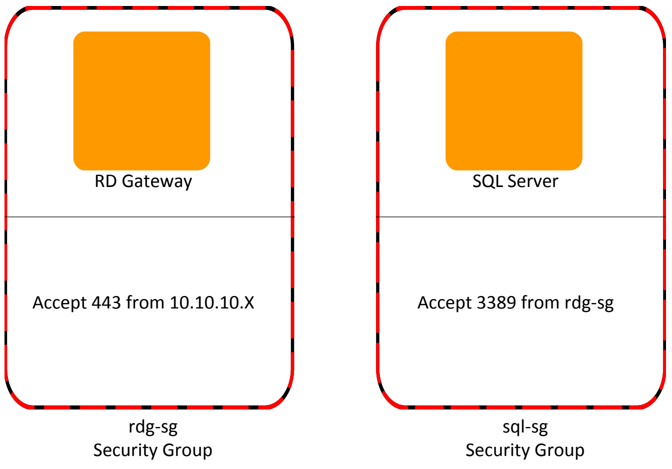


Figure 5 – Security Group Rule Configuration (Least Privilege)

In this case, we specifically accept inbound HTTPS traffic on port 443 from a specific machine or a CIDR block for a set of machines. From the security group that is protecting a SQL Server, we allow RDP connections, but only from the rdg-sg, which represents the RD Gateway security group. This control prevents a "bypass attack" by restricting machines from the corporate network or VPC from being able to connect directly into the servers via Remote Desktop Protocol (RDP).

NOTE: It is important that RDP is **never** opened up to the entire Internet—not even for testing purposes or temporarily. Always restrict ports and source traffic to minimum necessary to support the functionality of the application.

Increasing Availability of the Remote Desktop Gateway Infrastructure

A single VPC can have multiple subnets in which each subnet resides in a separate Availability Zone. Each subnet must reside entirely within one Availability Zone. As a best practice, we recommend running applications in a redundant configuration across multiple Availability Zones to provide additional fault separation for the application. For availability, you can then place a Remote Desktop Gateway in each of two private subnets that are running in separate Availability Zones.

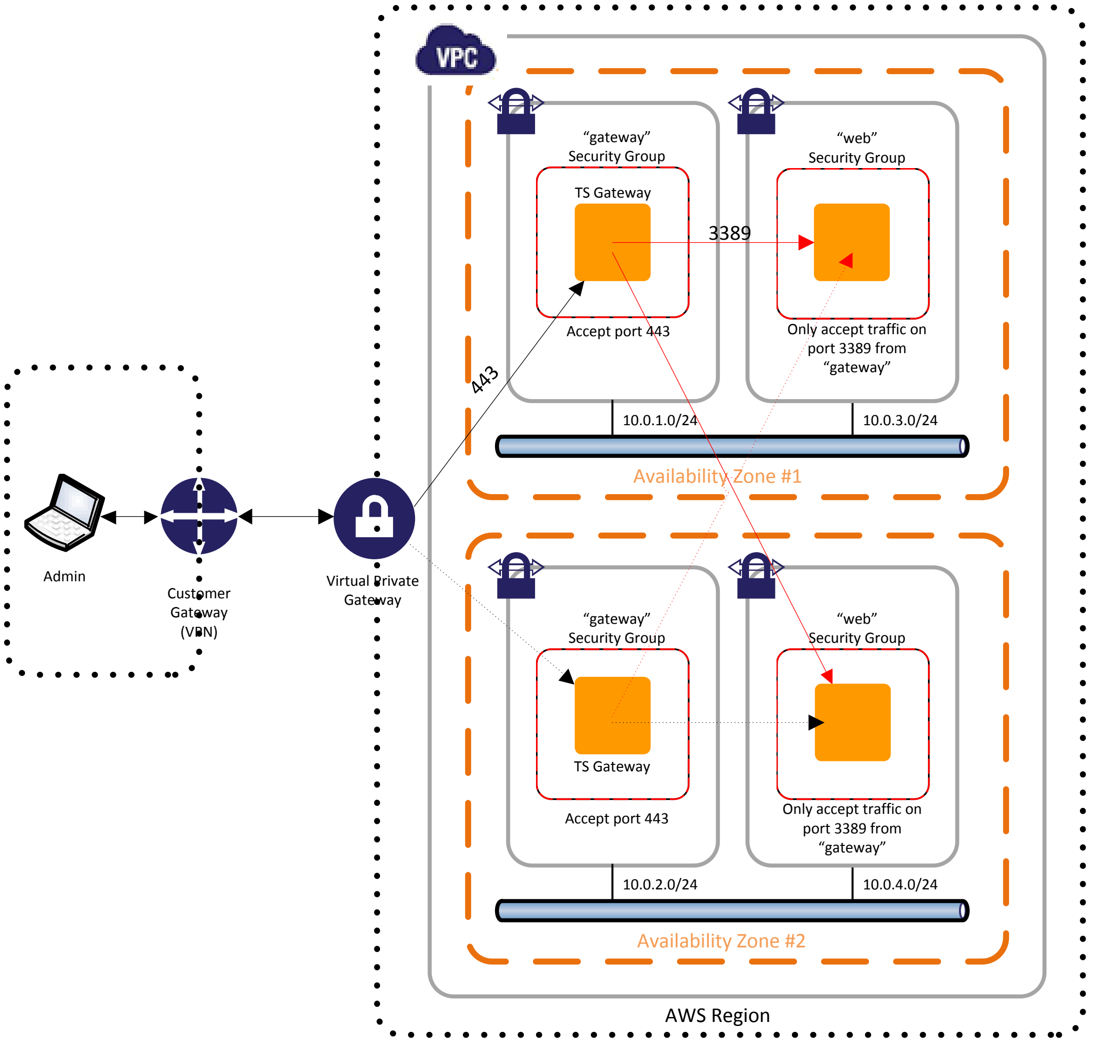


Figure 6 – Highly Available Remote Desktop Gateways for Privileged User Access

Patches and Updates

Keeping your patches up to date is an extremely important part of protecting your infrastructure. The Windows Server Update Services (WSUS) is the mechanism typically used to deploy patches to your servers in the environment. You always have the option of pointing your servers at a WSUS server that is running on your own data center network. The key factor in deciding this, though, is really the amount of infrastructure that needs to be updated and its potential to saturate the bandwidth of the VPN-VPC or AWS Direct Connect connection. For example, having hundreds of servers in the VPC grabbing updates at the same time would likely saturate the available bandwidth in the connection. You can partly mitigate this saturation by configuring updates with the Background Intelligent Transfer Service (BITS) with throttling enabled, but then you are just shifting the problem and increasing transfer latency.

By installing a Windows Server Update Services (WSUS) Server in the VPC, you can minimize the bandwidth requirements on the VPN-VPC connection, and you can manage the deployment of updates to your environment from within the VPC. There are three different types of configuration scenarios for deploying WSUS in the VPC:

- WSUS obtains updates directly from Microsoft Update via the Internet
- WSUS obtains updates from Microsoft Update via your corporate data center's Internet access
- WSUS obtains updates from an upstream WSUS server on your corporate network

To enable scenario #1, you can use a network address translation (NAT) instance in the VPC and configure it with a routable IP address. 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. To keep organized, you can place your WSUS server in the same "Admin Tier" as the Remote Desktop Gateway that you configured in the prior section. You can also create a new "public subnet" that contains your NAT instance and route to the Internet gateway. Figure 7 illustrates this update and shows the additional communication flow from the Internet.

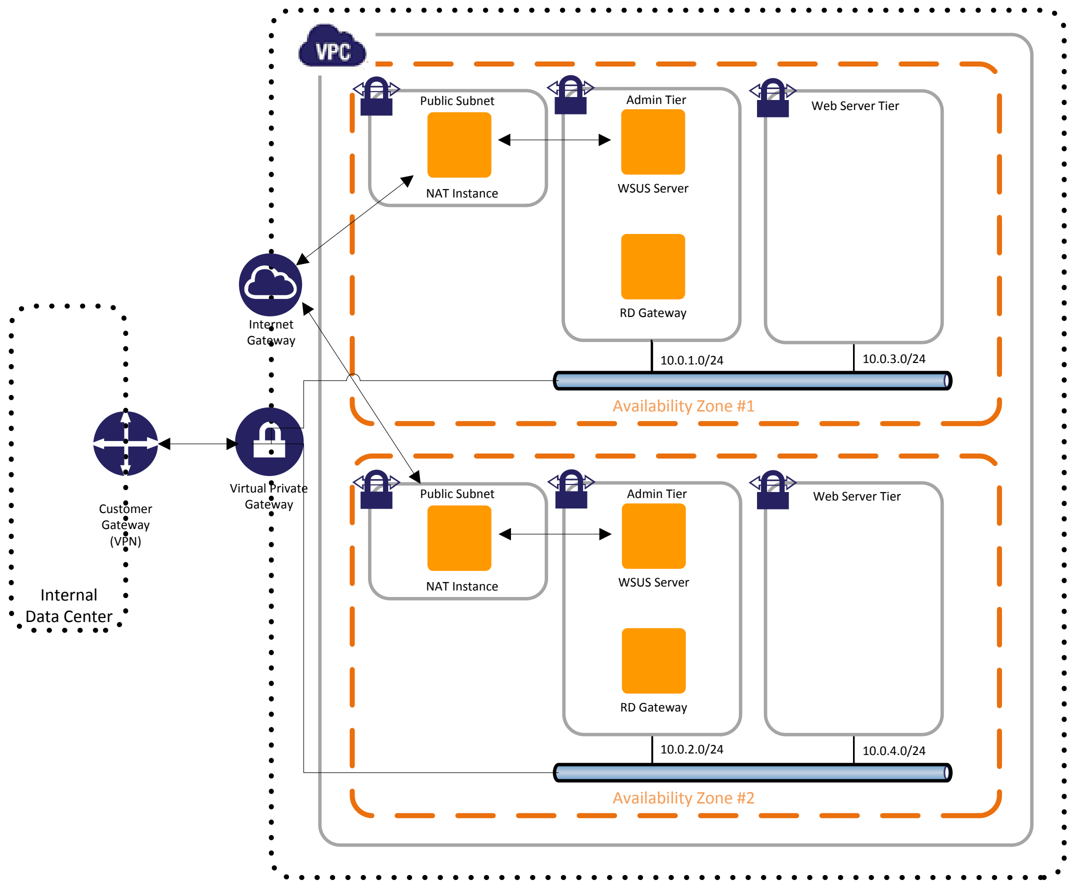


Figure 7 – Deployment of WSUS Server Using a NAT Instance and VPC Internet Gateway

Since the topology for scenario #2 and #3 is simple, you simply create a new private subnet for the “admin” tier and don’t use an Elastic IP and a route to the Internet gateway.

Just as we did with the Remote Desktop Gateway, we recommend using security groups to provide an additional layer of protection between the WSUS server and the clients that are accessing it. We also recommend enabling SSL for the communication between your servers and the WSUS server as an additional layer of protection. You can configure the SSL connection on the WSUS server on port 8531. In the case where you have a Remote Desktop Gateway, a WSUS Server, and a SQL Server, you could use the following security groups:

Security Group	Source IP/SG	Inbound Port Rules
rdg-sg	10.0.0.0/16 (Admin IP Range)	TCP 443
wsus-sg	rdg-sg	TCP3389, TCP8531
	sql-sg	TCP8531
sql-sg	rdg-sg	TCP3389

Table 1: Security Group Configuration for Remote Desktop Gateway

This means that the administrators on the intranet at the address range you specify for the Remote Desktop Gateway can communicate on port 443. The WSUS server can then be contacted by the Remote Desktop Gateway on port 3389 for administration and port 8531 to obtain updates. This also enables the SQL Server to obtain updates on port 8531. The SQL Server is restricted so that it can be managed only by the Remote Gateway on port 3389. If you added another layer that needed to communicate with the SQL Server database, you could then add its security group and open port 1433 to enable communications.

Active Directory for Authentication, Access Control, and Policy Management

As the centerpiece of the Microsoft platform, Active Directory facilitates authentication, authorization, and name resolution across the servers in the infrastructure. AD Domain Controllers can run within AWS and be replicated from on-premise domain controllers using the VPN-VPC connection. This action allows the servers to authenticate to local domain controllers, but still authenticate to corporate user identities and credentials. Although it is certainly possible to directly connect to corporate domain controllers through the VPN-VPC connection, replicating those services provides for better performance and is the recommended approach. We always recommend replicating the domain controllers across Availability Zones (as with your other resources) to provide high availability.

NOTE: It is also possible to support this scenario for corporate environments that do not use AD, but rather another Lightweight Directory Access Protocol (LDAP)–based directory service. You can use Active Directory Federation Services (AD FS) 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 8 shows the infrastructure needed for hosting AD FS in AWS using the approach highlighted in this section.

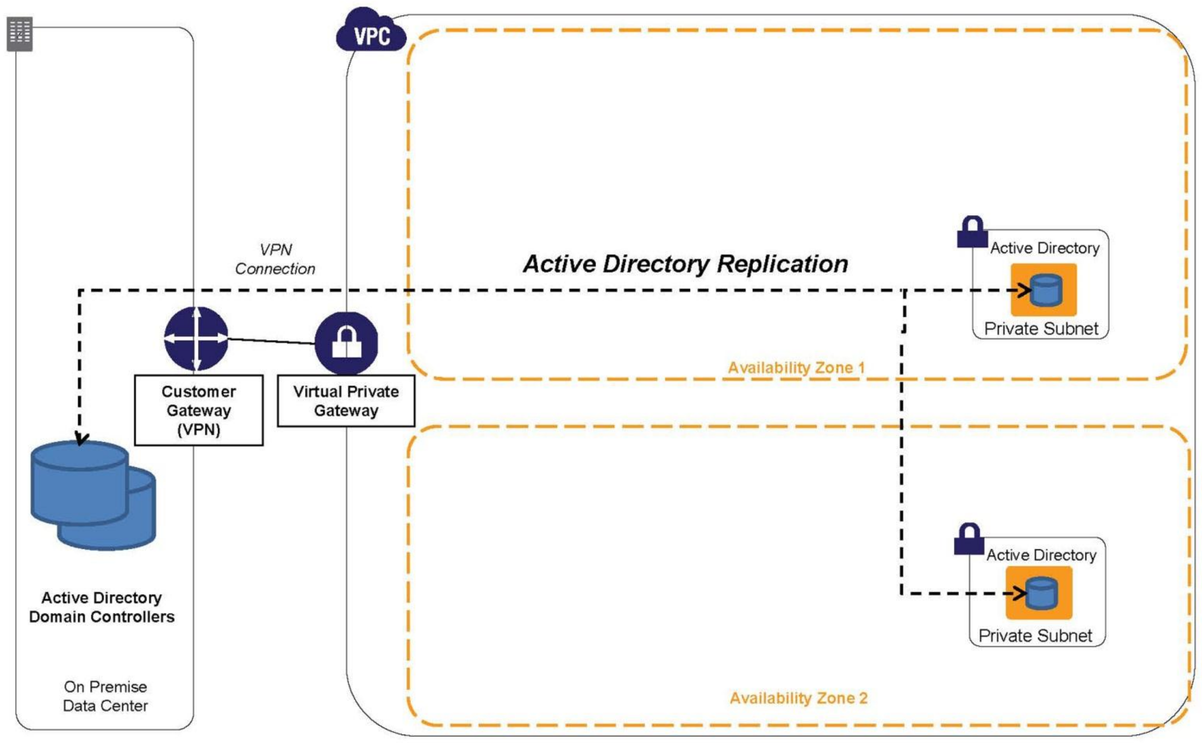


Figure 8: Active Directory Deployment in AWS with Replication

Protecting the Application Infrastructure

In this section, we focus on the runtime environment and employ techniques to isolate and reduce the attack surface while ensuring the availability of the application infrastructure. It is a good security practice when designing the architecture of your application to define all of the interfaces between layers and specify how they should be protected. For example, if the only consumer of information from the database is an application, there is no reason to allow the database to be reachable from other layers of the application. So, just as you did with the remote administration environment, you want to ensure that you minimize the attack surface by opening only the necessary ports. You also want to protect against the accidental or purposeful bypass of your security controls by locking layers together using security groups.

A useful technique for analyzing interfaces is to create a block diagram of the application subsystems and the relationship between them. Figure 9 illustrates the major subsystems of an arbitrary SharePoint application and the interaction between them:

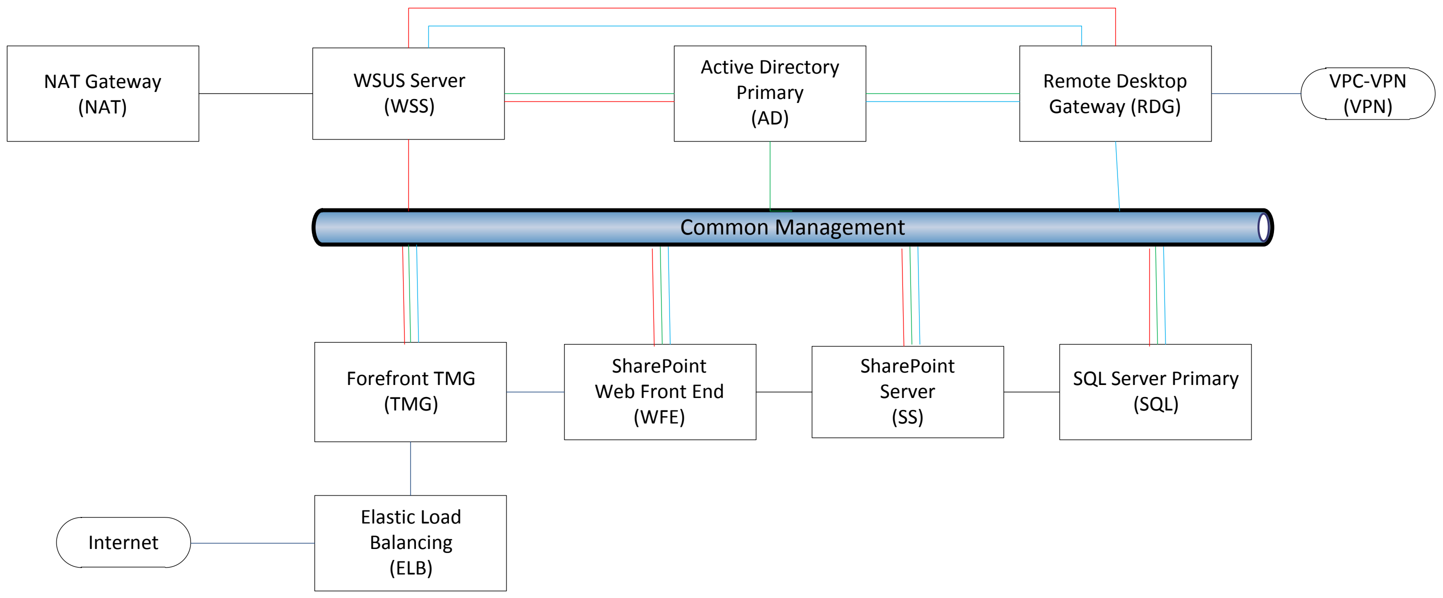


Figure 9: Major Subsystems and Interactions

Please see the appendix 1 (Subsystem Interface Port Mapping) for detailed information on the port configuration that follows the principle of least privilege between these subsystems.

You can then break the subsystems down into smaller "modules" if they have any interactions between them. For example, SQL Server uses mirroring to synchronously replicate data between an active node and a passive node. Active Directory uses asynchronous communication to replicate data between nodes.

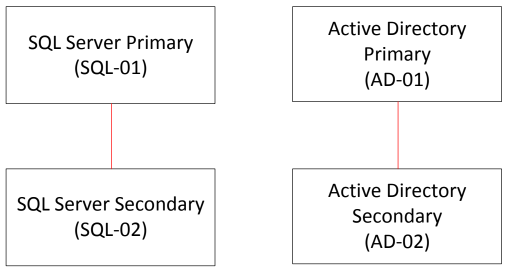


Figure 10: Module Interactions

Please see the appendix 2 (Module Interface Port Mapping) for detailed information on the port configuration that follow the principle of least privilege between these modules.

You can then repeat that exercise for the outbound interactions to compile a complete list of the interactions between modules and subsystems. The next step is to translate this compilation into a set of EC2 security groups. Members of the same security groups can communicate with each other on any ports. By placing AD-01 and AD-02 into a single AD-SG security group, for example, the two modules will be able to communicate with each other on all of the ports required. For this reason, you can enforce least privilege by using the Windows firewall to explicitly open the ports you need.

Please see the appendix 3 (Subsystems with External Port Mappings) for a detailed table of the security groups and associated rules for this example.

Centralized Threat Protection

Insider threats have caused many information security professionals to rethink their perimeter protection strategy. As a result, perimeter protection is collapsing from the edge of the network and getting much closer to the application assets. Trusting the users and resources in the LAN doesn't make much sense when a large portion of attacks are originating from inside the network. It makes much more sense to isolate application infrastructure entirely and tightly control access for both internal and external users and administrators. Having a central location to enforce policy for the perimeter of the application infrastructure is an important part of protecting the application. Microsoft's TMG server is a perimeter protection product that can be centrally managed and adds the following capabilities:

- URL filtering
- Black listing IP addresses
- Network inspection
- Granular HTTP controls
- DOS protections
- Enterprise policy enforcement
- External logging capabilities
- Traffic inspection

NOTE: There is one best practice you will need to succeed in installing TMG server from a client that is behind a NAT device (most clients). If you do an RDP install from your local machine, TMG will add your private IP address to the RDP ALLOWED list. Once it is installed and you try to reconnect, TMG will pick up your public NAT address, and since it won't see it in the ALLOWED list, it will drop your connection. To work around this, install a TMG by starting two Windows instances. Connect to the first instance, and then use RDP to get to the second instance where you can successfully complete the installation. After rebooting, RDP back into the TMG server using the first Windows server, and configure it as necessary.

Placing TMG server at the ingress point for each of your applications provides a point where you can deploy threat protection policy. Figure 11 shows how TMG can be deployed in this fashion.

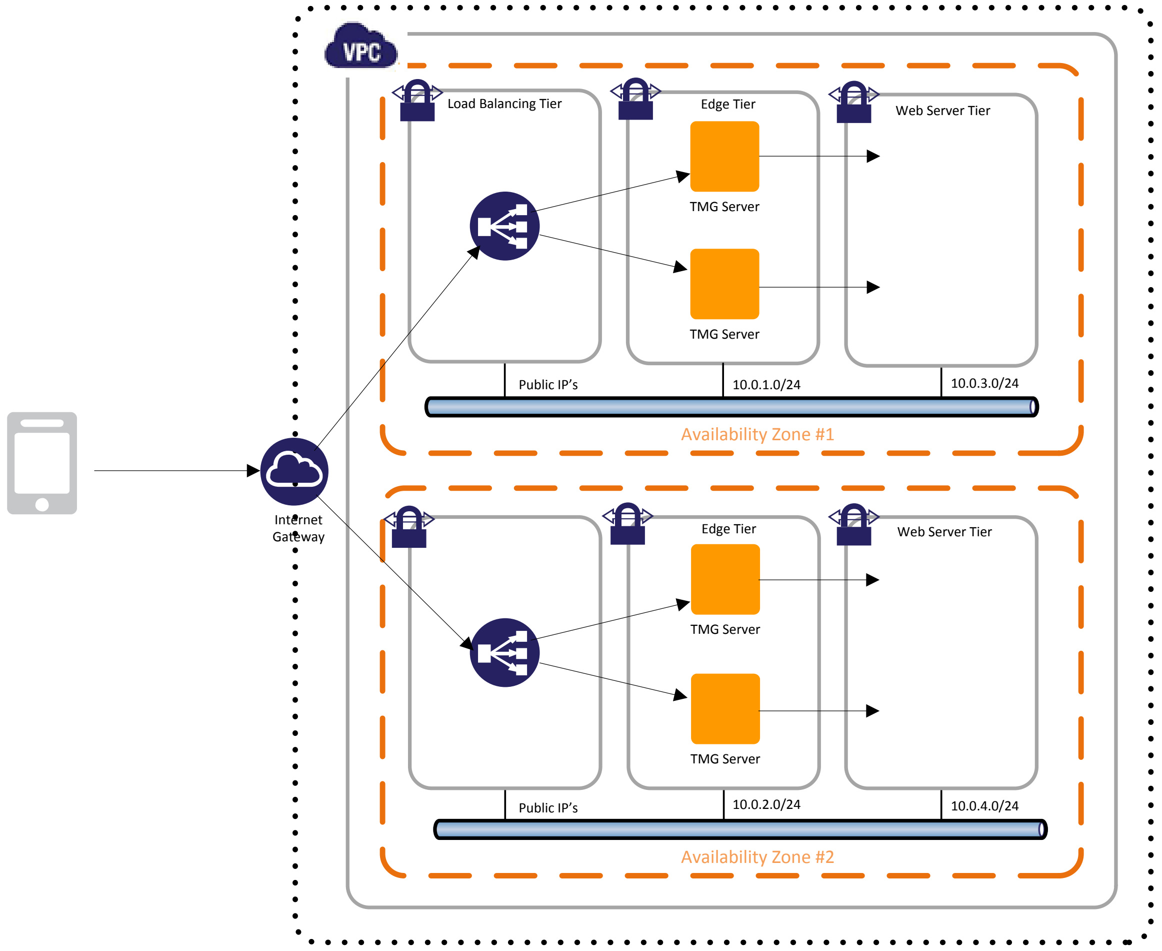


Figure 11: Using Forefront TMG for Threat Protection and Policy Enforcement

Firewall Usage

Every Amazon EC2 instance in VPC has a built-in stateful ingress and egress firewall that denies access to all ports by default; customers must explicitly allow traffic to ports. All ports on the firewall are closed by default, so it is up to the administrator to open ports using Amazon EC2 security group rules. Although it is common practice to use security groups for this purpose, it may be more desirable in some cases to leverage the Windows firewall in your Amazon EC2 instance in addition. In particular, the Windows firewall will give you audit detail about packet drops, which may be important to meet your security policy or compliance requirements.

Figure 12 incorporates the concepts discussed in this section into one illustration.

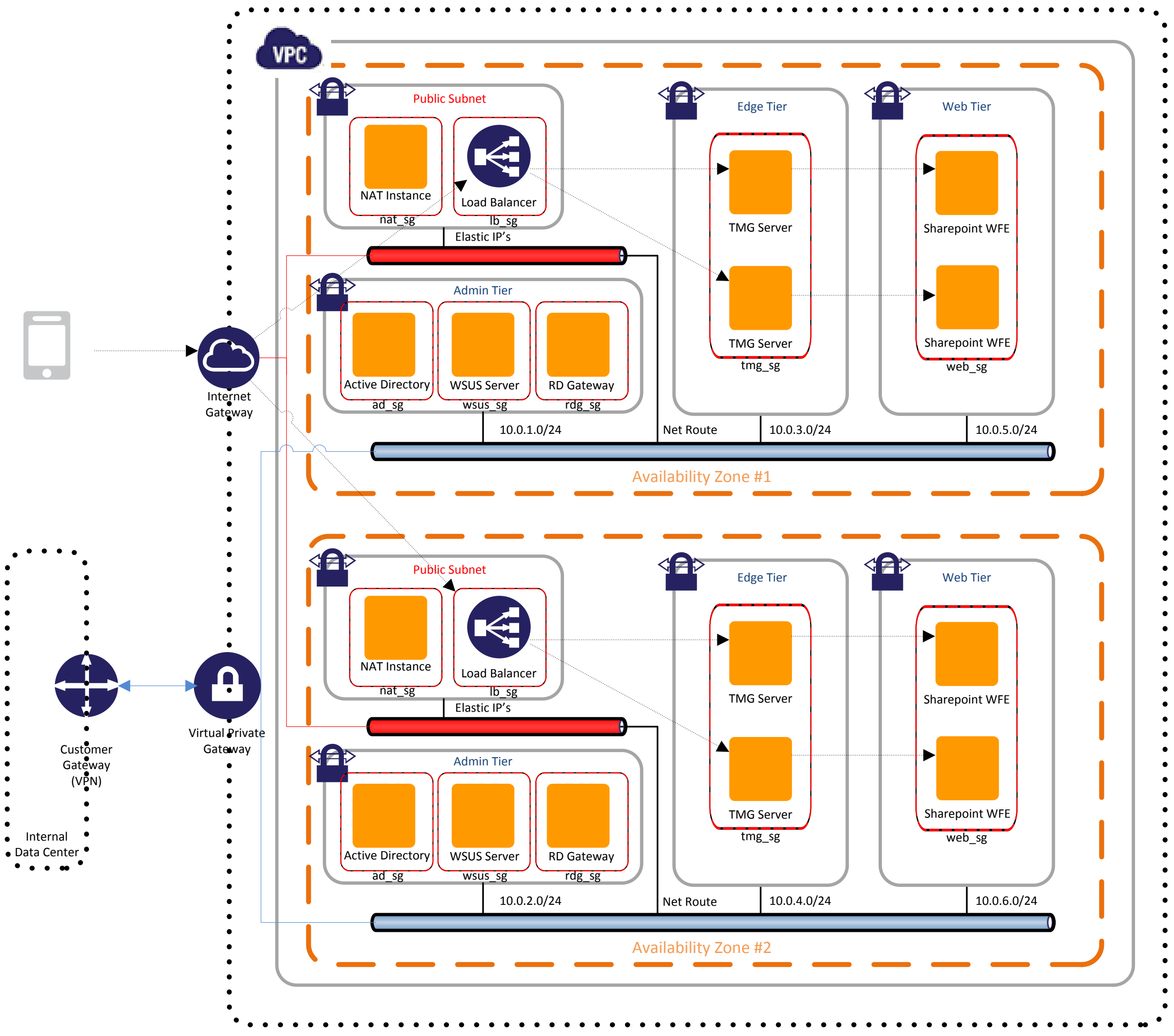


Figure 12 – Overview of Protecting the Application Infrastructure

Using Identity and Access Management to Lock Down Controls

AWS Identity and Access Management (IAM) allows you to control access to AWS resources and can be used as a way to protect and federate your infrastructure management. For example, you may have one group of administrators for the infrastructure and another for applications placed in the infrastructure. You can enable access to modify the VPC subnets, routes, and ACLs to the infrastructure team, but not allow the application team to modify them. This mechanism is also useful for federating management in a large enterprise and providing security controls to make sure that an application administrator can't open a security hole in the infrastructure.

Increasing Availability of the Application Infrastructure

Active redundancy is the technique that is commonly used to provide high availability for enterprise applications on the Microsoft platform. You implement this technique by having an active-active configuration of application components so that if one fails, the other is available to keep the application running. This principle does not change when running on

AWS, but the scale of the AWS infrastructure allows for an additional level of protection called Availability Zones. Availability Zones are fault separated areas of an AWS region where you can place application infrastructure. Availability Zones have different utility power providers and are in geographically separated physical data centers.

By separating the active-active components in two Availability Zones, you add additional resiliency that you couldn't achieve with a single data center. A load balancer balances traffic between components in both Availability Zones. Load balancing provides a loose coupling between the application consumer and the infrastructure. This abstraction allows the infrastructure to grow or shrink without affecting the ability to serve clients. AWS provides a load balancing service called Elastic Load Balancing that can automatically scale based on the traffic it receives. This is the cloud paradigm where you simply specify that you want a load balancer, but you don't need to worry about sizing it. You can place a load balancer into each Availability Zone when using VPC and assign a security group to it. This allows you to lock the front end of your application to the load balancer's security group.

While your load balancer is capable of automatically scaling as needed, you still need to consider how to scale the infrastructure running behind it. AWS has a capability called Auto Scaling, which you can use to automatically scale up or scale down infrastructure based on a pattern such as network traffic. You can also place PowerShell scripts on the instance so that when it is auto scaled, it can be configured properly in the environment before accepting traffic from the load balancer.

Anti-Virus

Microsoft recommends the use of anti-virus software on all domain controllers and any server that interacts with the domain controllers. Please refer to the Microsoft's guidance for the recommendations for virus scanning. This same guidance applies to your servers running in AWS, as it is extremely important to stop malware as early as possible. Anti-virus software will have an impact on performance for files that rapidly change such as database storage files. For this reason, Microsoft provides guidance on files to exclude on the server when running anti-virus scans.

Vulnerability Scanning and Penetration Testing

The AWS Acceptable Use Policy strictly forbids doing any kind of port scanning in the infrastructure, but there is a process in place if you want to run a vulnerability scan or penetration test of your own Amazon EC2 Instances. A vulnerability scanner is a process that can assess resources and interfaces to enumerate vulnerabilities present in the target. For example, a vulnerability scanner can detect that you left a port open to the entire Internet or find that you have weak crypto algorithms enabled for an SSL interface. A penetration test is different because it can find vulnerabilities, and it can then exploit them to determine their significance.

You can use the following link to whitelist a set of instances for a vulnerability scan or a penetration test. Note that you will have to be very specific about the instances you will be testing and the time that you will be testing them. Once you have your infrastructure standing, you can use a vulnerability scanning tool to evaluate the perimeter of your application and detect any vulnerability.

Summary

You can also review the AWS Security Whitepaper for a complete overview of the security processes available in AWS. The following control checklist summarizes the various techniques that are outlined in this paper:

✓	**Action**	**Comment**
	Data classification scheme	It is important to classify different types of data in order to properly assess the controls necessary to protect the data.
	Develop threat models	Threat modeling is a process to model potential threats to the system so you can determine the risk and develop mitigations.
	VPC subnet for each tier	A subnet for each application tier including administration tiers.
	Subnets spread across Availability Zones	Each tier has been built in a separate Availability Zone for high availability
	Active-active redundancy	Application components are duplicated for high availability and placed into separate Availability Zones.
	Network ACLs are used	Network ACLs are used to restrict traffic into the VPC subnets.
	Security groups are used	A separate security group defined for each application component that sends or receives network traffic. Security group rules reflect the principle of least privilege.
	IAM is used for AWS administrators	Identity and Access Management is used to differentiate classes of administrators. It is an important service for federating management responsibilities.
	Secure administration	Remote Desktop Gateway configured to proxy connections and write audits for remote server administration.
	Ability to distribute updates	Connectivity and/or WSUS server configured to deploy updates through an Internet gateway in the VPC or through the Internet connection in the corporate data center.
	Active Directory	Active Directory is placed in a highly available configuration using static IP addresses and is available for managed servers in the environment. It is used for authentication, access control, and group policy management for the infrastructure.
	Advanced auditing	A group policy is used to specify a balanced audit policy for all servers in the environment.
	Audit logs reviewed	Audit logs are reviewed on a regular basis to identify any attempts to circumvent the controls that are in place.
	Central threat protection	A chokepoint in the infrastructure that you can use to apply security policy such as URL filtering, IP blacklisting, and traffic inspection.
	Anti-virus deployed	Anti-virus should be used on all Active Directory domain controllers and all servers that are connected to them.
	Vulnerability scan	Whitelist the appropriate instances and run a vulnerability scan such as NESSUS to assess vulnerabilities that you may have missed.
	Data isolation	Ensure you have the appropriate boundary around your data depending on its classification. The techniques to apply are physical isolation, logical isolation, and cryptographic isolation.
	Regular backups	Ensure that you are taking regular backups of your data and storing them in Amazon S3 for durability.

References and Further Reading

1. Windows on Amazon EC2 Security Guide - http://aws.amazon.com/articles/1767
2. AWS Security and Compliance Center – http://aws.amazon.com/security/
3. Overview of AWS Security processes Whitepaper – http://media.amazonwebservices.com/pdf/AWS_Security_Whitepaper.pdf
4. AWS Risk and Compliance Whitepaper - http://media.amazonwebservices.com/AWSRiskandComplianceWhitepaperJanuary2012.pdf
5. AWS Application Security Best Practices - http://media.amazonwebservices.com/Whitepaper_Security_Best_Practices_2010.pdf
6. Using Windows ADFS for Single Sign-On to Amazon EC2 - http://media.amazonwebservices.com/EC2_ADFS_howto_2.0.pdf
7. Vulnerability Scan and Penetration testing - http://aws.amazon.com/security/penetration-testing/
8. Microsoft's guidance on Recommendation on Virus Scanning - http://support.microsoft.com/kb/822158
9. Active Directory Federation Services - http://www.microsoft.com/en-us/server-cloud/windows-server/active-directory-overview.aspx
10. Amazon VPC User Guide http://docs.amazonwebservices.com/AmazonVPC/latest/UserGuide/index.html

Appendices

Appendix 1: Subsystem Interface Port Mappings

Subsystem	Inbound Interface	Port(s)
NAT	WSS -> NAT	TCP 443
WSS	AD, RDG, TMG, WFE , SS, SQL -> WSS	TCP8531
	RDG -> WSS	TCP3389
AD*	WSS, RDG, TMG, WFE, SS, SQL -> AD	UDP123, TCP135, UDP135, TCP1024-65535, TCP137, UDP137, TCP139, TCP445, UDP445, TCP389, TCP636, TCP88, UDP88, TCP53, UDP53, ICMP8, ICMP13, ICMP15, ICMP17
	RDG -> AD	TCP3389
RDG	VPN -> RDG	TCP443
TMG	RDG -> TMG	TCP3389
	ELB -> TMG	TCP80
WFE	RDG -> WFE	TCP3389
	TMG ->WFE	TCP80
SS	RDG -> SS	TCP3389
	WFE ->SS	TCP56737
SQL	RDG -> SQL	TCP3389
	SS -> SQL	TCP1433
ELB	Internet -> ELB	TCP443

* Active Directory incorporates many different services and protocols; the following link shows the complete list of ports that are used http://support.microsoft.com/kb/832017. Many ports need to be open to authenticate between an instance and the primary domain controller. It is a common practice to open all ports on the AD server and use source traffic restrictions to control access. For example, you could open all ports on the AD server to the SQL Server's Security Group. This practice is not following our principle of least privilege though, and it is a far better practice to get familiar with the specific ports you need and enable those to be open and no more.

Appendix 2: Module Interface Port Mappings

Module	Inbound Interface	Port(s)
SQL-01	SQL-02 -> SQL-01	TCP5022
SQL-02	SQL-01 -> SQL-02	TCP5022
AD-01	AD-02 -> AD01	TCP135, UDP135, TCP137, UDP137, UDP138, TCP139, TCP1024-65535, TCP445, UDP445, TCP389, UDP389, TCP636, TCP3268, TCP3269, TCP88, UDP88, TCP53, UDP53
AD-02	AD-01 -> AD02	TCP135, UDP135, TCP137, UDP137, UDP138, TCP139, TCP1024-65535, TCP445, UDP445, TCP389, UDP389, TCP636, TCP3268, TCP3269, TCP88, UDP88, TCP53, UDP53

Appendix 3: Subsystems with External Port Mappings

Security Group	Subsystem	Source IP/SG	Inbound Port Rules
nat-sg	NAT	wss-sg	TCP 443
wss-sg	WSS	ad-sg, rdg-sg, tmg-sg, wfe-sg ,ss-sg, sql-sg -> wss-sg	TCP8531
		rdg-sg	TCP3389
ad-sg	AD	wss-sg, rdg-sg, tmg-sg, wfe-sg ,ss-sg, sql-sg -> wss-sg	UDP123, TCP135, UDP135, TCP1024-65535, TCP137, UDP137, TCP139, TCP445, UDP445, TCP389, TCP636, TCP88, UDP88, TCP53, UDP53, ICMP8, ICMP13, ICMP15, ICMP17
		rdg-sg	TCP3389
rdg-sg	RDG	(IP range of Admins)	TCP443
tmg-sg	TMG	elb-sg	TCP80
		rdg-sg	TCP3389
wfe-sg	WFE	tmg-sg	TCP80
		rdg-sg	TCP3389
ss-sg	SS	wfe-sg	TCP56737
		rdg-sg	TCP3389
sql-sg	SQL	ss-sg	TCP1433
		rdg-sg	TCP3389
elb-sg	ELB	0.0.0.0/0	TCP443