**AWS Application Load Balancer with an IaaS Web Application**



Author(s) / Document Version Control Information

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Author** | **Description** | **Version** |
| 3/30/2020 | Sayyed Mohammed Arshad | Initial Revision | 1.0 |

**TABLE OF CONTENTS**

[1 Recipe Targets 2](#_Toc11664231)

[2 Recipe Scenario 2](#_Toc11664232)

[3 Recipe Architecture 2](#_Toc11664233)

[4 Pertinent Cloud Component Details](#_Toc11664234) 4

[5 Constraints and Limitations](#_Toc11664235) 6

[6 APPENDIX A – Control Reference](#_Toc11664236) 7

# Recipe Targets

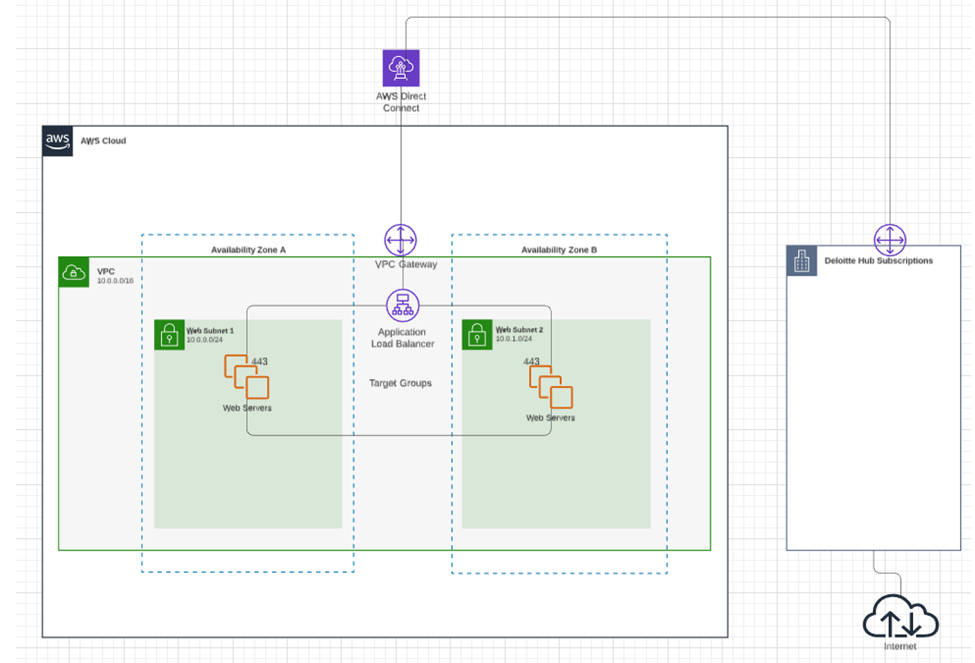
* **Cloud Platform:** AWS
* **Application Type**: Internal application with sensitive data. Lift and Shift applications in general.
* **Compliance Target**: SOC2 Type 2, ISO 2700x
* **Data Classifications**: PII, PHI, Customer Confidential, and Firm Confidential
* **Primary Cloud Services:** AWS Application Load Balancer, IaaS, Deloitte HUB
* **Geographies**: All

# Recipe Scenario

This recipe targets Web applications hosted in a cluster of AWS IaaS EC2 Instances leveraging any OS and web server such as IIS. The scenario targets internal Deloitte applications that need to be compliant with PM40 and other Deloitte Cyber Security standards. The recipe was used specifically for a globally hosted US application that utilizes OneCloud provisioned EC2 Instances and the OneCloud Deloitte HUB for internet connectivity. The scenario was the result of a lift and shift migration of a Deloitte on premises application. Beyond Windows AD services provided by the Deloitte HUB no connectivity to the Deloitte on premises services was required. The recipe does not target or cover a database technology nor a disaster recovery scenario.

# Recipe Architecture

The diagram below illustrates the conceptual AWS architecture of the recipe.



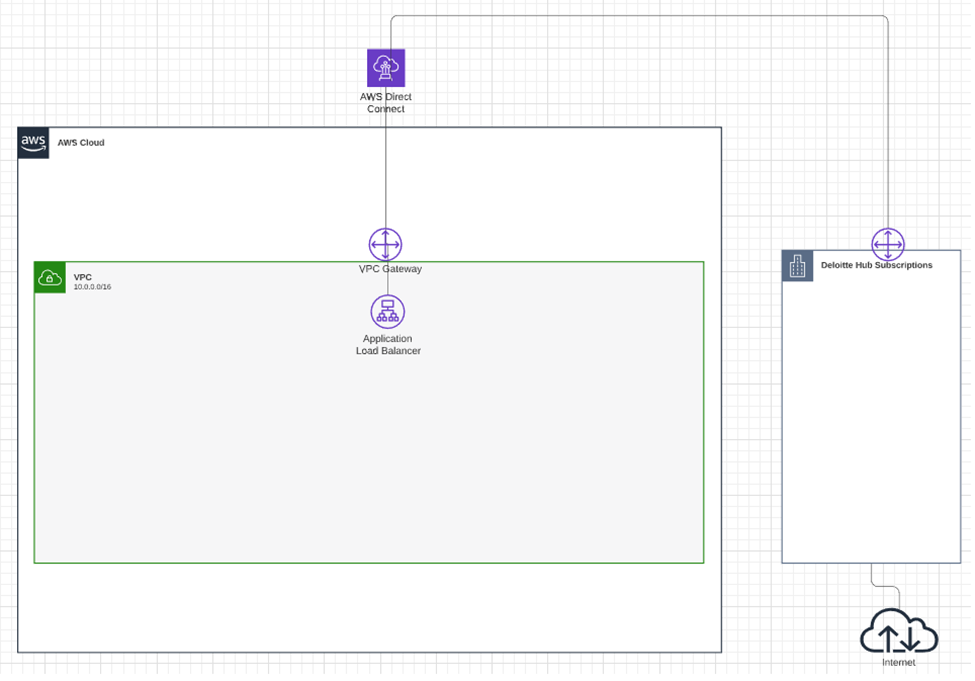
As one can see from the diagram all internet traffic ingress and egress from the AWS Infrastructure goes through a Deloitte HUB subscription to mitigate cyber security threats (Team n.d.). Traffic ingress from the internet flows from the Deloitte HUB subscription to the Deloitte target subscription by leveraging VNet peering to a direct connection with a single subnet within the target AWS subscription (the one hosting the application).  
  
The subnet mentioned above contains all of the application’s infrastructure consisting of an AWS Standard Internal Load Balancer, two IaaS EC2 Instances hosting Windows Server and IIS in an AWS Target Groups (any firm approved OS and web server could be used). It should be noted that more than 2 EC2 Instances can be used in this recipe.

# Pertinent Cloud Component Details

**Deloitte HUB**

The Deloitte HUB can be thought of as a Deloitte SaaS service that provides a web application firewall (WAF) and enforces Cyber Security controls on internet traffic. This allows Cyber Security and other entities within Deloitte to ensure the application is protected from common security threats and that malicious traffic is tracked.

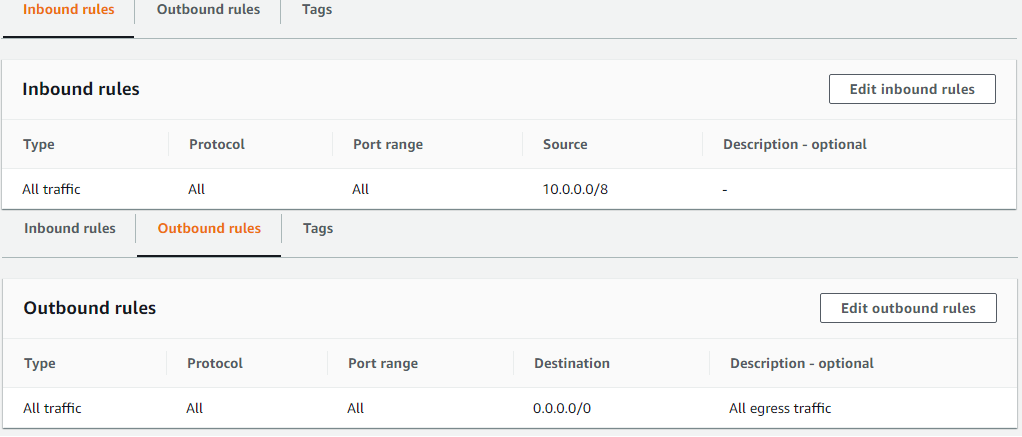
The HUB is used to communicate from the internet to the load balancer. To achieve this goal an external IP was requested from the networking team. This IP was then configured by the networking team to route all traffic from port 443 to the IP used for the AWS Application Internal Load Balancer. As you can see from the diagram below traffic from the Deloitte HUB simply enters the load balancer as if it came from an AWS internet IP associated with an AWS Application External Load Balancer.



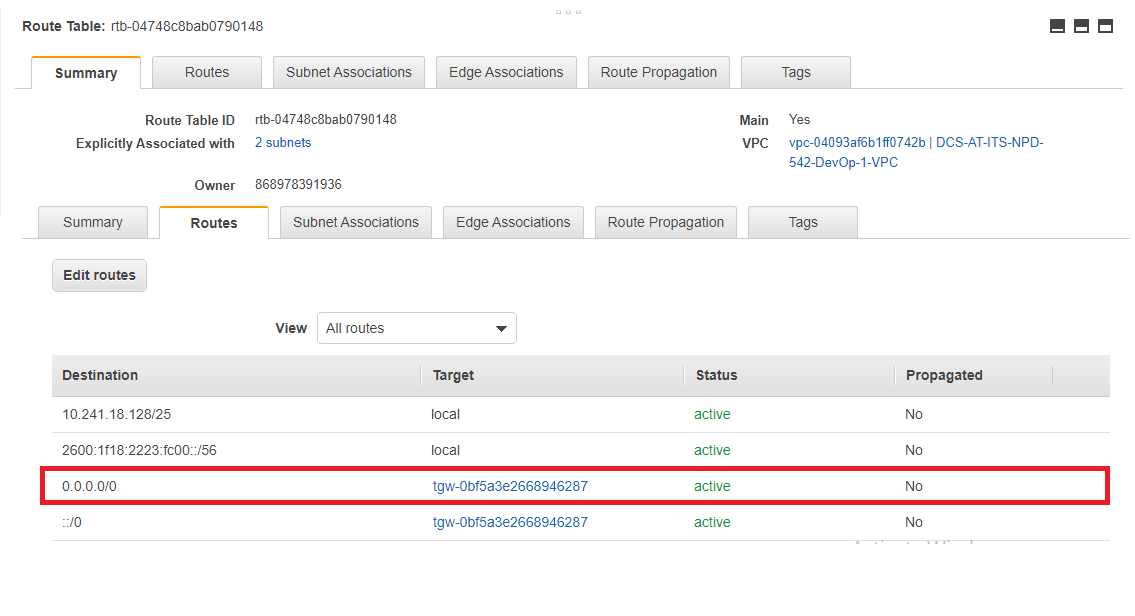
**AWS Application Internal Load Balancer**

The AWS Application Internal Load Balancer was assigned an IP address on the same subnet as the AWS IaaS EC2 Instances. The EC2 Instances where assigned to the load balancer’s target group so that traffic entering the load balancer could be distributed between them. The target group required SSL certificate for port 443 to open. One health probe was created targeting the TCP protocol and was assigned to probe port 443 and assigned to the target group with the same target port, 443. It is important to note that TCP was targeted for the probe.

The AWS Internal Application Load Balancer also required special network configuration to ensure the load balancer could talk to the EC2 Instances and the EC2 Instances to contact external web services on the internet. By default, the Application Load Balancer does not allow communications between the load balancer and EC2 Instances so a network security group was created to allow traffic to and from the load balancer and to the Deloitte HUB subscription as illustrated below.



It was additionally necessary to modify the route table to route all unhandled traffic to the hub subscription for outbound internet connectivity. This ensures all traffic is scanned by the HUB. A sample configuration is illustrated below.



The route highlighted in red sends all unhandled traffic to the Deloitte HUB. Be careful not to make assumption about the HUBs internals because the F5s or any other components could be replaced in the future. This implementation is very specific to Deloitte’s networking strategy.

**Target Group**

The Target Group was used in this recipe to attach EC2 Instances to the AWS Application Internal Load Balancer. The Target Group was attached with HTTPS Listener on Port 443 with Health Checks. All the EC2 Instances were attached to Target Group.

**Subnet**

As mentioned previously the direct connection was peered with the HUB subscription to allow for internet traffic to flow to the load balancer and out of the EC2 Instances. The route table changes in the AWS Application Internal Load Balancer section above were required.

**IaaS EC2 Instances**

The IaaS EC2 Instances in this scenario functions as a web server connecting to a backend datastore on the same subnet. HTTP Host Headers are used to host multiple sites. Port 443 is utilized for internet traffic.

Important: All EC2 Instances in the recipe were provisioned using OneCloud Cloud Script which ensures Puppet is installed on the EC2 Instances. This allows the EC2 Instances to be patched to Deloitte standards and ensure appropriate Cyber Security services are installed. You can learn more about OneCloud at <http://onecloud.deloitte.com>.

# Constraints and Limitations

This recipe is recommended for quick lift and shift migrations of on premises applications that do not have greater than a tier 3 BCP requirement. An AWS Application Internal Standard Load Balancer was chosen specifically because it is a lightweight level three load balancer that easily mimics on premises concepts.

While this recipe as it stands does not provide a high degree of business continuity adding AWS Route53 and CloudEndure will enhance it to meet most business continuity needs.

# APPENDIX A – Control Reference

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| **Amazon Elastic Load Balancer (ELB):**  Elastic Load Balancing automatically distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, and IP addresses. It can handle the varying load of the application traffic in a single Availability Zone or across multiple Availability Zones. Elastic Load Balancing offers three types of load balancers that all feature the high availability, automatic scaling, and robust security necessary to make the applications fault tolerant. | | | |
| **Control requirement** | **Control #** | **Control Description** | **Security and Control Requirements** |
| **Cyber Risk Management, Metrics & Reporting** | **IT-51** | IPS events are logged into Security Incident and Event Management (SIEM) and monitored by Security Operations Center (SOC) | • Log Intrusion Prevention System (IPS) events into a SIEM (Security Incident and Event Management) solution and integrate for monitoring with the Security Operations Center (SOC)  • Monitor IPS logs continuously for alerts and suspicious network activity |
| **Implementation** | The Recipe’s architecture ensures that all HTTP traffic from the web frontend flows through an **internal** Deloitte network IP on port 443 exposed by an internal load balancer. To be more specific the Recipe’s internal load balancer is only configured to allow inbound HTTP egress from the internal Deloitte network. Internal and external users can only access the internal HTTP endpoint if traffic is routed to the internal load balancer through additional network configurations. As an example, the network team could allow external user HTTP requests to be routed to the load balancer’s internal IP through a Deloitte HUB subscription[[1]](#footnote-1) with VPC peering. This network configuration addresses the specified IPS requirement given it inherits compliance from mechanisms configurated by the networking. The recipe does not provide any mechanism to configure a means to expose traffic directly on the internet or internal Deloitte network other than the previously mentioned endpoint on the internal load balancer. | |
| **Data Privacy** | **DS-05** | Application owners must ensure data persists in the appropriate geo according to end user or business rules | • Select appropriate region (i.e., region in which the application is hosted) while creating an Application Load Balancer |
| **Implementation** | The AWS EC2 Instances and Load Balancer deployed by the Recipe by default are placed in the same region which is specified by the User. | |
|  | **DS-07** | Member firm engagement data is restricted from transfer outside of the geo where the engagement is hosted | |
| **Implementation** | The AWS EC2 Instances and Load Balancer deployed by the Recipe by default are placed on Deloitte HUB which restricts the data transfer outside of the geo where the engagement is hosted. | |
|  | **DS-10** | Sensitive information must not be stored in log files | • Ensure any sensitive and critical information, which has been classified as confidential information (CI) based on the data classification guidelines, is not stored in application and platform related logs |
| **Implementation** | The AWS EC2 Instances and Load Balancer are deployed by the Recipe by default do not have contain sensitive information to expose. Application deployed on the EC2 Instances will be responsible for implementing this controls at the application level. | |
| **Identity Lifecycle Management** | **HI-06** | Access to systems is controlled through authentication that requires a unique user ID and password | • Integrate with Deloitte AD for authenticating user access to Application Load Balancers |
| **Implementation** | The recipe uses Deloitte AD for authenticating user access to Application Load Balancers. | |
|  | **IT-37** | Access to systems is controlled through authentication that requires a unique user ID and password. System password security settings are implemented and require a minimum password length of 10 characters and password complexity enabled for all accounts. Additionally, passwords for internal users require an initial password change and password maximum age of 90 days | • Integrate with Deloitte AD for authenticating user access to Application Load Balancers |
| **Implementation** | The recipe uses Deloitte AD for authenticating user access to Application Load Balancers. | |
| **Network Security** | **CI-13** | Native IP filtering must be enabled for PaaS services, where available. | • Configure a security group on the ELB to specify an IP address or define a range of IP addresses and ports from trusted clients |
| **Implementation** | The Application Load Balancer creates and uses security group with IP addresses and ports according to the Deloitte compliance. | |
|  | **IT-39** | Firewalls are in place to prevent unauthorized access to the corporate network. New firewall rules or changes to existing firewall rules are approved prior to being implemented | • Configure a security group on the ELB to specify an IP address or define a range of IP addresses and ports from trusted clients  • Configure a WAF rule (if applicable) to block undesirable source IPs or CIDR blocks, while monitoring and scanning the EC2 instances  • Ensure any new firewall rules or changes made to the existing firewall rules are tracked, are assigned to the appropriate personnel for approval, and a ticket is associated with every change request, per the Firewall Change Management process  ***Note****- WAF will monitor all the HTTP/HTTPS requests that are forwarded to Application Load Balancer* |
| **Implementation** | The Application Load Balancer recipe creates and uses security group with cidr blocks according to deloitte compliance. The Application Load Balancer recipe does not create new firewall rules but is dependent on the existing one. | |
| **Privileged User Access Control** | **DS-01** | Application role-based access must limit end user access to only data corresponding to engagements they are participating in by member firm. For super user access, access must be limited to only engagement / project data for the member firm that the super user belongs to | **Service used for Implementation – Amazon IAM**  • Leverage AWS Role-Based Access within the AWS IAM to enable fine-grained access management by allowing access for job-only related activities  • Do not assign multiple users to the “Owner” role as this role provides unrestricted management and data access rights  • While creating a new role or even for the existing role; update the policies attached with the roles as needed  • Enforce domain level authentication and authorization controls on all instances. Always follow principle of least privilege when assigning user roles  • Never login with root user, instead use sudo command to obtain root privileges after login  • Enforce root user login using SSH and certificate based authentication, which is setup at the time of creating the instance |
| **Implementation** | The Application Load Balancer recipe uses the existing IAM role for SSM and other authentication and authorization controls on all instances. The recipe enforces root user login using SSH and certificate based authentication, which is setup at the time of creating the instance. | |
| **Secure Software Development Lifecycle** | **IT-36**  **DS-24** | Privileged access to sensitive resources is restricted to defined user roles, and access to these roles must be approved by the respective data owner. This access is reviewed by the respective data owner on a periodic basis | **Service used for implementation – Amazon IAM**  • Leverage AWS Role-Based Access within the AWS IAM to enable fine-grained access management by allowing access for job-only related activities  • Do not assign multiple users to the “Owner” role as this role provides unrestricted management and data access rights  • Provision all privileged access, considering the following principles, per the Deloitte access provisioning standards and procedures:  1. Least-privilege - Everything is generally forbidden unless expressly permitted  2. Segregation of duties and roles, as defined by Application / System and / or Business Owner (e.g., access request, access authorization, access administration, audit)  3. For each privileged access request, ensure that the Elevated Access Form is completed with the type of access, description and approver details  4. Never login with root user, instead use sudo command to obtain root privileges after login |
| **Implementation** | The Application Load Balancer recipe uses the existing IAM role for SSM and other authentication and authorization controls on all instances. The recipe enforces root user login using SSH and certificate based authentication, which is setup at the time of creating the instance. | |
|  | **DS-27** | Cloud hosted applications with dependencies on on-premise applications should communicate via secure Internet-accessible endpoints | • Use AWS PrivateLink (VPC endpoint has IP addresses drawn from the VPC’s subnets, without the need for an Internet or NAT Gateway) to access Application Load Balancer |
| **Implementation** | The recipe uses VPC and subnets provided by deloitte Hub to access Application Load Balancer. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Elastic Cloud Compute (EC2):**  Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity inthe cloud. It is designed to make web-scale cloud computing easier for developers. Amazon EC2’s simple web service interface allows to obtain and configure capacity with minimal friction. It provides with complete control of the computing resources and lets user run on Amazon’s proven computing environment. | | | |
| **Control requirement** | **Control #** | **Control Description** | **Security and Control Requirements** |
| **Data Privacy** | **DS-05** | Application owners must ensure data persists in the appropriate geo according to end user or business rules | • Select appropriate region while creating an EC2 instance |
| **Implementation** | The AWS EC2 Instances and Load Balancer deployed by the Recipe by default are placed in the same region which is specified by the User. | |
| **DS-09** | Sensitive information must not be stored in log files | |
| **Implementation** | The AWS EC2 Instances and Load Balancer are deployed by the Recipe by default do not have contain sensitive information to expose. Application deployed on the EC2 Instances will be responsible for implementing this controls at the application level. | |
| **DS-14** | Access to systems is controlled through authentication that requires a unique user ID and password | • Allow only encrypted connections between EC2 instances and the AWS API endpoints or other sensitive remote network services. This can be enforced through the use of outbound security group rules |
|  |  | • Use Encryption protocols like Transport Layer Security (TLS), IPsec etc. |
| **Implementation** | The AWS EC2 Instances and Load Balancer are using Outbound security rules. | |

1. [↑](#footnote-ref-1)