**R&D Document: Understanding and Creating a Three-Tier Architecture**

**Table of Contents**

1. **Introduction**
2. **Overview of Three-Tier Architecture**
3. **Detailed Explanation of Each Tier**
4. **Inbound and Outbound Networking**
5. **VM Host Firewall Configuration**
6. **Best Practices**
7. **Conclusion**

**1. Introduction**

This document provides a comprehensive guide to understanding and creating a three-tier architecture, including the Web Tier, App Tier, and DB Tier. It covers inbound and outbound networking configurations and VM host firewall settings to ensure a secure and efficient deployment.

**2. Overview of Three-Tier Architecture**

Three-tier architecture is a client-server software architecture pattern that divides an application into three logical and physical computing tiers: the presentation tier (Web Tier), the application logic tier (App Tier), and the data storage tier (DB Tier). Each tier is designed to be independent of the others, which allows for better scalability, manageability, and security.

**3. Detailed Explanation of Each Tier**

**Web Tier (Presentation Layer)**

* **Function**: Handles all user interactions, user interface (UI) components, and client-side logic.
* **Components**: Web servers, load balancers, and static content delivery.
* **Technologies**: HTML, CSS, JavaScript, and web servers like Apache, Nginx, or IIS.

**App Tier (Business Logic Layer)**

* **Function**: Contains the core application logic, processing commands, making logical decisions, and performing calculations.
* **Components**: Application servers and microservices.
* **Technologies**: Java, .NET, Python, Node.js, and application servers like Tomcat, JBoss, or WebLogic.

**DB Tier (Data Layer)**

* **Function**: Manages data storage, data retrieval, and database management.
* **Components**: Database servers.
* **Technologies**: SQL databases (MySQL, PostgreSQL, SQL Server) and NoSQL databases (MongoDB, Cassandra).

**4. Inbound and Outbound Networking**

**Inbound Networking**

* **Web Tier**: Allow incoming traffic on HTTP (port 80) and HTTPS (port 443) from the internet.
* **App Tier**: Allow incoming traffic from the Web Tier on specific application ports (e.g., port 8080).
* **DB Tier**: Allow incoming traffic from the App Tier on database ports (e.g., port 3306 for MySQL).

**Outbound Networking**

* **Web Tier**: Allow outgoing traffic to the App Tier.
* **App Tier**: Allow outgoing traffic to the DB Tier and, if necessary, to external services (e.g., APIs).
* **DB Tier**: Typically, restrict outbound traffic for security reasons.

**Example NSG Configuration**

**Web Tier NSG**

sh

Copy code

# Allow HTTP and HTTPS inbound

az network nsg rule create --resource-group <ResourceGroupName> --nsg-name <WebTierNSG> --name AllowHTTP --priority 1000 --source-address-prefixes Internet --destination-port-ranges 80 --direction Inbound --access Allow --protocol Tcp

az network nsg rule create --resource-group <ResourceGroupName> --nsg-name <WebTierNSG> --name AllowHTTPS --priority 1001 --source-address-prefixes Internet --destination-port-ranges 443 --direction Inbound --access Allow --protocol Tcp

**App Tier NSG**

sh

Copy code

# Allow traffic from Web Tier

az network nsg rule create --resource-group <ResourceGroupName> --nsg-name <AppTierNSG> --name AllowWebTier --priority 1000 --source-address-prefixes <WebTierSubnet> --destination-port-ranges 8080 --direction Inbound --access Allow --protocol Tcp

**DB Tier NSG**

sh

Copy code

# Allow traffic from App Tier

az network nsg rule create --resource-group <ResourceGroupName> --nsg-name <DBTierNSG> --name AllowAppTier --priority 1000 --source-address-prefixes <AppTierSubnet> --destination-port-ranges 3306 --direction Inbound --access Allow --protocol Tcp

**5. VM Host Firewall Configuration**

**Configuring Firewall Rules**

1. **Windows VM**: Use Windows Defender Firewall with Advanced Security to create inbound and outbound rules.
   * Example: Allow inbound SQL traffic on port 1433.

sh

Copy code

New-NetFirewallRule -DisplayName "Allow SQL" -Direction Inbound -Protocol TCP -LocalPort 1433 -Action Allow

1. **Linux VM**: Use iptables or firewalld to manage firewall rules.
   * Example: Allow inbound MySQL traffic on port 3306.

sh

Copy code

sudo iptables -A INPUT -p tcp --dport 3306 -j ACCEPT

sudo service iptables save

**6. Best Practices**

* **Segmentation**: Use NSGs and subnets to segment each tier and control traffic flow.
* **Least Privilege**: Apply the principle of least privilege for all network and firewall rules.
* **Monitoring and Logging**: Enable logging for NSGs and firewalls to monitor traffic and detect anomalies.
* **Regular Audits**: Perform regular security audits and vulnerability assessments.
* **Redundancy**: Implement redundancy and load balancing for high availability.

**7. Conclusion**

Implementing a three-tier architecture with proper inbound and outbound networking configurations and VM host firewall settings can significantly enhance the security, scalability, and manageability of your application. Following best practices ensures robust protection against unauthorized access and potential threats.

**References:**

* [Azure Network Security Groups (NSG) Documentation](https://docs.microsoft.com/en-us/azure/virtual-network/security-overview)
* [Azure Application Security Groups (ASG) Documentation](https://docs.microsoft.com/en-us/azure/virtual-network/application-security-groups)
* [Azure Firewall Documentation](https://docs.microsoft.com/en-us/azure/firewall/overview)
* [Windows Firewall with Advanced Security](https://docs.microsoft.com/en-us/windows/security/threat-protection/windows-firewall/windows-firewall-with-advanced-security)
* Linux iptables Documentation