### Azure के 4 Load‑Balancers — एक नज़र में “पासा‑पलट” टेबल

(इसे interview याद‑sheet की तरह रख सकते हो)

| **Feature 🏷️** | **Azure Load Balancer (Basic/Standard)** | **Azure Application Gateway (v2/WAF)** | **Azure Traffic Manager** | **Azure Front Door (Std/Premium)** |
| --- | --- | --- | --- | --- |
| **OSI Layer / Protocol** | L4 – TCP/UDP (no HTTP logic) | L7 – pure HTTP/HTTPS | DNS‑level (name resolution) | L7 – HTTP/HTTPS only |
| **Scope** | **Regional** (एक region के अंदर) | **Regional** | **Global** DNS | **Global** edge (anycast POPs) |
| **Backend मशीनें कहाँ?** | VM/VMSS **same VNet**; single VNet per rule [learn.microsoft.com](https://learn.microsoft.com/en-us/azure/load-balancer/components?utm_source=chatgpt.com) | Any IP/FQDN reachable, लेकिन Gateway **dedicated subnet** में बैठता है [learn.microsoft.com](https://learn.microsoft.com/en-us/azure/application-gateway/configuration-infrastructure?utm_source=chatgpt.com)[learn.microsoft.com](https://learn.microsoft.com/en-us/azure/application-gateway/application-gateway-faq?utm_source=chatgpt.com) | कोई भी public‑facing endpoint (Azure, on‑prem, दूसरे cloud) [learn.microsoft.com](https://learn.microsoft.com/en-us/azure/traffic-manager/traffic-manager-overview?utm_source=chatgpt.com) | Public IP/FQDN; Private Link केवल Premium edition |
| **Same VNet/Availability‑Set condition?** | • Backend NICs same VNet  • Per availability‑set: max 1 Public + 1 Internal LB [learn.microsoft.com](https://learn.microsoft.com/en-us/azure/load-balancer/components?utm_source=chatgpt.com) | Gateway खुद dedicated subnet में; backend को same VNet होना **ज़रूरी नहीं** (public भी चलेगा) | VNet irrelevant (DNS answer देता है) | VNet irrelevant (Edge POP से public traffic) |
| **Dedicated Subnet needed?** | ❌ | ✅ (GatewaySubnet‑2) [learn.microsoft.com](https://learn.microsoft.com/en-us/azure/application-gateway/configuration-infrastructure?utm_source=chatgpt.com) | ❌ | ❌ |
| **HTTP goodies (path‑based, host‑based, SSL offload, WAF)** | कोई नहीं | ✔ Path/host; SSL off/on; Optional WAF | ❌ (DNS only) | ✔ Path/host; Rules Engine; Built‑in WAF; SSL certs auto‑managed [learn.microsoft.com](https://learn.microsoft.com/en-us/azure/frontdoor/front-door-faq?utm_source=chatgpt.com) |
| **Global/Geo routing** | ❌ (क्लासिक regional) | ❌ (regional) | ✔ Performance, Priority, Weighted, Geographic, MultiValue policies [learn.microsoft.com](https://learn.microsoft.com/en-us/azure/traffic-manager/traffic-manager-load-balancing-azure?utm_source=chatgpt.com) | ✔ Anycast POP => nearest POP; can steer by latency/health |
| **Private endpoints** | Internal LB subnet‑private | Private front‑end via VNet integration | ❌ | Premium: Private Link origins |
| **Health probing** | TCP/HTTP/HTTPS probes | HTTP/HTTPS custom probes (+active/passive) | DNS checks, probe HTTP/HTTPS | HTTP/HTTPS health probes |
| **Typical use‑case** | VM Scale Set या AKS internal traffic | Web app layer‑7 routing + WAF | Region‑level fail‑over via DNS | Global web acceleration + WAF + CDN‑style routing |
| **Free‑tier support?** | Basic LB free | Not free (v2 billed per‑capacity) | Free (traffic‑based charges) | Not free (Std/Prm) |
| **Remember** | Same VNet, layer‑4 only | Dedicated subnet, layer‑7 brain | DNS answer बदलता है, कोई TCP सेशन touch नहीं | Edge POP पर TLS offload, geo‑wide smart routing |

 **Inside one VNet VM‑to‑VM?** ➜ **Azure Load Balancer**.

 **One region, path‑based URL routing + WAF?** ➜ **Application Gateway**.

 **Cross‑region fail‑over via DNS?** ➜ **Traffic Manager**.

 **Globally distributed web front door + acceleration + WAF?** ➜ **Azure Front Door**.

### 🔐 ****Azure Backup Services – Default Retention Period Table (AZ-104 Ready)****

| **🔢 #** | **🔧 Backup Type / Service** | **💾 Data Source** | **⏳ Default Retention Period** | **📌 Notes** |
| --- | --- | --- | --- | --- |
| 1 | **Azure VM Backup** | Azure VMs (Windows/Linux) | **30 days** | - Daily snapshot - Policy customizable: up to 99 years |
| 2 | **Azure File Share Backup** | Azure Storage (Files) | **30 days** | - Daily backup - Can go up to 10 years |
| 3 | **Azure Blob Backup (Preview)** | Block Blobs | **None by default** | - You define your own backup schedule and retention |
| 4 | **Azure SQL Database Backup** | Azure SQL DB / MI | **7 days** (Default) **Up to 35 days** (Basic/Svc/Std) | - Long-term retention (LTR) possible: **up to 10 years** with LTR policies |
| 5 | **Azure Backup Server (MABS)** | On-prem VMs/Files/Apps | **5 days** (daily) **2 weeks** (weekly) – by default | - Highly customizable: daily/weekly/monthly/yearly |
| 6 | **Microsoft Azure Recovery Services (MARS Agent)** | On-prem Servers (Files/Folder) | **30 days** | - Max retention: up to **3,360 days (~9 years)** |
| 7 | **Snapshot-only (Instant Restore)** | Azure VMs | **2 days** (default) | - Part of Azure Backup - Kept in staging location |
| 8 | **Azure Backup Vault - Soft Delete** | VM backups | **14 days** (Soft delete period) | - After deletion, recovery available for 14 days |
| 9 | **Azure Kubernetes Backup (via Azure Backup)** | AKS Persistent Volumes (Preview) | **7 days** | - Retention is configurable |
| 10 | **Azure Backup for SAP HANA** | SAP DB on Azure VMs | **30 days** | - Can go up to 10 years |
| 11 | **Recovery Services Vault - Item Retention** | Multiple sources | **Retention policy-driven** | - Default backup policies define retention at item-level |

## 🔐 What is WAF in Azure Application Gateway?

**WAF** = **Web Application Firewall**  
It's a **layer-7 (HTTP/HTTPS)** security feature in **Azure Application Gateway** that **protects your web applications** from common threats and attacks.

### 🔍 Key Features of WAF on Application Gateway:

| **Feature** | **Details** |
| --- | --- |
| 🔧 **Deployment** | WAF is available on **Application Gateway v2 SKU** |
| 🧠 **Protection Ruleset** | Based on **OWASP Core Rule Set (CRS)**  ➡️ Protects against: SQL injection, XSS, CSRF, command injection, etc. |
| 🔒 **Modes** | 1. **Detection mode** – logs alerts only  2. **Prevention mode** – actively blocks malicious traffic |
| 🧾 **Custom Rules** | You can define your own match conditions (IP, headers, geo, etc.) to allow/block traffic |
| 📝 **Logging** | Logs are stored in Log Analytics, Storage, or Event Hub (Diagnostic settings) |
| 📊 **Monitoring** | Integrated with Azure Monitor; shows WAF logs, matches, and rule hits |
| 📁 **Exclusions** | Can exclude certain paths, params, headers from rules |
| 🌍 **Geo-filtering** | You can block/allow traffic by country using custom rules |
| ⏱️ **Autoscaling** | App Gateway WAF v2 supports autoscaling and zone redundancy |

### 📦 SKUs That Support WAF:

| **SKU** | **WAF Support** | **Notes** |
| --- | --- | --- |
| **Application Gateway v1** | ❌ No | WAF not supported |
| **Application Gateway v2** | ✅ Yes | Required for WAF |
| **Application Gateway WAF v2** | ✅ Yes | WAF-enabled version |

### 🔥 Common WAF Use Cases

* Protect public web apps from common attacks
* Filter traffic based on country, IP, headers
* Stop bot traffic and known bad patterns
* Compliance (OWASP, PCI-DSS, etc.)

### 💡 Exam/Interview Tips (AZ-104):

* WAF works only on **Application Gateway v2** SKU.
* **Prevention mode** is used to **actively block** suspicious traffic.
* Uses **OWASP Core Rule Set (CRS)**.
* WAF can be **integrated with Azure Front Door** too (global WAF).

### WAF **does not protect VMs directly** — only web layer (L7 traffic). 🔐 WAF (Web Application Firewall) vs 🔥 Azure Firewall

| **Feature** | **WAF (on App Gateway)** | **Azure Firewall** |
| --- | --- | --- |
| 🔍 **Layer** | **Layer 7** (HTTP/HTTPS) | **Layer 3–4–7** (network, transport, some app protocols) |
| 🧠 **Focus** | Protect **web apps** (HTTP/HTTPS) from OWASP attacks like SQLi, XSS, CSRF | Control **all outbound/inbound** traffic: any protocol (HTTP, RDP, DNS, SSH, SMB, etc.) |
| 🌐 **Scope** | Works only for **HTTP/S apps** behind App Gateway | Applies to **entire VNet or subnet** — VMs, containers, etc. |
| ⚙️ **Protocols supported** | HTTP / HTTPS | Any TCP/UDP + FQDN rules (HTTP, HTTPS, RDP, SSH, DNS, etc.) |
| 🛡️ **Use case** | Secure web apps | Network-level firewall for **non-web traffic** + **centralized** control |
| 🎯 **Granularity** | Request path, headers, cookies | Source/dest IP, port, protocol, FQDN |
| 📦 **Integration** | Tied to Application Gateway only | Works with **all network traffic** in a VNet |
| 💵 **Pricing** | App Gateway + WAF pricing | Azure Firewall (per-hour + per-GB) |

**🧠 Example Scenarios**

| **Scenario** | **Use WAF?** | **Use Azure Firewall?** |
| --- | --- | --- |
| Web app (HTTPS) exposed to internet | ✅ Yes | ❌ Maybe not |
| VMs need to browse internet (outbound control) | ❌ No | ✅ Yes |
| Block specific countries from web app | ✅ Yes (WAF custom rule) | ✅ Optional |
| Secure RDP/SSH access from only 1 IP | ❌ No | ✅ Yes |
| Stop SQL injection/XSS | ✅ Yes | ❌ No |
| Enforce FQDN-based outbound rules (e.g., only allow \*.microsoft.com) | ❌ No | ✅ Yes |

**📌 Final Verdict**

✅ **Use WAF when you want to secure your web applications (HTTP/HTTPS) against common threats.**  
✅ **Use Azure Firewall when you need to control/monitor network-level traffic (RDP, SSH, DNS, etc.) across your VNet.**

➡️ So, **WAF ≠ Azure Firewall**.  
If your architecture includes **web apps + backend VMs**, you'll often use **both together**:

csharp

CopyEdit

[User] → App Gateway (WAF) → Web App / API

↓

Azure Firewall

↓

Backend Subnet VMs

## 🎯 Example Scenarios = ✅ Outbound Traffic

| **क्या हो रहा है** | **Traffic Direction** |
| --- | --- |
| VM ne Stripe/PayPal API se payment verify kiya | ✅ Outbound |
| VM ne Azure Blob Storage se image fetch kiya | ✅ Outbound |
| VM ne weather API se data liya user ko dikhाने के लिए | ✅ Outbound |
| VM ne Linux update ke लिए internet से package download किया | ✅ Outbound |
| App ne SendGrid se email bhejna initiate किया | ✅ Outbound |

🌍 Internet

↓

[App Gateway + WAF] ←─ HTTP/S web traffic

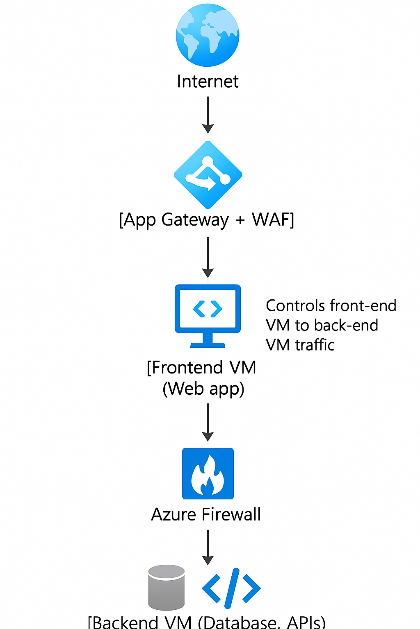
↓

[Frontend VM (Web app)]

↓

[Azure Firewall] ←─ Controls front-end VM to back-end VM traffic

↓

[Backend VM (Database, APIs)]  
  
  


### 🔷 Scenario 1 – “Classic 3‑tier with ****App Gateway + WAF**** and ****Azure Firewall****”

pgsql

CopyEdit

🌍 Internet

│ (HTTPS 443)

┌──▼────────────────────────────┐

│ App Gateway (+ WAF v2 SKU) │ 🔹SQL‑i / XSS inspection

└──┬────────────────────────────┘

│ (HTTP → Backend Pool‑A)

┌──▼──────────────┐

│ Front‑end VMSS │ 🔸 NSG‑FE (allow 80/443 from AgW)

└──┬──────────────┘

│ (Port 8080 internal)

┌──▼────────────────────────────┐

│ 🔥 Azure Firewall │ 🔹L3/L4 rules, FQDN filtering

└──┬────────────────────────────┘

│ (1433, 5671…)

┌──▼──────────────┐

│ Back‑end VMSS │ 🔸 NSG‑BE (allow only from FW ASG‑FE)

└─────────────────┘

**🔷 Scenario 2 – Same as #1 plus Azure Bastion for management**

scss

CopyEdit

(RDP/SSH via browser)

🌍 Admin PC────────────▶🌐

│

┌───────▼────────┐

│ Azure Bastion │ placed in Mgmt subnet

└───────┬────────┘

│ (RDP 3389 / SSH 22 \*within\* VNet)

<— (same flow as Scenario 1 below) —>

**Why:** No public IPs on VMs; admins connect over Bastion. Add **NSG‑Mgmt** to allow TCP 443 from corporate IPs only.

**🔷 Scenario 3 – App Gateway + WAF only (no Firewall) – small stateless site**

sql

CopyEdit

🌍 Internet

│

┌──▼─────────────────┐

│ App Gateway + WAF │

└──┬─────────────────┘

│

┌──▼─────────┐

│ Web VMSS │ 🔸 NSG‑Web (allow 80/443 only from AgW)

└────────────┘

**Use when:** Pure web traffic, no complex back‑end or outbound restrictions. Cheaper & simpler than adding Firewall.

**🔷 Scenario 4 – Azure Firewall DNAT (no App Gateway) – lift‑and‑shift VM farm**

pgsql

CopyEdit

🌍 Internet

│ Public‑IP:443 ──► DNAT → 10.0.1.4:443

┌──▼──────────────────┐

│ 🔥 Azure Firewall │

└──┬───────────────────┘

│

┌──▼──────────┐

│ Web VM NIC │ 🔸 NSG (allow only from Firewall)

└─────────────┘

**When to use:** Mixed protocols (HTTPS, RDP, SFTP) and you only need L3/L4 control; no WAF logic required.

**🔷 Scenario 5 – Global web: Azure Front Door + WAF, regional App Gateway, internal Firewall**

scss

CopyEdit

🌍 Internet

│

┌──▼─────────────────────────────┐

│ Azure Front Door (+ WAF) │ Anycast POPs, TLS offload

└──┬─────────────────────────────┘

│ (Origin = regional AgW VIP)

┌──▼─────────────────────────────┐

│ App Gateway (in region) │ Path‑based → micro‑services

└──┬─────────────────────────────┘

│

┌──▼─────────────────────────────┐

│ 🔥 Azure Firewall │ Outbound + east‑west

└──┬─────────────────────────────┘

│

┌──▼────────┐ ┌──▼────────┐

│ API VMSS │ │ DB PaaS │ 🔸 ASG‑API, ASG‑DB for policy

└───────────┘ └───────────┘

**Why:** Need global acceleration + centralized WAF (Front Door), regional app‑layer routing (AgW), and full network security (Firewall).

**🔑 Where NSG & ASG fit**

| **Layer** | **Typical NSG / ASG Rule** |
| --- | --- |
| **NSG‑Internet‑Subnet** | Deny all inbound (only FDoor/AgW public IPs allowed) |
| **NSG‑FE (front‑end)** | Allow 80/443 from AgW subnet only; Deny all else |
| **NSG‑BE (back‑end)** | Allow SQL/from **ASG‑FE**; Deny internet |
| **ASG‑FE / ASG‑API / ASG‑DB** | Logical grouping so Firewall/NSG rules stay readable |

**DNS Zone in Azure**

### Zone ke andar ye records hote hain:

| **Record Type** | **Kya kaam karta hai** | **Example** |
| --- | --- | --- |
| A Record | Naam → IP deta hai | www.brarsite.com → 52.168.1.1 |
| CNAME | Naam → Naam | api.brarsite.com → backend.azurewebsites.net |
| TXT | Info ya verify | Email sender verify, SPF, DKIM etc. |
| MX | Mail exchange | Mail server define karta hai |
| NS | Name servers | Ye batata hai ki zone ka master kaun hai |
| SOA | Start of Authority | Zone ka boss record |
| PTR | Reverse lookup | IP se naam dhoondhna |

## **7. Ye sab record kya hote hain (simple examples):**

| **Record Type** | **Matlab kya hai** | **Example** |
| --- | --- | --- |
| **A** | Naam → IP | www.brarsite.com → 20.1.1.4 |
| **CNAME** | Naam → Dusra naam | api.brarsite.com → xyz.azurewebsites.net |
| **NS** | Name server address | ns1-01.azure-dns.com |
| **TXT** | Text info (verify domain, SPF, etc.) | "v=spf1 include:\_spf.google.com" |
| **MX** | Mail server specify karta hai | 10 mail.brarsite.com |
| **PTR** | IP → Name (reverse lookup) | 20.1.1.4 → myhost.brarsite.com |

| **DNS Concept** | **Real-life Analogy** |
| --- | --- |
| Domain Name | Aadmi ka naam (Google.com) |
| IP Address | Aadmi ka phone number |
| DNS Zone | Ek diary jisme naam-number likhe hain |
| DNS Server | Ek teacher ya operator jo naam ka number batata hai |
| A record | Naam ka asli number |
| CNAME | Naam ka nickname |
| TXT | Notes / Verification |
| NS Record | Diary kahan rakhi hai ye info |
| Public DNS | Sabko dikhne wali diary |
| Private DNS | Sirf apni team ki diary |

| **Feature** | **Azure VM (IaaS)** | **App Service (PaaS)** | **AKS (CaaS)** |
| --- | --- | --- | --- |
| Deployment Model | Infrastructure as a Service | Platform as a Service | Container as a Service |
| Control over OS/Software | ✅ Full | ❌ Limited | 🔄 Moderate (base node OS) |
| Suitable for | Legacy apps, custom software | Web apps, REST APIs | Microservices, CI/CD |
| Scaling | Manual or script-based | Auto scaling supported | Auto scaling with config |
| Learning Curve | Low | Very Low | High (Kubernetes knowledge) |
| DevOps Integration | Moderate | Easy (GitHub, DevOps) | Strong with Helm + CI/CD |
| CI/CD + Containers | Possible but manual | Supported (basic containers) | Built for containers |
| Cost | High (Pay per minute) | Medium (Optimized pricing) | Medium-High (but cost efficient) |
| Example | Windows legacy app | Shopping site, portfolio site | Netflix-type architecture |