Application-load-balancer

- operate at layer 7 of OSI model, unless azure load-balancer operates at Later 4 (TCP and UDP) of OSI model, which basically route traffic based on source and destination IP-address and ports.
- ! It is an Web traffic load balancer
- An example scenario can be a user requesting images and videos from a url *contoso.com*. The images are stored in a server, and these servers can be in an *imageServerPool* with multiple servers which handles the traffic request for images. same goes to videos *VideoServerPool*.
- The Application load balancer routes to a specific pool as per the request of the user for images and videos.
- The urls change as per the request for images or videos, example: */images/* *or* /videos/**

Features

1. Secure Sockets Layer (SSL/TLS) termination
2. Autoscaling
3. Zone redundancy
4. Static VIP
5. Web Application Firewall
6. Ingress Controller for AKS
7. URL-based routing
8. Multiple-site hosting
9. Redirection
10. Session affinity
11. Web-socket and HTTP/2 traffic
12. Connection draining
13. Custom error pages
14. Rewrite HTTP headers and URL
15. Sizing

Creating an Application gateway

Tier

- Load-balancing facility
  - Standard
  - Standard V2
    - scalability
    - availability zone

- Load-balancing + firewall
  - WAF
  - WAF V2
    - scalability
    - availability zone

Firewall mode

- detection
  - Any suspicious activity, alert me

- Prevention
  - Detect and prevent the suspicious activity.

#HTTP2

- Second version of HTTP.
- The App gateway will translate HTTP2 -> HTTP internally.

Routing Rules

Listener

- The Listener “listens” on a specific port and IP address for traffic that uses a specified protocol. If the listener criteria are not met, the App gateway will apply the routing rule.
  - Frontend IP
  - Protocol
    - HTTP
    - HTTPS
  - port

- Additional settings
  - Listener type
    - Basic
    - Multi site
  - Error page url
    - Yes
    - No

Backend targets

- Target type
  - Backend-pool
  - Redirection
- Backend target
- HTTP setting
  - `#cookie_based_affinity`
    - similar to session persistence.
  - `#Connection_draining`
    - Used in case of any maintenance of a server, the connection-draining will temporarily (in seconds) re-direct the traffic from load-balancer to other servers in the backend pools.
    - also a buffer time for already entered requests to the maintenance server to process the request.
    - remaining will be forcefully terminated.