**HA Architecture**

**Elastic Load Balancers**

1. Application Load Balancer
   1. HTTP/HTTPS
   2. Layer 7 – Intelligent – Advanced Routing
   3. Target groups
2. Network Load Balancer
   1. TCP
   2. Layer 4 – Performance
   3. Target groups
   4. Static IP supported
3. Classic Load Balancer
   1. HTTP/HTTPS
   2. Not application aware
   3. Support sticky sessions and X-Forwarded-for header
4. 504 Error – Gateway time out => Indicating that the application is not responding, not a problem at the Load Balancer
5. X-forwarded-for => The head for the IP of the original requester

**Load Balancer Health Checks – Demo**

Graphical user interface, table

Description automatically generated with medium confidence­­

**Advanced Load Balancer Theory**

1. Sticky sessions -> ALB sticky to Target Group
2. Cross Zone Load Balancing
3. Path Patterns

**Auto Scaling**

1. Auto Scaling Group
2. Configuration Templates – Launch configuration
3. Scaling Options
   1. Maintain current instance levels
   2. Scale manually
   3. Scale on schedule
   4. Scale on demand
   5. Predictive scaling – Proactive / Reactive

**Launch Configuration and Autoscaling Group – Demo**

1. You can choose the VPC and the subnets to create the instances
2. You have the option to specify the Min/Max instance numbers

Graphical user interface, text, application, email

Description automatically generated

­**HA Architecture**

1. Prepared for Failures
2. Use Multi-AZ and Multi-Region whenever possible
3. Multi-AZ and Read-replica in RDS
4. Scaling out / Scaling up
5. Cost effective
6. S3 storage classes

**Elastic Beanstalk**

Elastic Beanstalk is a service that you can use to quickly deploy and manage applications in the AWS without worrying about the infrastructure that runs the applications. You simply upload your application, and Elastic Beanstalk automatically handles the details of capacity provisioning, load balancing, scaling, and application health monitoring.

**CloudFormation**

1. Much more powerful than Elastic Beanstalk but more difficult to use
2. Either YAML or JSON
3. Only required section is the “**Resources**”. All other 8 sections optional
4. Resources in the template file is named logical resources, and the ones created in AWS are called physical resources.

­­Graphical user interface, text, application

Description automatically generated

In CloudFormation => When modify a stack if “Replacement = True” can be destructive

**High Availability Bastion Hosts**

1. 2 hosts in 2 AZs. Use a NLB with static IP and health checks to fail over from one host to the other - You cannot use ALB, as it is layer 7
2. 1 host in 1 AZ behind an Auto Scaling Group with health checks and a fixed EIP (Elastic IP). You use user data (script) to associate the EIP to the new host – Cheaper option

**On-premises Strategies with AWS**

1. DMS – Database Migration Service
2. SMS – Server Migration Service
3. AWS Application Discovery Service
4. VM Import / Export
5. Download Amazon Linux 2 as ISO

**HA Architecture Summary**