Monitoring

Smart Internz

Monitoring gives insights into your applications that help you detect, investigate, and remediate problems faster.

- **Monitoring** tracks resource health and usage through data collection.
- It provides **real-time insights** to identify operational issues.
- **Metrics** help address performance, availability, and capacity concerns.


Amzron SB Buckets
Latency
ECć Intences
s3
ec2
RDS Instances
Tnttugpht
Datsbaees

- **Amazon S3 Metrics:**
- Size of objects in a bucket.
- Number of objects in a bucket.
- Number of HTTP requests to a bucket.

- **Amazon RDS Metrics:**
- Database connections.
- CPU utilization.
- Disk space usage.

- **Amazon EC2 Metrics:**
- CPU utilization.
- Network utilization.
- Disk performance.
- Status checks.

Benefits

- Detect and resolve issues proactively.
- Enhance resource performance and reliability.
- Identify and address security threats.
- Optimize operations with data-driven insights.
- Reduce costs through effective monitoring.

Smart Internz

Amazon CloudWatch

Amazon CloudWatch is a monitoring and observability service that collects your resource data and provides actionable insights into your applications.

You can use CloudWatch to do the following:

- Detect anomalous behavior in your environments.
- Set alarms to alert you when something is not right.
- Visualize logs and metrics with the AWS Management Console.
- Take automated actions like scaling.
- Troubleshoot issues.
- Discover insights to keep your applications healthy.

How CloudWatch Works

- **Managed Service:** Centralized monitoring without managing infrastructure.
- **Basic Monitoring:** Free, 1 data point per 5 minutes; suitable for most apps.
- **Detailed Monitoring:** 1-minute granularity at additional cost.
- **Metrics:** Time-ordered data points representing variables (e.g., CPU usage).
- **Dimensions:** Name-value pairs for filtering metrics (e.g., InstanceId).
- **Custom Metrics**: Record application-level data (e.g., page views, error rates).
- **High-Resolution Metrics:** 1-second granularity for precise tracking.

CloudWatch Dashboards

- **Customizable Views:** Visualize multiple metrics in widgets (e.g., graphs, text).
- **Global Insights:** Combine metrics from different AWS Regions.
- **Live Data:** Display recent data published within the last minute.
- **External Tools:** Use APIs to ingest metrics into third-party tools.
- **Access Control:** Use IAM policies to manage dashboard permissions.

Amazon CloudWatch Logs

- **Centralized Storage**: Store and analyze logs from AWS and on-premises apps.
- **Query & Filter Logs**: Search for specific data (e.g., error stack traces).
- **Metric Filters**: Convert log data into metrics for monitoring.
- **Log Agents**: Send data from EC2 instances using the CloudWatch Logs agent.
- **Source Examples**: Lambda logs (minimal setup), EC2 logs (requires agent).

CloudWatch Logs Terminology

- Log Events: Individual data entries in log files.
- Log Streams: Sequence of log events from a specific source (e.g., EC2 instance).
- Log Groups: Collection of log streams sharing the same retention policies.

CloudWatch Alarms

- **Triggering Actions:** Automatically respond to metric threshold breaches.
- States:
- **OK:** Metric within threshold; normal operation.
- **ALARM:** Metric exceeds threshold; potential issue.
- **INSUFFICIENT_DATA:** Insufficient data to determine state.

Steps to Create Alarms:

- Set up a metric filter (e.g., HTTP 500 errors).
- Define an alarm (e.g., threshold for 500 errors over 5 minutes).
- Define an action (e.g., send email or auto-remediate).
- Integrated Responses: Trigger EC2 actions, scaling, or Lambda functions.

Prevent & Troubleshoot Issues

- Notification Example: Alert for HTTP 500 errors via Amazon SNS.
- Automated Actions: Auto-reboot EC2, scale resources, or invoke Lambda functions.
- Faster Remediation: Use alarms to preemptively address operational problems.

Smart Internz

Solution Optimization

Design your system to have no single point of failure by using automated monitoring, failure detection, and failover mechanisms.

Understanding Availability

Availability Metrics:

- Expressed in **percentages** or "nines" (e.g., 99.9% = 8.77 hrs/year downtime).

Why It Matters:

- High availability ensures uninterrupted customer access to applications.

Current Issue:

- Single EC2 instance = Single point of failure.

Solution:

- Deploy a second EC2 in a different Availability Zone for redundancy.

Challenges:

- Replication, customer redirection, and choosing availability type (active-passive vs. active-active).

High Availability with Multiple Servers

Benefits:

- Reduces risk from hardware or data center failures.

Customer Redirection:

- Use DNS or Load Balancer for smooth traffic routing.

Active-Passive Systems:

- Only one server active; suitable for stateful apps.

Active-Active Systems:

- Both servers active; ideal for stateless apps.

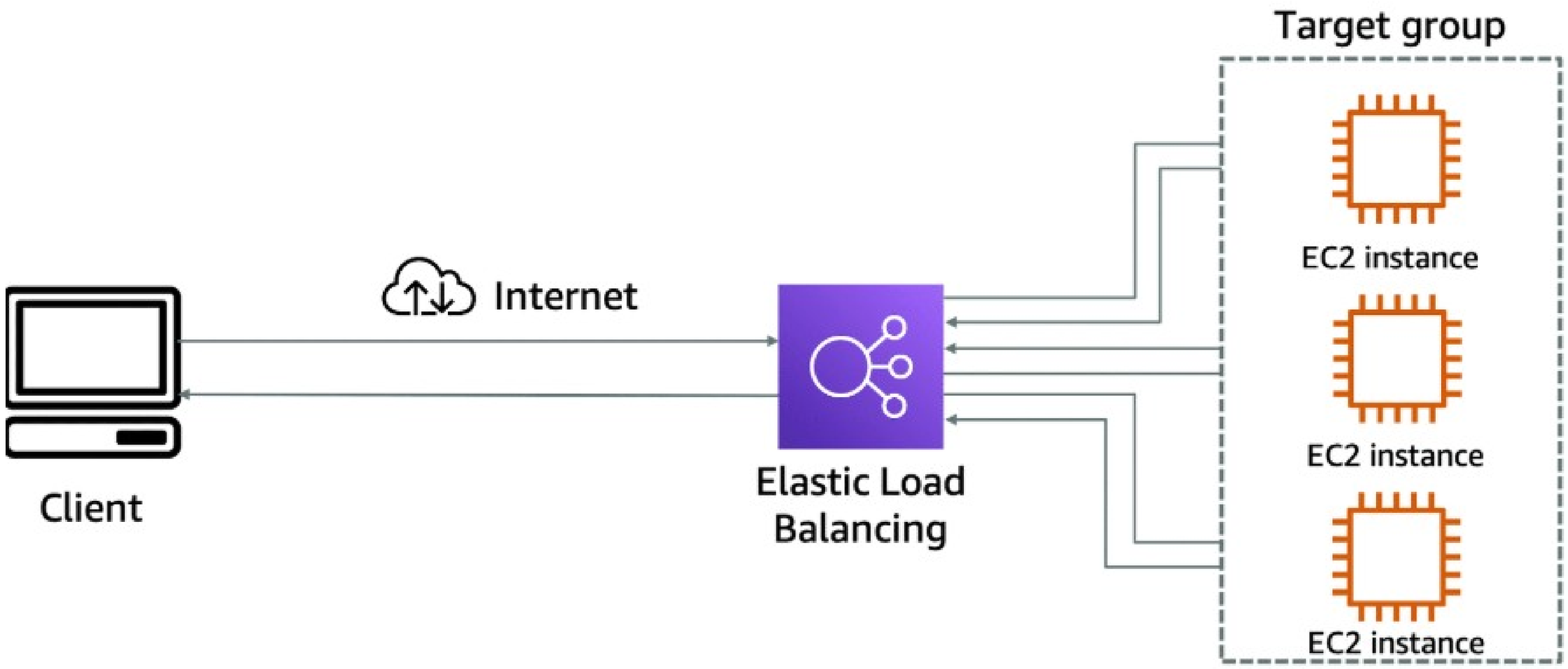
Cost vs. Availability:

- Higher availability means more infrastructure costs.

Smart Internz

Traffic Routing with Elastic Load Balancing

The Elastic Load Balancing (ELB) service can distribute incoming application traffic across EC2 instances, containers, IP addresses, and Lambda functions.


Target group
Internet
Client
Elastic Load Balancing
EC2 instance
EC2 instance
EC2 instance

Health Checks in Load Balancers (ELB)

Health Check Types:

- TCP: Verifies connection to EC2 instance.
- HTTP/HTTPS: Checks response from specified URLs (e.g., /monitor).

Purpose:

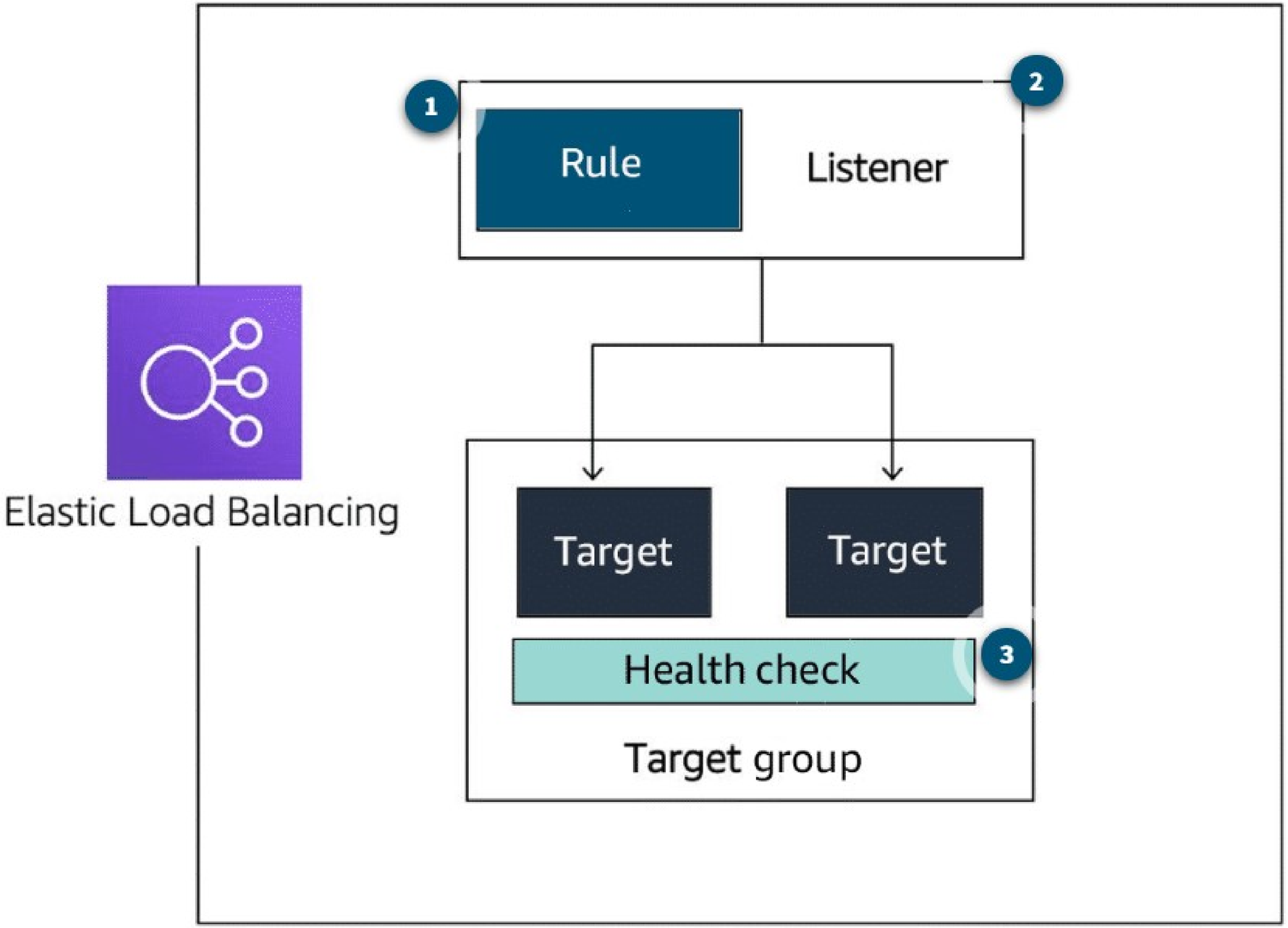
- Routes traffic only to healthy EC2 instances.

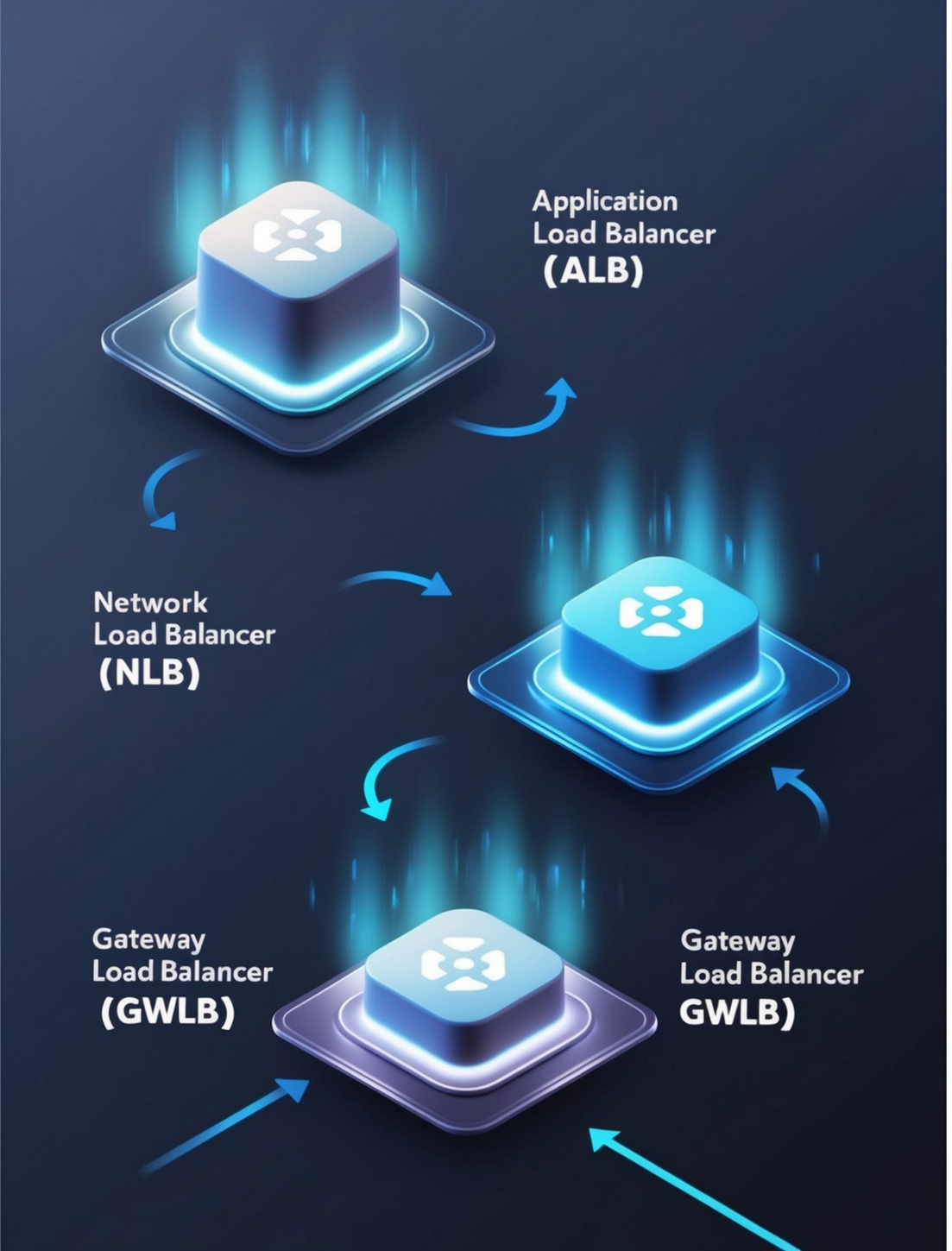
Auto Scaling Integration:

- ELB informs Auto Scaling to replace unhealthy instances.

Connection Draining:

- Prevents termination of instances with active connections.


1
2
Rule
Listener
Elastic Load Balancing
Target
Target
Health check
3
Target group



Smart Internz

Application Load Balancer (ALB)

- User authorization
- Rich metrics & logging
- Redirects
- Fixed response

Network Load Balancer (NLB)

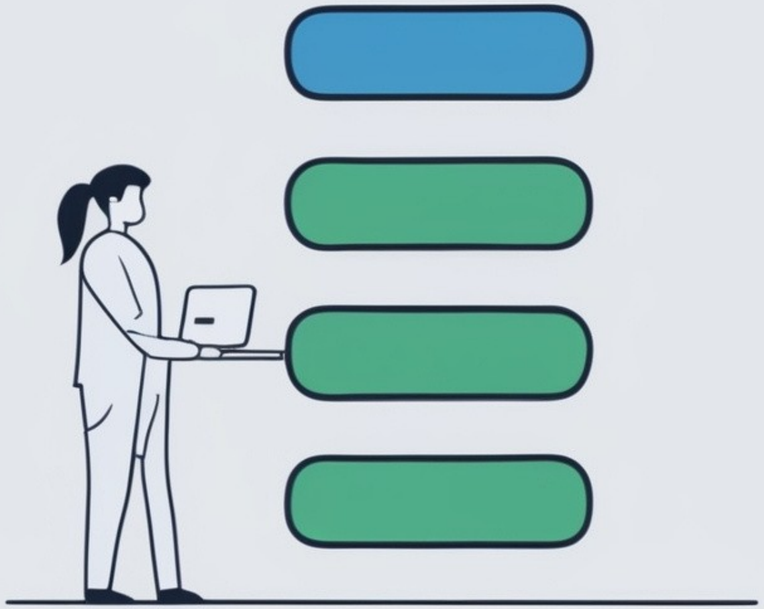
- TCP & UDP connection-based
- Source IP preservation
- Low latency

Gateway Load Balancer (GLB)

- Health checks
- Gateway Load Balancer Endpoints
- Higher availability for third-party virtual appliances

AUTO-SCALIING

Automic-Scaling



Dynamic – real-time adjustments

Automated resource allocation

VS.

TRADITIONAL-SCALIING

Tradlitional Scalling

Manual processes – slower times

Static resource allocation

Auto Scaling vs. Traditional Scaling

- **Traditional Scaling**
- Provision servers to handle peak traffic, leading to underutilized resources during low traffic periods.
- **Auto Scaling**
- Automatically adjusts capacity based on demand, ensuring cost-efficiency and maintaining performance.


AWS Trusted Advisor's Dashboard

What is AWS Trusted Advisor?

A web service that inspects your AWS environment and provides real-time recommendations based on AWS best practices.

Key Categories:

- Cost Optimization – Reducing AWS costs and improving resource usage.
- Performance – Improving the performance of AWS resources.
- Security – Identifying and addressing security vulnerabilities.
- Fault Tolerance – Enhancing availability and reliability.
- Service Limits – Alerts for approaching service usage limits.
- Dashboard Indicators:
- Green Check: No issues detected.
- Orange Triangle: Investigation recommended.
- Red Circle: Action required.

Cost Optimization

0 9 0

$7,516.85

Potential monthly savings

Performance

3 7 0

Security

2 4 11

Fault Tolerance

0 15 5

Service Limits

37 0 1

Cloud Trail

What is AWS CloudTrail?

- Records API calls in your AWS account, capturing detailed information such as the identity of the API caller, the time of the API call, and the source IP address.
- Functions as a “breadcrumb trail” of logs, allowing you to track and review all activities within your AWS resources.

Key Features:

- Event History: Provides a complete history of user activities and API calls.
- Real-time Updates: Events are updated within 15 minutes after an API call.
- Filters: Filter events by time, date, user, resource type, etc., for better insights.

Example:
If a new IAM user "Mary" is created but the owner doesn't know who made the change, CloudTrail will provide details such as:

Who: IAM user "John"
When: January 1, 2020, at 9:00 AM
How: Created through the AWS Management Console.

CloudTrail Insights:
Automatically detects unusual API activities.
Example: If there’s an unusual spike in Amazon EC2 instance launches, CloudTrail Insights will highlight this and allow further investigation.