ENGENHARIA DE SOFTWARE

41492-ES

Nuno Sá Couto / Rafael Direito

(nuno.sacouto@ua.pt / rafael.neves.direito@ua.pt)

Department of Electronics, Telecommunications and Informatics (DETI)

UNIVERSITY OF AVEIRO (UA), PORTUGAL

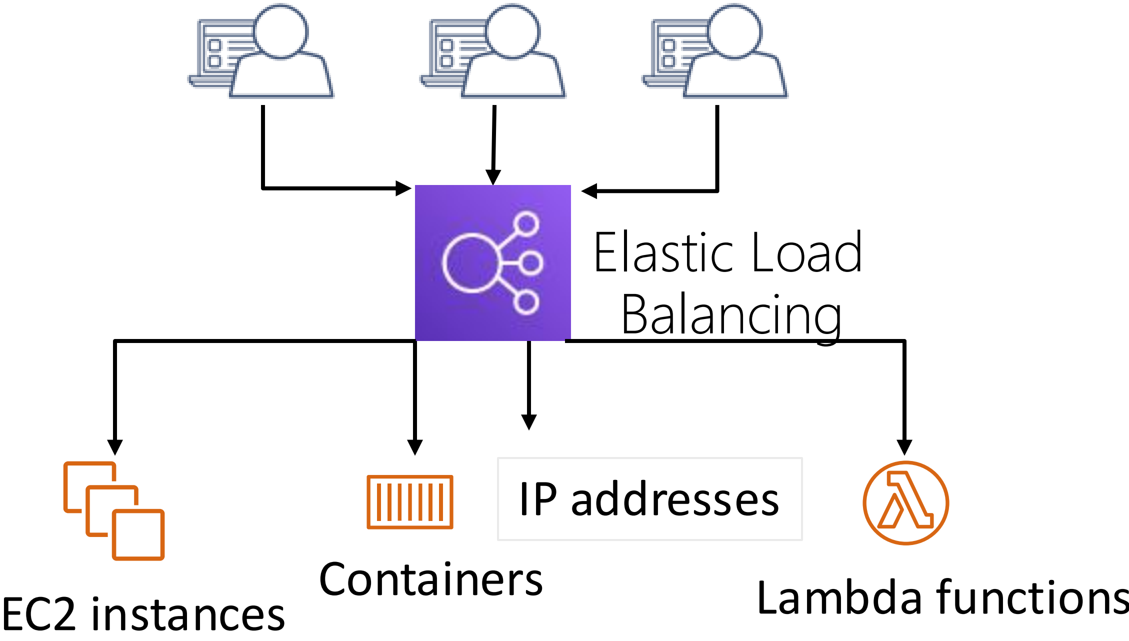
2024

Module 10: Automatic Scaling and Monitoring

SECTION 1: ELASTIC LOAD BALANCING

Elastic Load Balancing

- Distributes incoming application or network traffic across multiple targets in a single Availability Zone or across multiple Availability Zones.
- Scales your load balancer as traffic to your application changes over time.

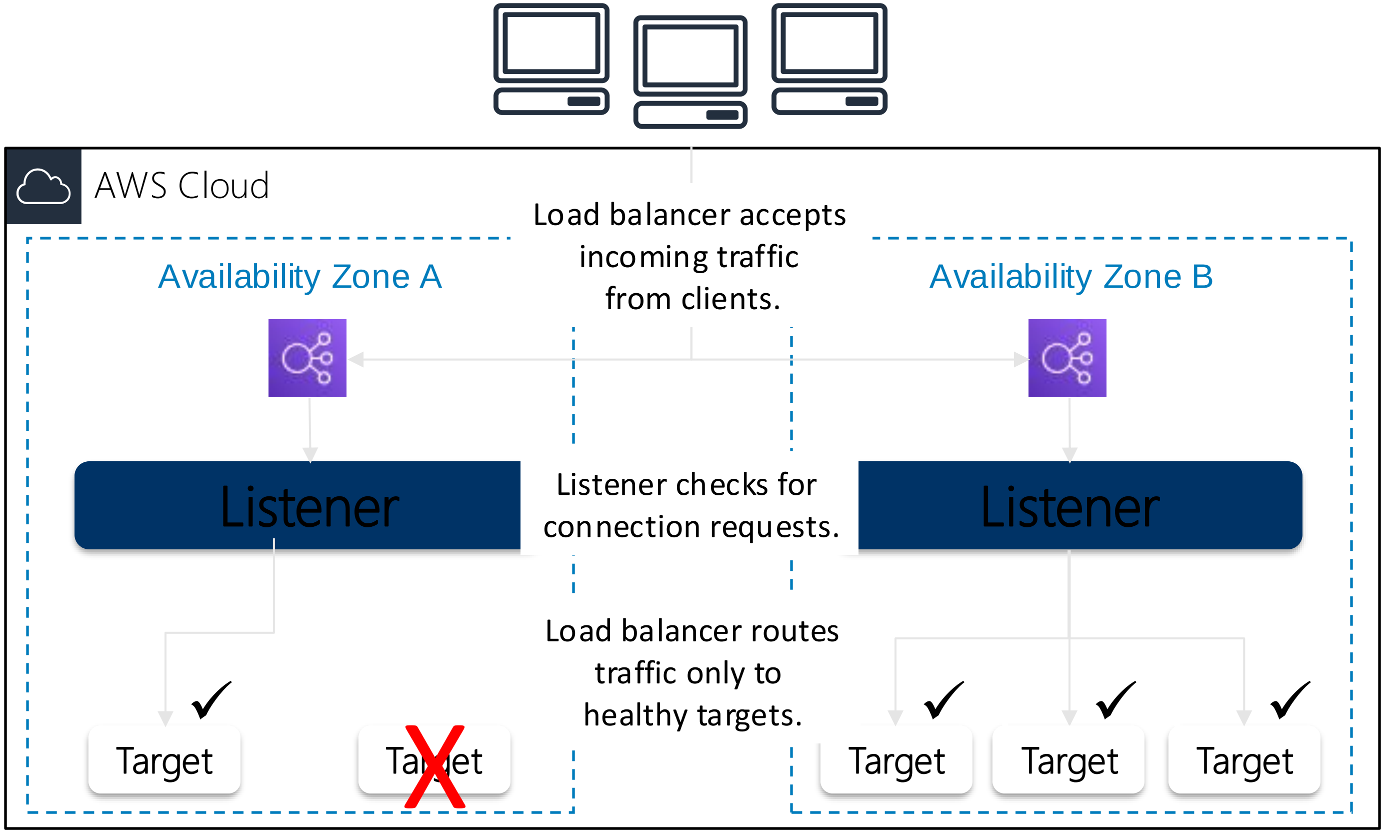


Types of load balancers

Application Load Balancer	Network Load Balancer	Classic Load Balancer (Previous Generation)
• Load balancing of HTTP and HTTPS traffic	• Load balancing of TCP, UDP, and TLS traffic where extreme performance is required	• Load balancing of HTTP, HTTPS, TCP, and SSL traffic
• Routes traffic to targets based on content of request • Provides advanced request routing targeted at the delivery of modern application architectures, including microservices and containers	• Routes traffic to targets based on IP protocol data • Can handle millions of requests per second while maintaining ultra-low latencies • Is optimized to handle sudden and volatile traffic patterns	• Load balancing across multiple EC2 instances
• Operates at the application layer (OSI model layer 7)	• Operates at the transport layer (OSI model layer 4)	• Operates at both the application and transport layers.

How Elastic Load Balancing works

- With Application Load Balancers and Network Load Balancers, you register targets in target groups, and route traffic to the target groups.
- With Classic Load Balancers, you register instances with the load balancer.



Load balancer accepts incoming traffic from clients.

Listener checks for connection requests.

Load balancer routes traffic only to healthy targets.

Load balancer performs health checks to monitor health of registered targets.

Elastic Load Balancing use cases

Highly available and fault-tolerant applications

Containerized applications

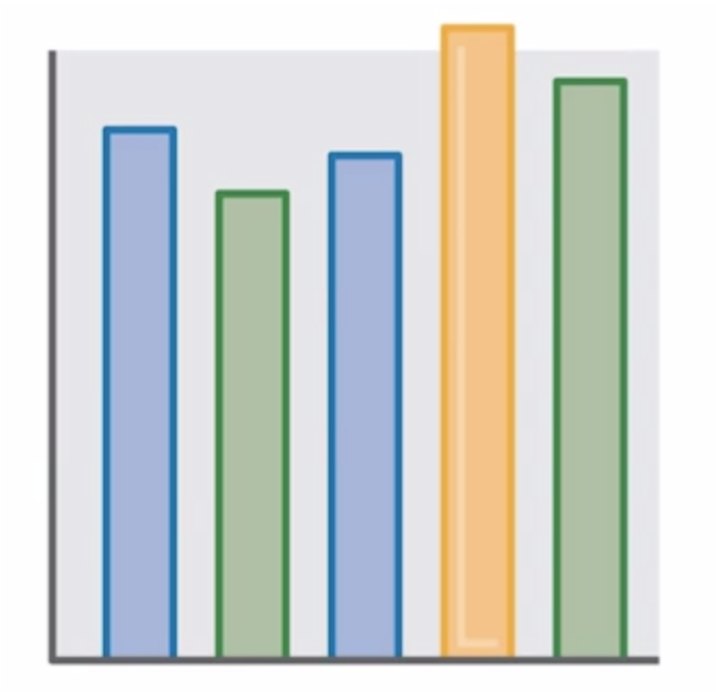
Elasticity and scalability

Virtual private cloud (VPC)

Hybrid environments

Invoke Lambda functions over HTTP(S)

Load balancer monitoring



- Amazon CloudWatch metrics – Used to verify that the system is performing as expected and creates an alarm to initiate an action if a metric goes outside an acceptable range.
- Access logs – Capture detailed information about requests sent to your load balancer.
- AWS CloudTrail logs – Capture the who, what, when, and where of API interactions in AWS services.

Module 10: Automatic Scaling and Monitoring

SECTION 2: AMAZON CLOUDWATCH

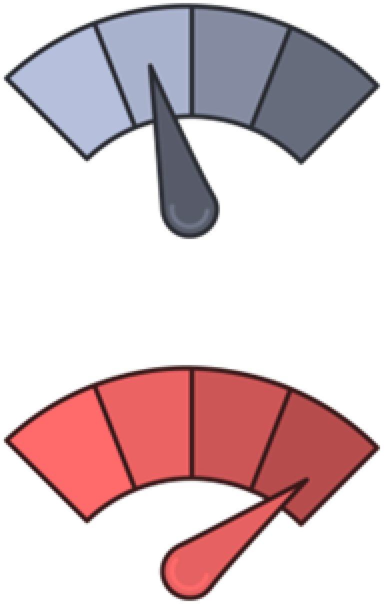
Monitoring AWS resources

To use AWS efficiently, you need insight into your AWS resources:

- How do you know when you should launch more Amazon EC2 instances?
- Is your application's performance or availability being affected by a lack of sufficient capacity?
- How much of your infrastructure is actually being used?

Amazon CloudWatch

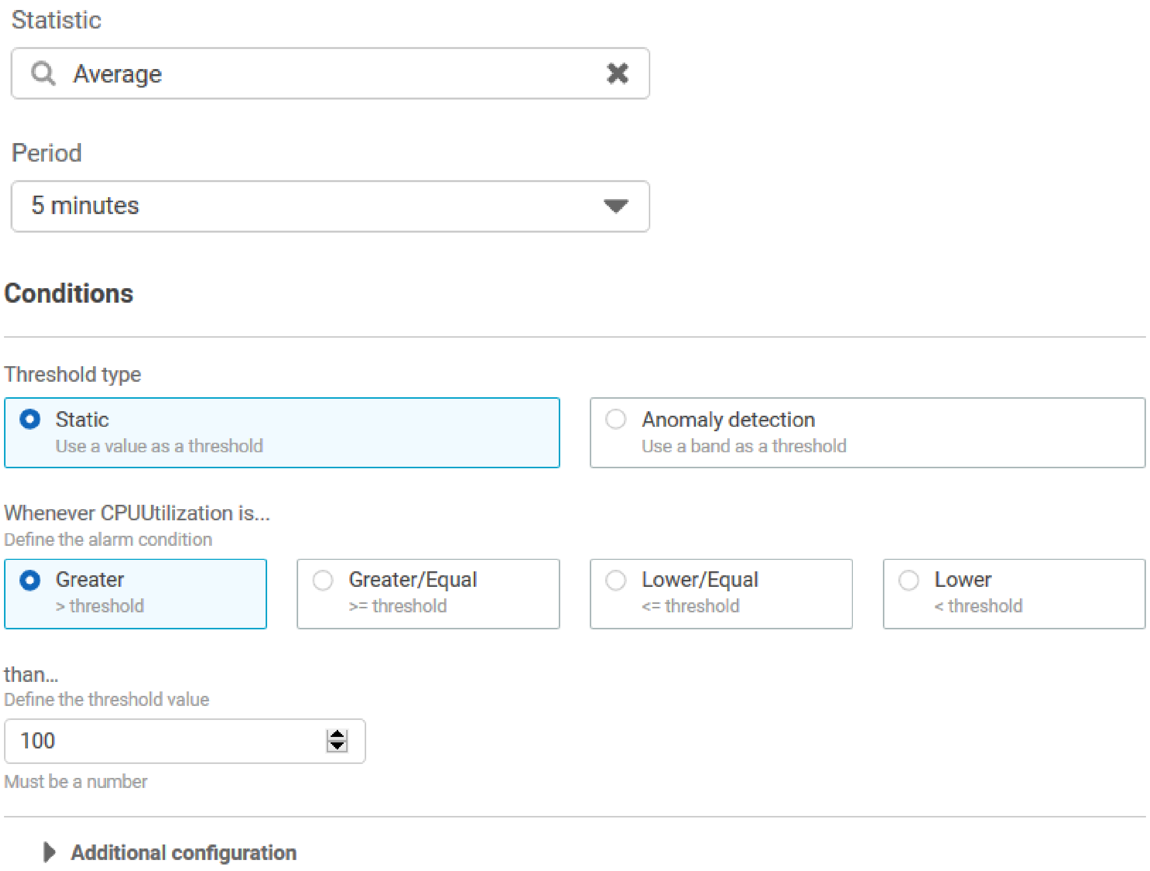
Amazon CloudWatch



- Monitors –
  - AWS resources
  - Applications that run on AWS
- Collects and tracks –
  - Standard metrics
  - Custom metrics
- Alarms –
  - Send notifications to an Amazon SNS topic
  - Perform Amazon EC2 Auto Scaling or Amazon EC2 actions
- Events –
  - Define rules to match changes in AWS environment and route these events to one or more target functions or streams for processing

CloudWatch alarms

- Create alarms based on –
  - Static threshold
  - Anomaly detection
  - Metric math expression
- Specify –
  - Namespace
  - Metric
  - Statistic
  - Period
  - Conditions
  - Additional configuration
  - Actions



Statistic

Average

Period

5 minutes

Conditions

Threshold type

● Static
Use a value as a threshold

○ Anomaly detection
Use a band as a threshold

Whenever CPUUtilization is...
Define the alarm condition

● Greater
> threshold

○ Greater/Equal
>= threshold

○ Lower/Equal
<= threshold

○ Lower
< threshold

than...
Define the threshold value

100

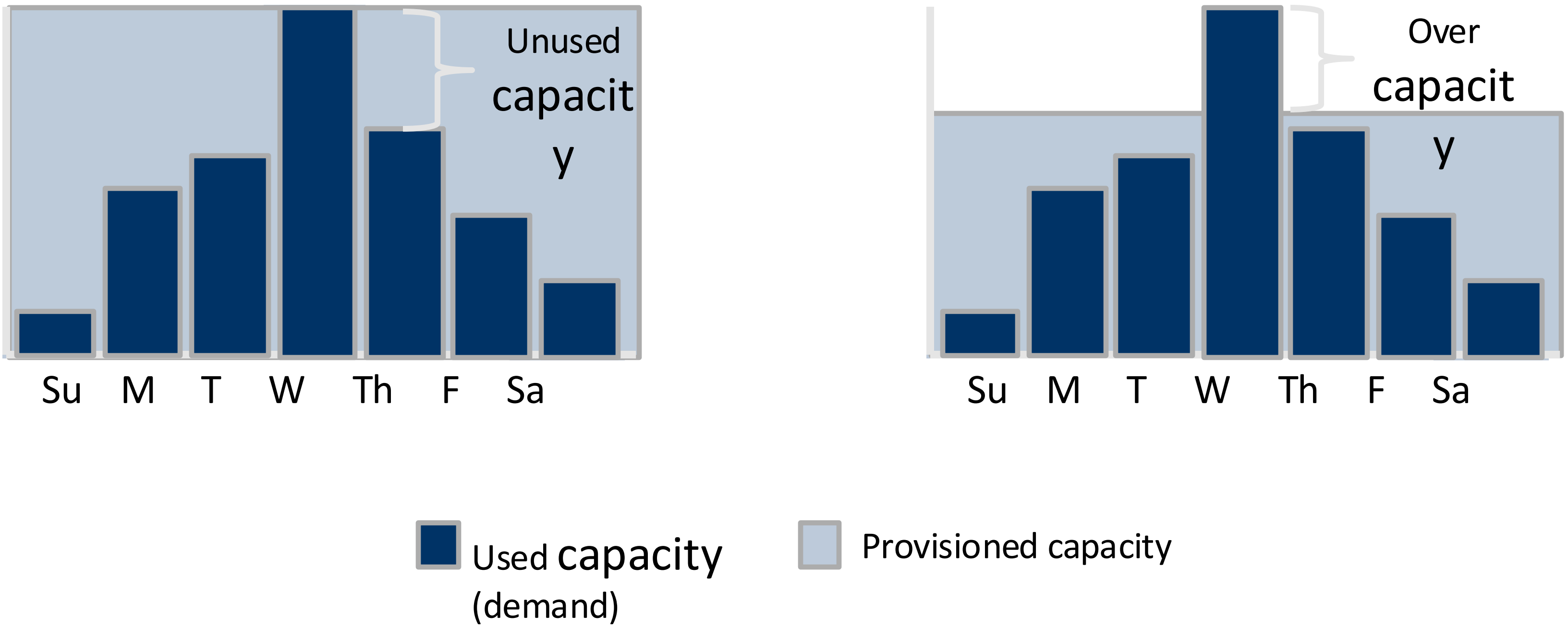
Must be a number

Additional configuration

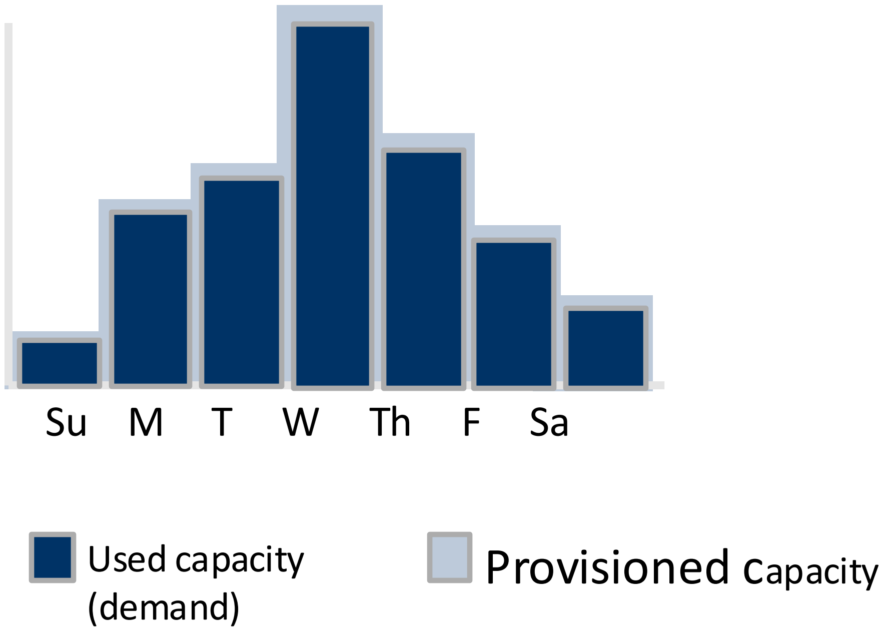
Module 10: Automatic Scaling and Monitoring

SECTION 3: AMAZON EC2 AUTO SCALING

Why is scaling important?

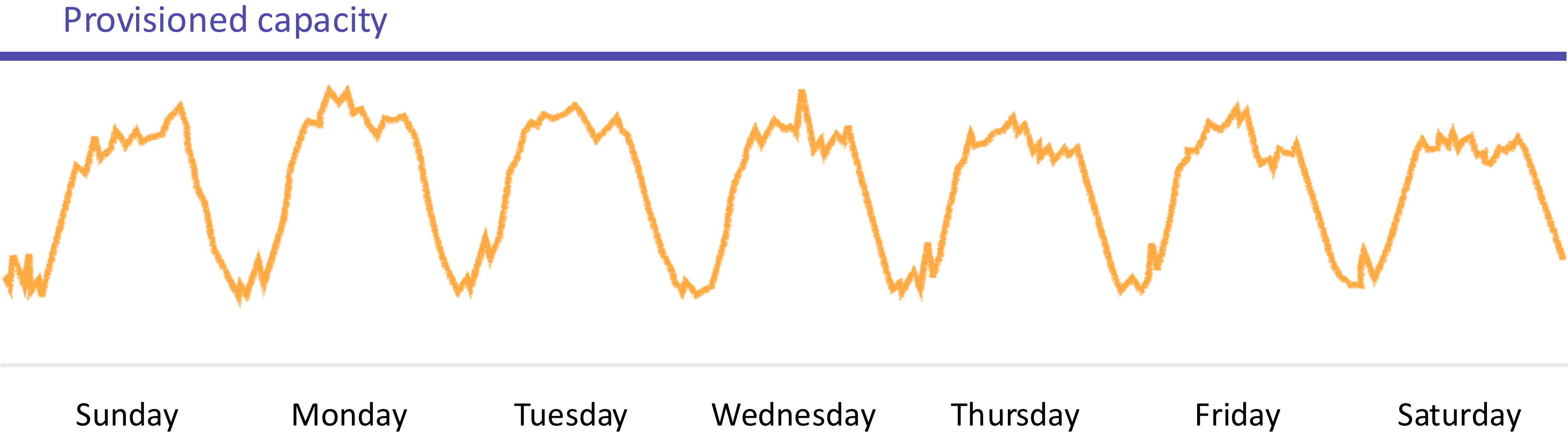


Amazon EC2 Auto Scaling

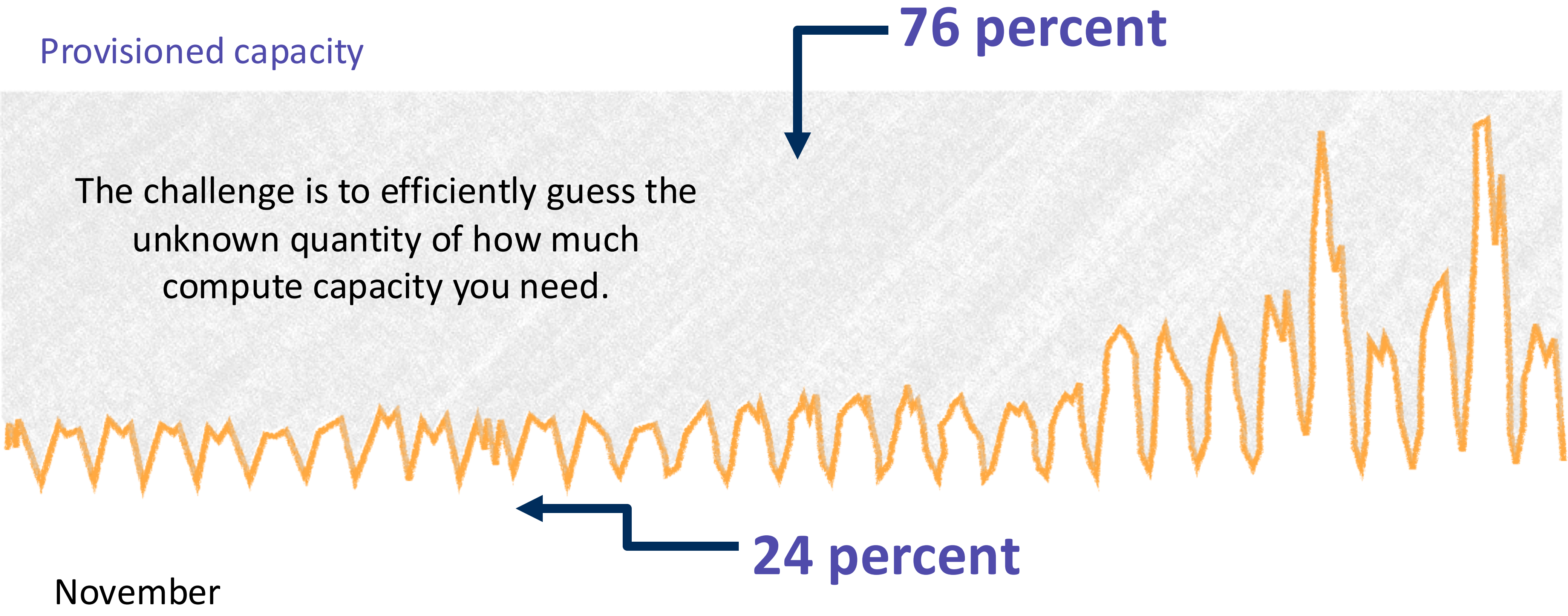


- Helps you maintain application availability
- Enables you to automatically add or remove EC2 instances according to conditions that you define
- Detects impaired EC2 instances and unhealthy applications, and replaces the instances without your intervention
- Provides several scaling options – Manual, scheduled, dynamic or on-demand, and predictive

Typical weekly traffic at Amazon.com

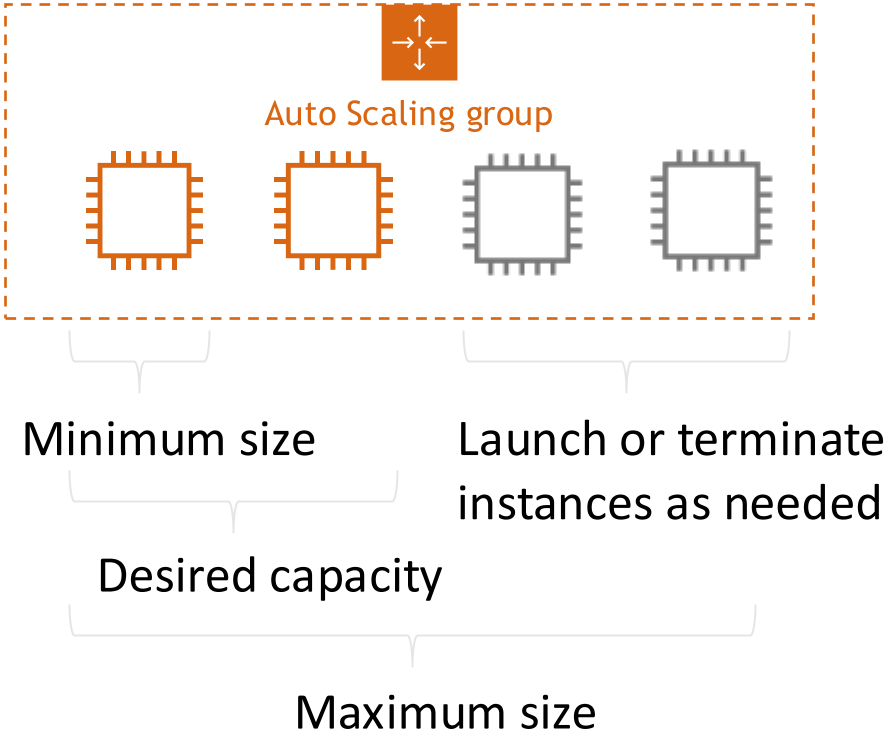


November traffic to Amazon.com



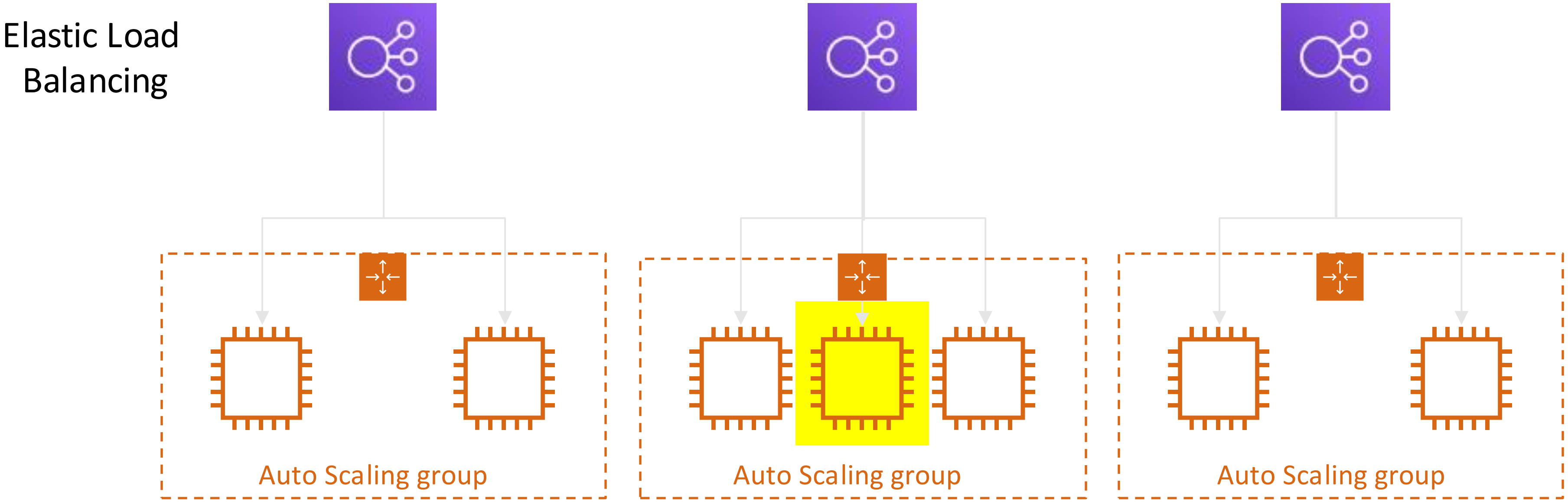
The challenge is to efficiently guess the unknown quantity of how much compute capacity you need.

Auto Scaling groups



An **Auto Scaling group** is a collection of EC2 instances that are treated as a logical grouping for the purposes of automatic scaling and management.

Scaling out versus scaling in



Elastic Load Balancing

Auto Scaling group

Auto Scaling group

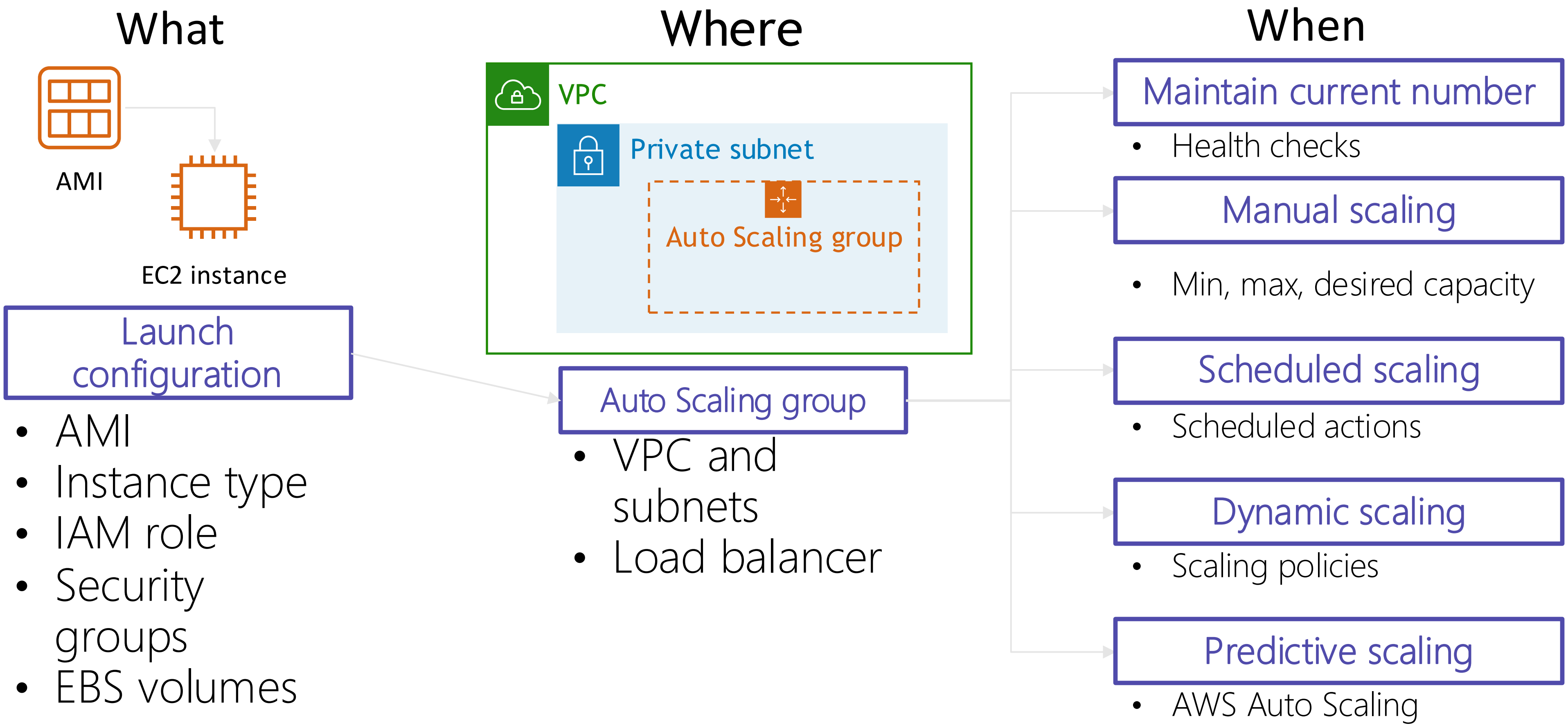
Auto Scaling group

Base configuration

Scale out
(launch instances)

Scale in
(terminate instances)

How Amazon EC2 Auto Scaling works



What

Launch configuration

- AMI
- Instance type
- IAM role
- Security groups
- EBS volumes

Where

Auto Scaling group

- VPC and subnets
- Load balancer

When

Maintain current number

- Health checks

Manual scaling

- Min, max, desired capacity

Scheduled scaling

- Scheduled actions

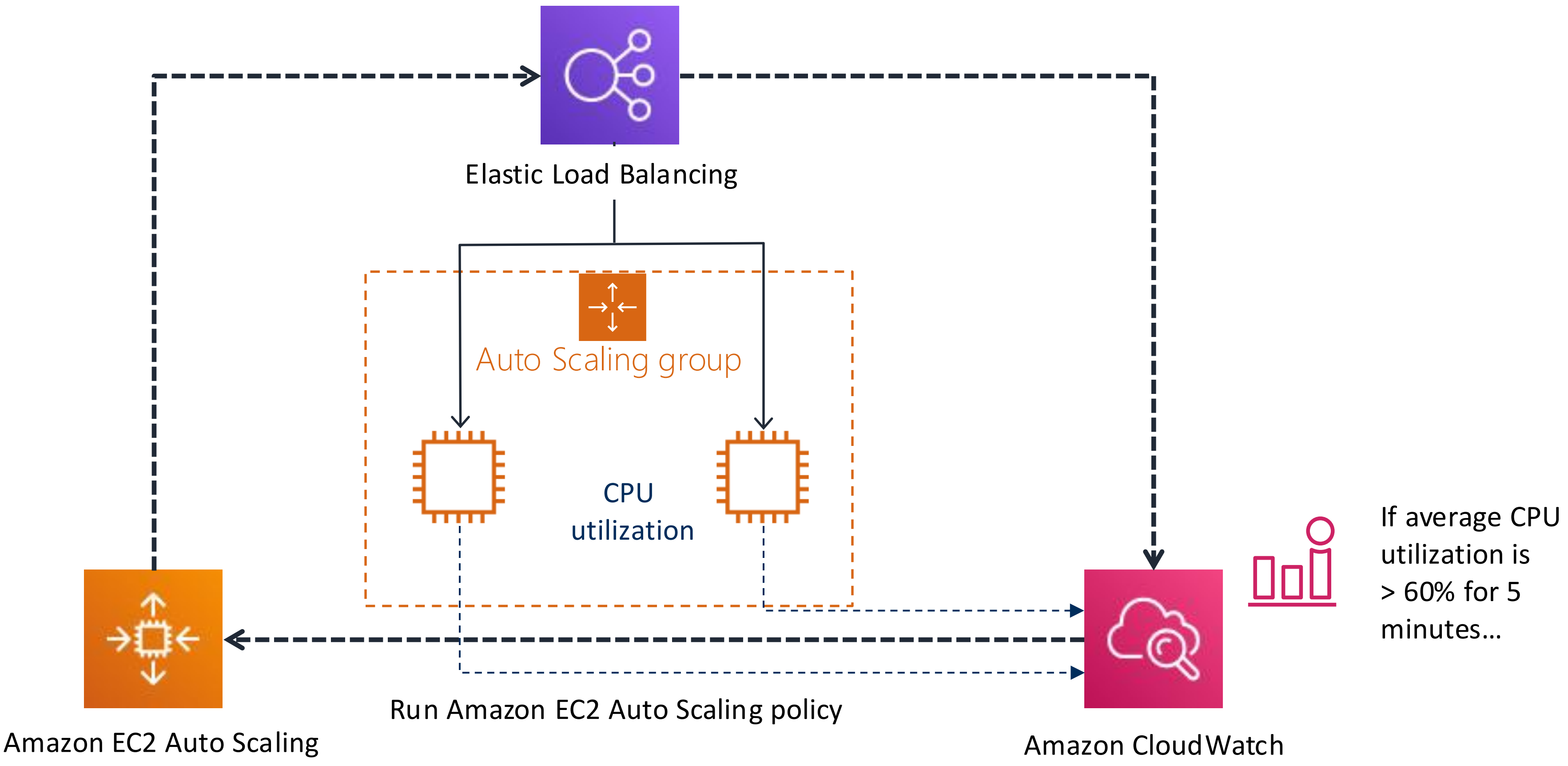
Dynamic scaling

- Scaling policies

Predictive scaling

- AWS Auto Scaling

Implementing dynamic scaling



If average CPU utilization is > 60% for 5 minutes…

AWS Auto Scaling

AWS Auto Scaling

- Monitors your applications and automatically adjusts capacity to maintain steady, predictable performance at the lowest possible cost
- Provides a simple, powerful user interface that enables you to build scaling plans for resources, including –
  - Amazon EC2 instances and Spot Fleets
  - Amazon Elastic Container Service (Amazon ECS) Tasks
  - Amazon DynamoDB tables and indexes
  - Amazon Aurora Replicas

OFF TOPIC

We use Agile
Be honest
We implemented SCRUM for our tasks
I said honest
We cut Waterfall into sprints
Thank you
imgflip.com

IF YOU ARE

NOT BUILDING SW

YOU ARE

NOT LEARNING!