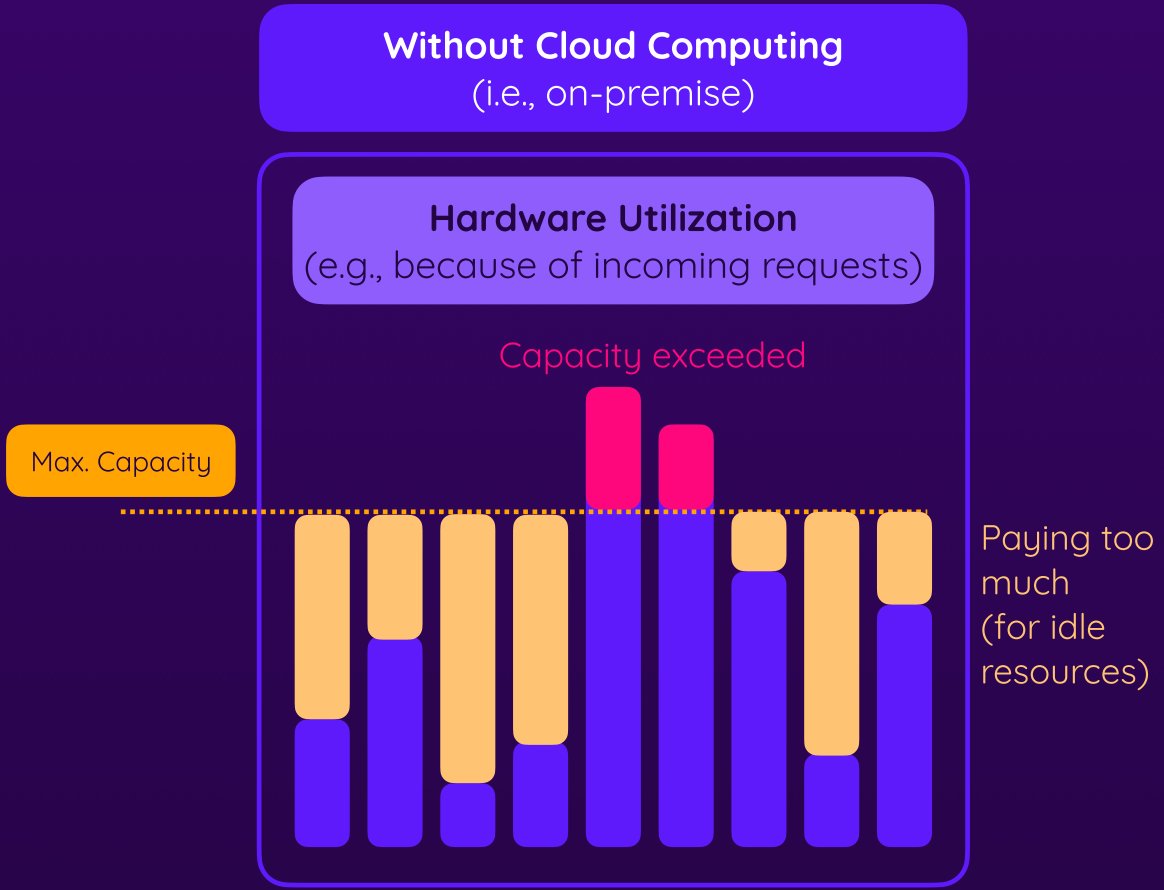
Dynamic Scaling & Load Balancing

Building for scale

- Scaling & Load Balancing: What & Why?
- Understanding AWS Auto Scaling
- Understanding AWS Elastic Load Balancers

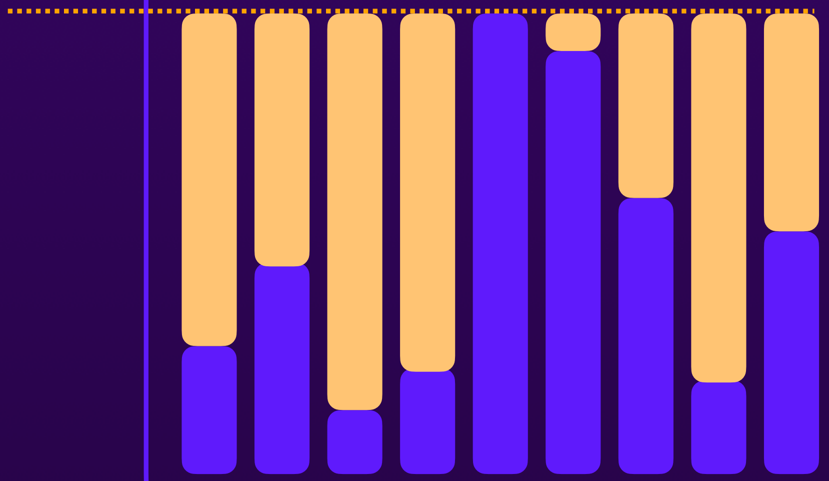
The Need For Flexibility



The Need For Flexibility

Without Cloud Computing
(i.e., on-premise)

Hardware Utilization
(e.g., because of incoming requests)

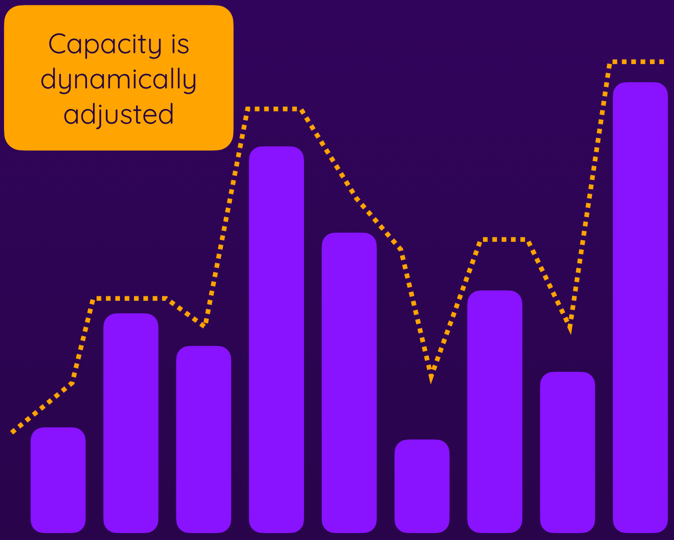


Max. Capacity

Paying too much (for idle resources)

With Cloud Computing
(e.g., via AWS services)

Hardware Utilization
(e.g., because of incoming requests)



Capacity is dynamically adjusted

AWS Compute Scaling Services

EC2 Auto Scaling

Service which can be used to automatically add / remove EC2 instances (based on conditions)

Ensures sufficient capacity at all times, without over-provisioning

Elastic Load Balancer (ELB)

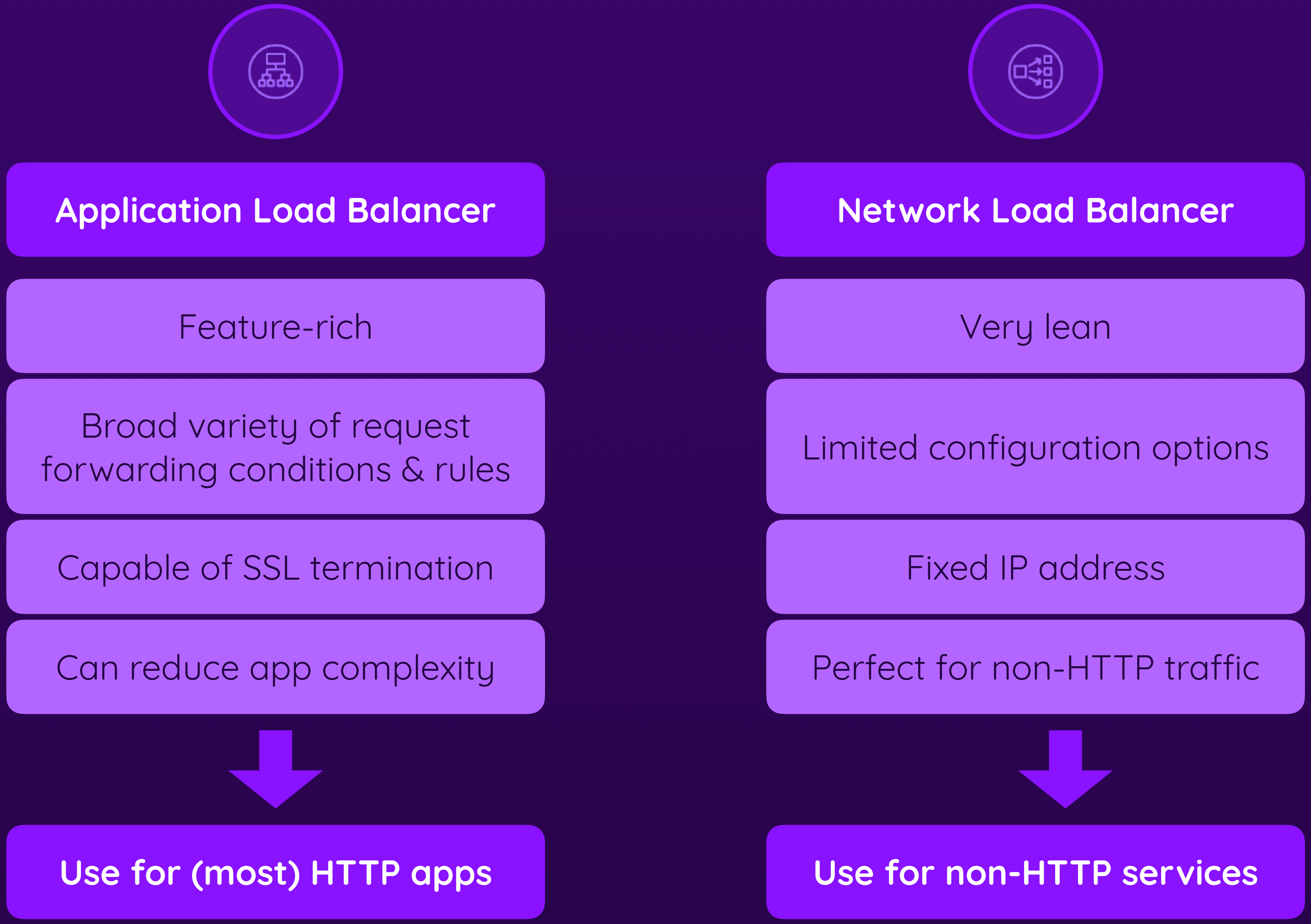
Service to distribute load (e.g., incoming requests) evenly across available instances

Ensures that all available instances are utilized equally

Application Load Balancer

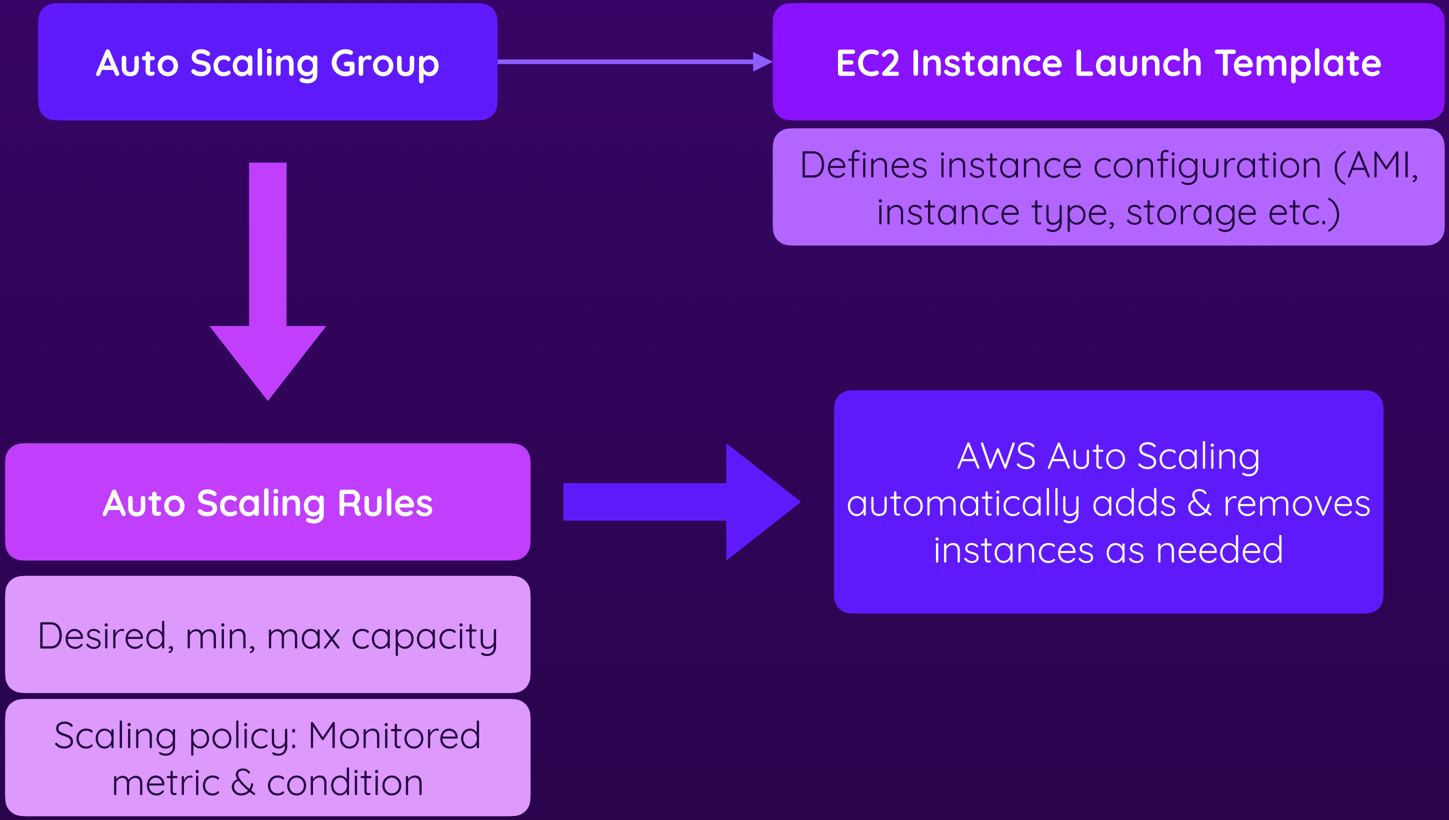
Network Load Balancer

Elastic Load Balancer



Application Load Balancer	Network Load Balancer
Feature-rich	Very lean
Broad variety of request forwarding conditions & rules	Limited configuration options
Capable of SSL termination	Fixed IP address
Can reduce app complexity	Perfect for non-HTTP traffic
Use for (most) HTTP apps	Use for non-HTTP services

Using Auto Scaling



Auto Scaling Group

EC2 Instance Launch Template

Defines instance configuration (AMI, instance type, storage etc.)

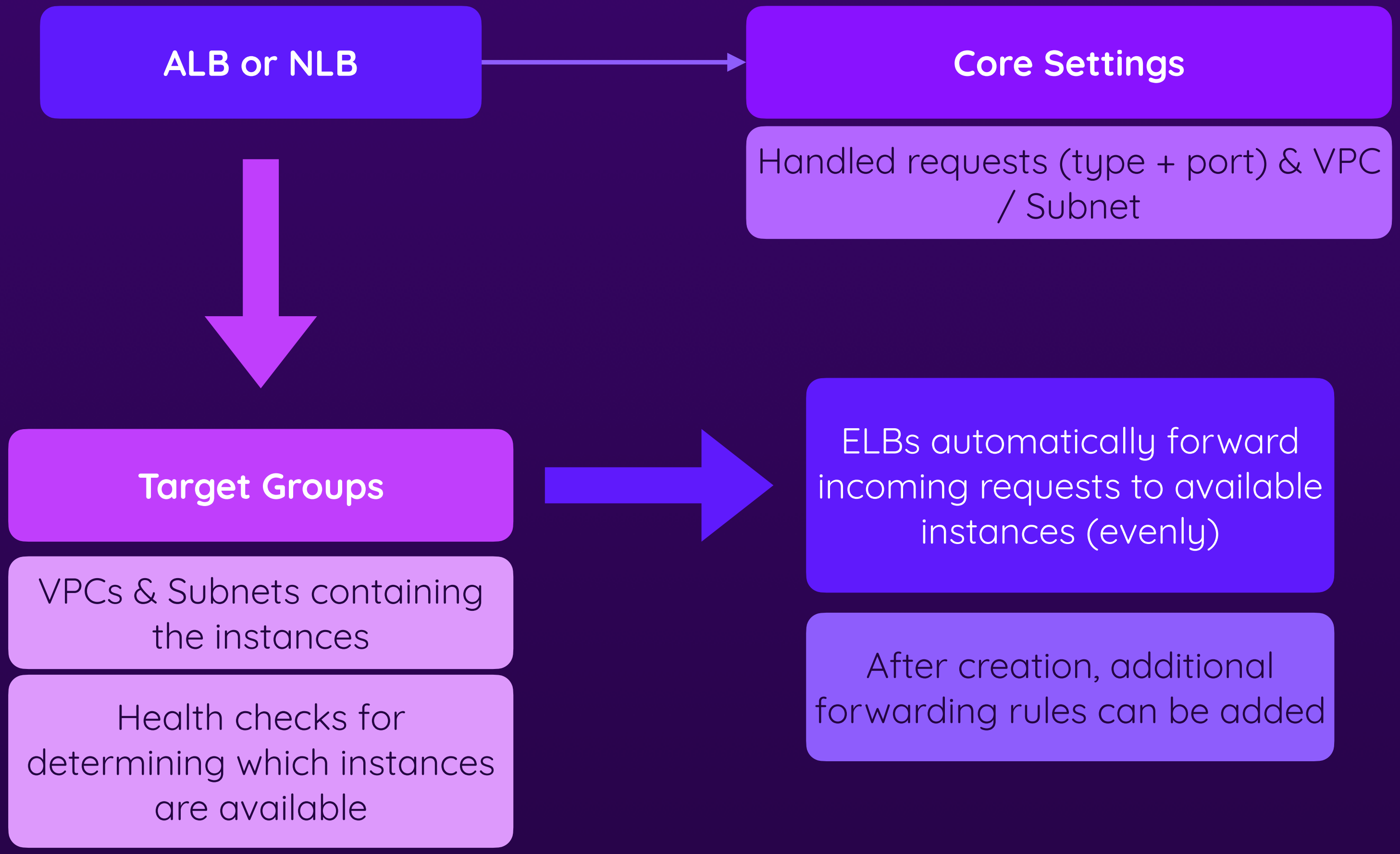
Auto Scaling Rules

Desired, min, max capacity

Scaling policy: Monitored metric & condition

AWS Auto Scaling automatically adds & removes instances as needed

Using Load Balancers



ALB or NLB

Core Settings

Handled requests (type + port) & VPC / Subnet

Target Groups

VPCs & Subnets containing the instances

Health checks for determining which instances are available

ELBs automatically forward incoming requests to available instances (evenly)

After creation, additional forwarding rules can be added

Summary

Elasticity, Scalability & High Availability	Auto Scaling	Elastic Load Balancer
Workloads don’t necessarily have even load patterns	Automatically add / remove instances	ALB & NLB can be used for distributing traffic evenly
Too little or too much capacity can be a big problem	Set clear rules and min / max requirements	Define target groups (in VPCs / Subnets) and forwarding rules
Being able to scale instantly & automatically is important	Instance count is adjusted to incoming load based on rules	ALB is perfect for HTTP traffic (and feature-rich)
Load should also be distributed evenly to avoid downtimes	Use launch templates & VPC / subnet settings	NLB is great for other network traffic