Agenda

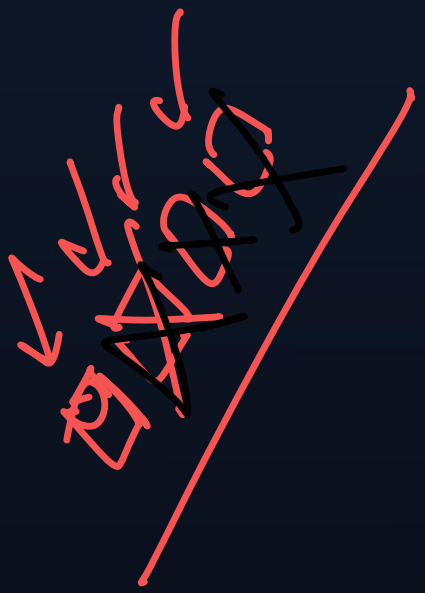
1. Capacity management
2. Scaling options
   - Horizontal/Vertical scaling
3. Setup HS, VA
   - ASG → LT → Scheduled Action → Lifecycle policy
4. Types of LB - ALB, NLB, GLB, CLB -

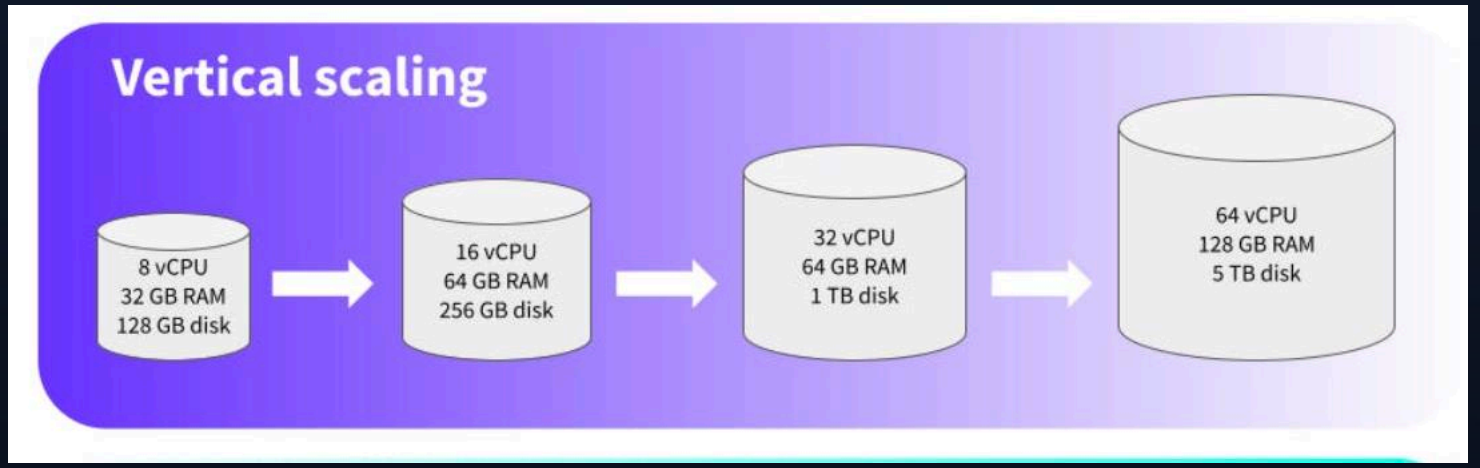
Quick Recap of EC2 Instance, AMI and Backups

1. Vertical Scaling

- *Vertical scaling* refers to increasing the capacity of a system by adding capability to the machines it is using (as opposed to increasing the overall number of machines). This is also called *scaling up*.
- Vertical scaling can be upgrading the physical machine that's running your system, or it can be swapping over to a different, more capable machine. As long as the *number* of machines our system uses isn't changing, that's scaling vertically.

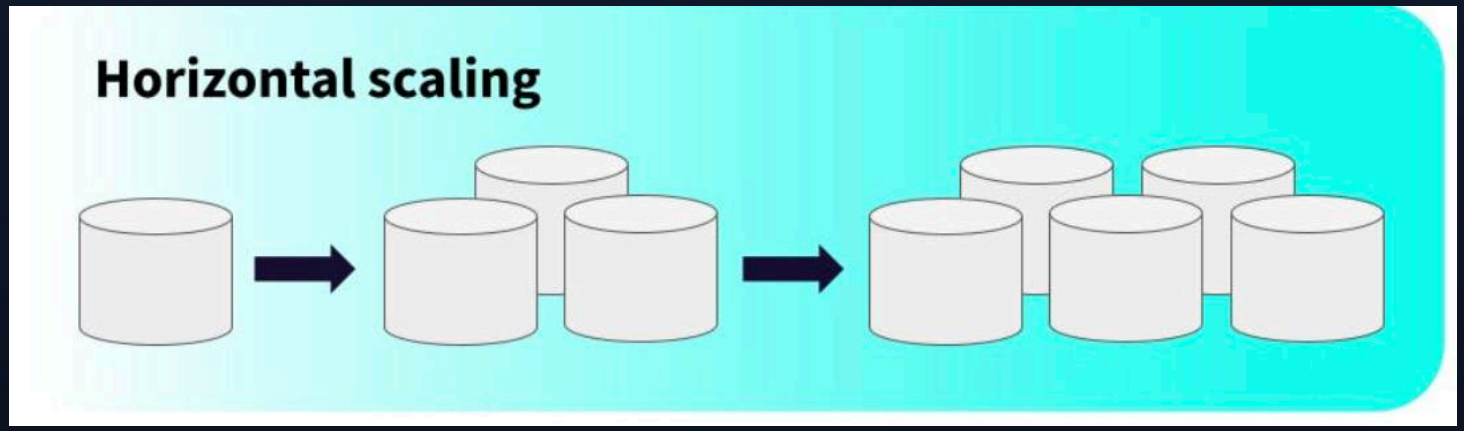
For example, imagine that we have an application with a cloud database that has reached the limits of the server it's running on: a single 8 vCPU GCP instance with 32 GB of RAM.





2. Horizontal Scaling

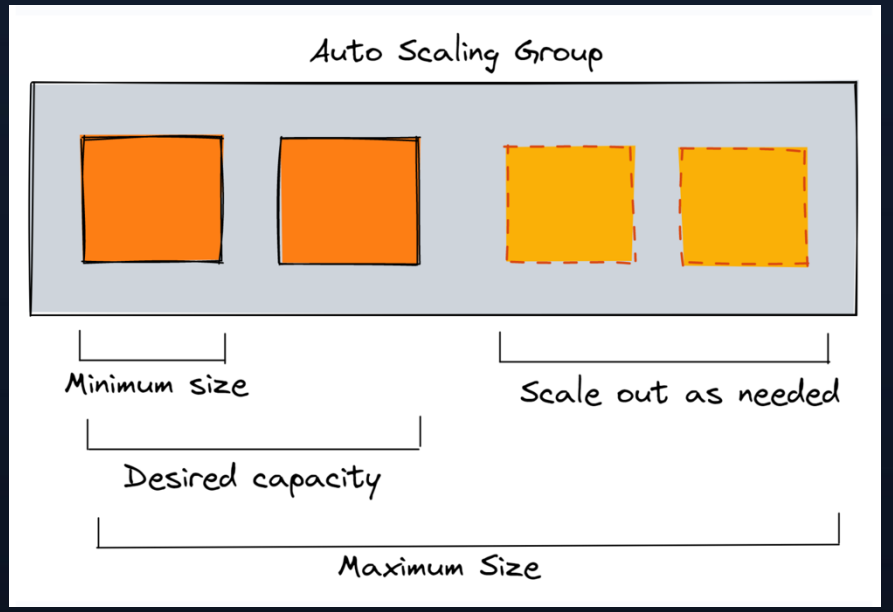
- *Horizontal scaling* refers to increasing the capacity of a system by adding additional machines (nodes), as opposed to increasing the capability of the existing machines. This is also called *scaling out*.
- *For example, imagine that we again have an application with a cloud database that has reached the limits of the server it's running on: a single 8 vCPU GCP instance with 32 GB of RAM.*



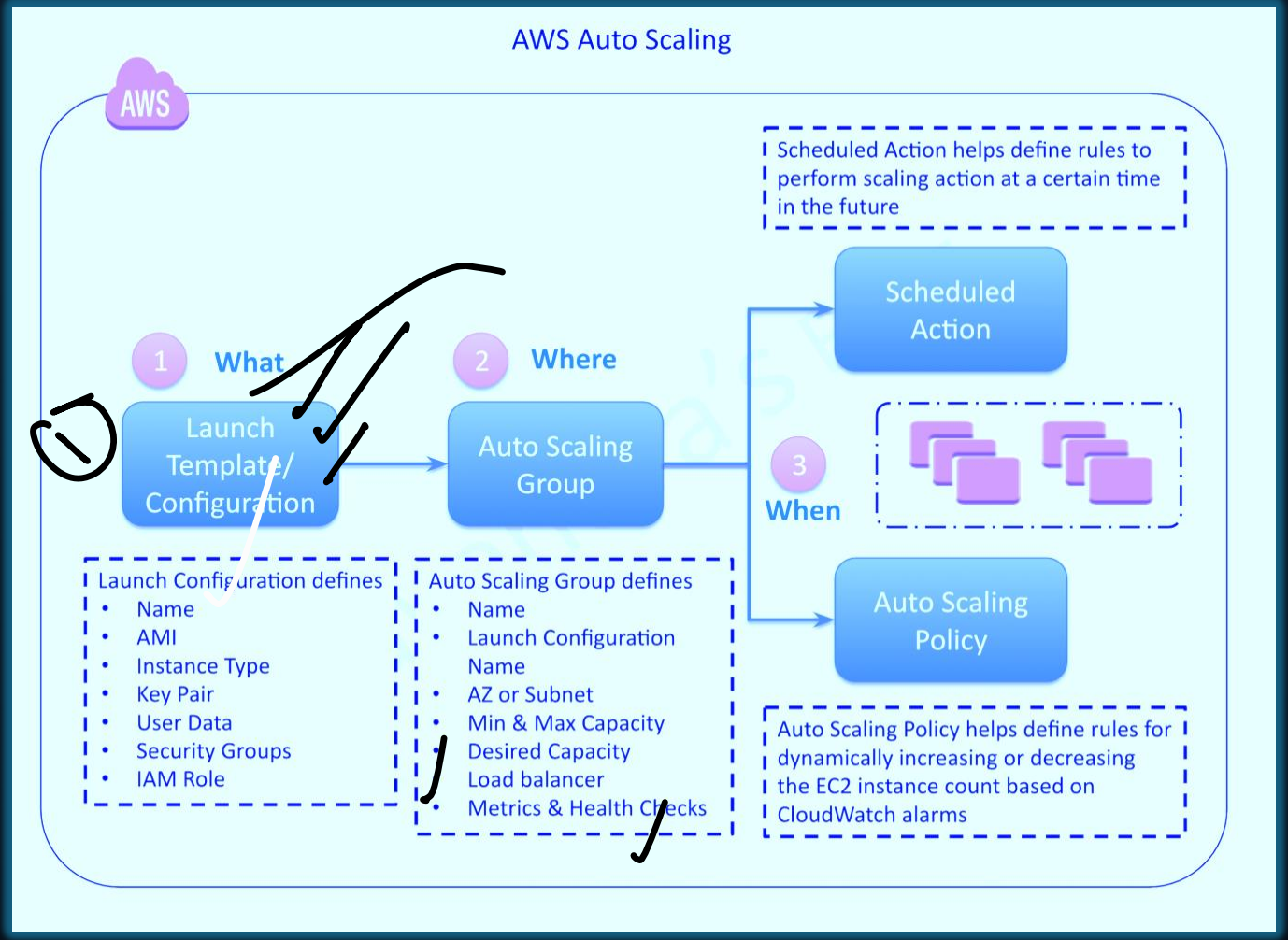
Introduction to Auto Scaling

Auto-scaling is a feature of **cloud computing** that helps organizations scale the capacity of cloud services or virtual machines up or down based on traffic levels.

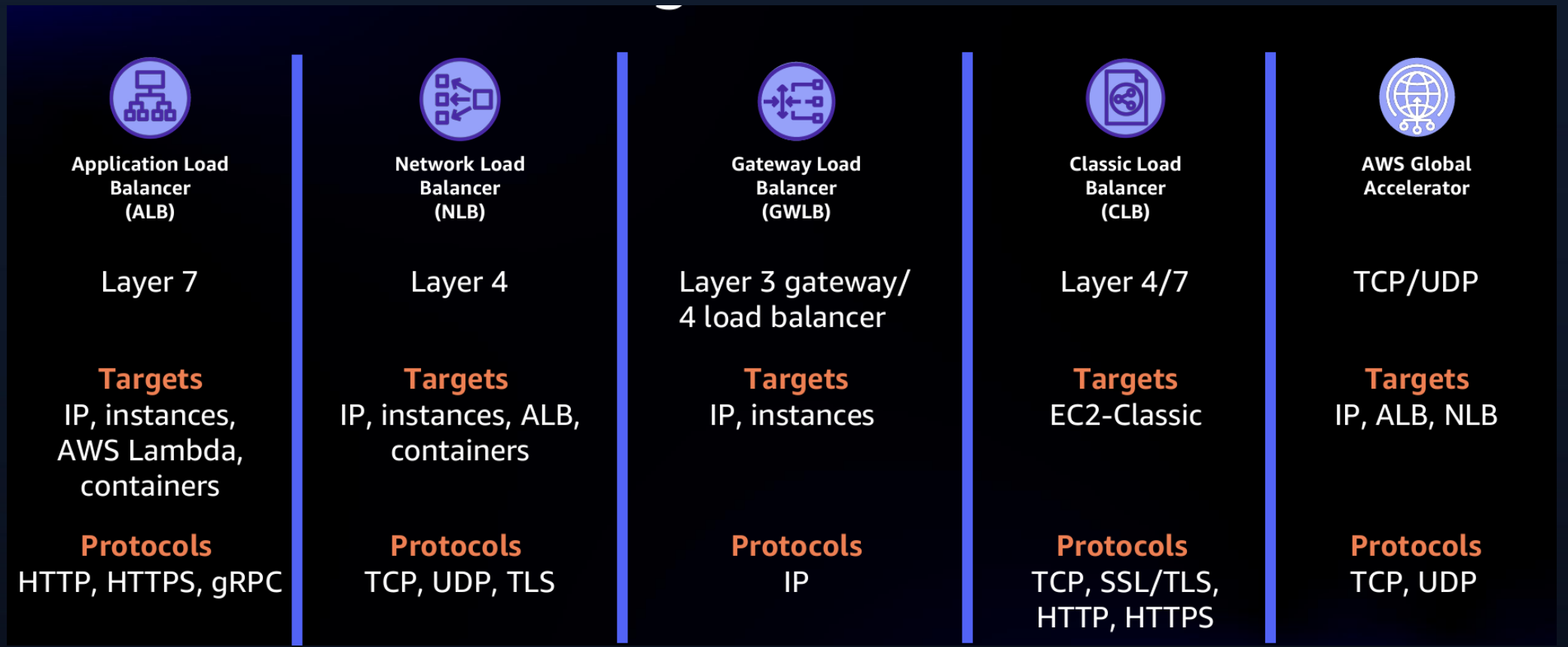
By automatically expanding and lowering fresh instances as demand rises and falls, auto-scaling reduces cost and enables consistent functionality. As a result, amidst changing and often unforeseen requests for services, auto-scaling ensures stability. Auto-scaling also avoids the need to react explicitly to heavy traffic in real-time which would necessitate more tools and instances. Furthermore, auto-scaling allows for the installation, tracking, and deactivation of each unit.



ASG Components



Types of Load balancer



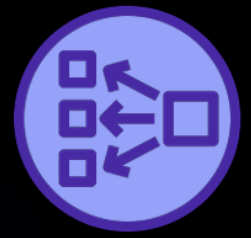
Which load balancing technology should we use?

Targets	Requires
Instances	Layer 7 routing
AWS Lambda	HTTP2/gRPC
Containers	Redirects, web sockets
IP	Fixed response
	Authentication
	Web application firewall, AWS Outposts/AWS Local Zones
	Cookie stickiness, HTTP Desync mitigation
	Best option for the AWS Load Balancer Controller for containers

Application Load Balancer

Which load balancing technology should we use?

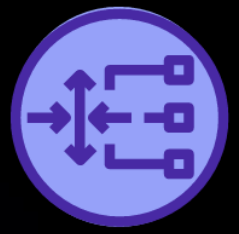
Targets	Requires
Instances	Low latency
ALB	Zonal isolation
Containers	Long-lived TCP connections
IP	Connection-based Layer 4 load balancing
	PrivateLink support
	Elastic IP support
	Hybrid architecture support
	AWS Fargate support direct to K8s pod



Network Load Balancer

Which load balancing technology should we use?

Targets	Requires
Instances	Bump in the wire
	Auto scaling for packet processing devices (firewall, IdP)
	Packet preservation for inspection
IP	PrivateLink GWLB endpoint
	Multi-port to same instance
	Route table entry



Gateway Load Balancer

Which load balancing technology should we use?

Targets	Requires
NLB	Accelerate latency-sensitive applications
ALB	Improve resiliency and availability on a global scale
IP	Simplified global traffic management
	Global set of anycast static IP addresses

AWS Global Accelerator