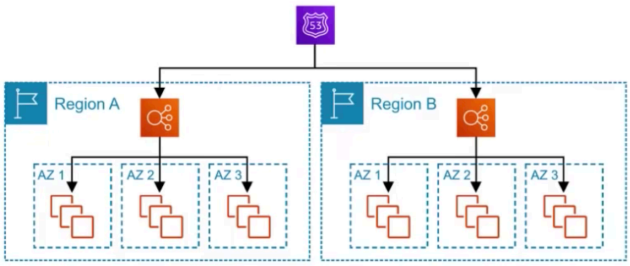
# High Availability (HA)

The ability for a system to remain available

**Think about what could cause a service to become unavailable**:

1. When an AZ becomes unavailable eg. data-center flooded
2. When a Region becomes unavailable eg. meteor strike
3. When an web-application becomes unresponsive eg. too much traffic
4. When an instance becomes unavailable eg. instance failure
5. When a web application becomes unresponsive due to distance in geographic location




The solution we need to implement in order to ensure **High Availability**:

1. We should run our instances in Multi-AZ, an **Elastic Load Balancer** can route traffic to operational AZs.
2. We should run instances in another region. We can route traffic to another Region via **Route53**
3. We should use **Auto Scaling Groups** to increase the amount of instances to meet the demand of traffic
4. We should use **Auto Scaling Groups** to ensure a minimum amount of instances are running and have **ELB** route traffic to healthy instances
5. We should use **CloudFront** to cache static content for faster delivery in nearby regions. We can also run our instances in nearby regions and route traffic using a geolocation policy in **Route53**

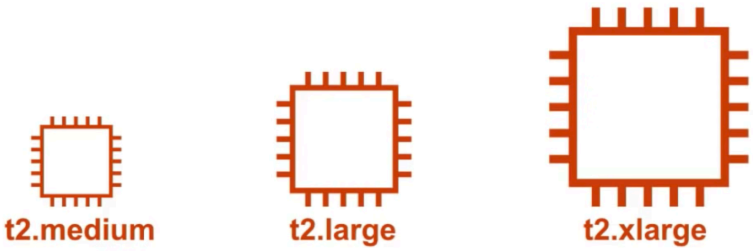
# Scale Up vs Scale Out

When utilization increases and we are reaching capacity we can:

**Scale up (Vertical Scaling)**

Increasing the size of instances

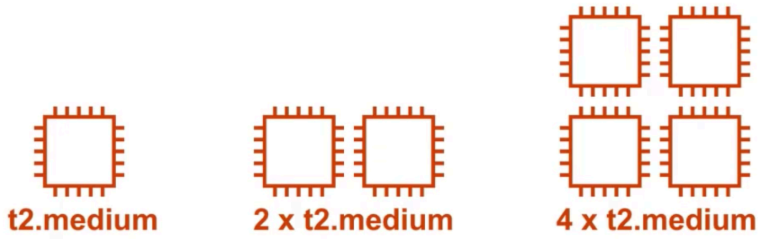
- Simpler to manage.
- Lower availability (if a single instance fails service becomes unavailable)




**Scale out (Horizontal Scaling)**

Adding more of the same

- More complexity to manage.
- Higher availability (if a single instance fail it doesn't matter)




You will generally want to **scale out** and **then up** to balance complexity vs availability