

Scalability on AWS

Scalability and Cloud Native Architecture on AWS

Gaurav Raje

Director of Engineering

Author - Security and Microservice Architecture On AWS (O'Reilly 2021)



Level: Beginner




Structure of This Presentation

<u>Introduction</u>	Who I am, why do I have a job? What is architecture? Why you may want one?
<u>Problem Description</u>	What is scalability? What problem does it solve? How does it solve the problem?
<u>Cloud Toolbox</u>	What tools does AWS provide us with to make our application more scalable? What are the pros and cons of each set of tools? When to use what tool?
<u>Best Practices</u>	What are some select rules of thumb that you can use with the tools described above?
<u>Conclusion</u>	While I could not cover everything, what message do I think you should walk away with so that I believe you will gain the most out of this presentation?

Weaving business outcomes with security and technology.

What is the abstract of this presentation?

Level: Beginner



Gaurav Raje

<u>Author</u>	Security and Microservice Architecture on AWS (O'Reilly Media, 2021) *
<u>Director of Engineering (By the day)</u>	Worked in various software companies as a developer, architect, engineering leader and a cloud evangelist. Find me on LinkedIn
<u>Research Scholar and Doctorate - Business Administration</u>	Research Scholar – Rutgers University (International Business – Corruption in International Business) **
<u>Master of Business Administration (MBA)</u>	NYU Stern School of Business (Finance)
<u>Master of Science (AI)</u>	Rochester Institute of Technology

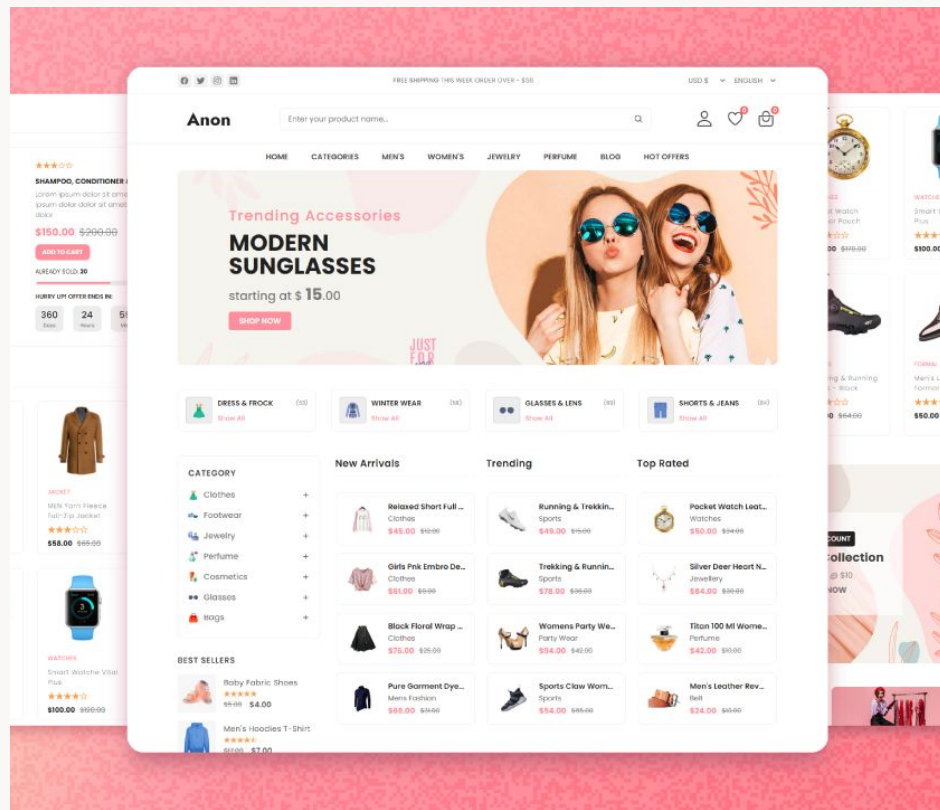
Who am I? What is my background?

** All proceeds are donated*

*** Ongoing, Dissertation defense January 2024*

Example for the Presentation

- Throughout the presentation, I will use a typical E-Commerce website as a sample application for all of my examples.
- Think of amazon.com where you can search for items in different ways and buy them
- Searches can be ad hoc or just based on title



What example will I use throughout the presentation?

Dimensions of Architecture



Efficiency

Write code that can process more



Scalability

Ability to add resources during traffic spikes.



Availability

Avoid downtime



Security

Reduce the threat posed by malicious actors



Cost

Reduce the cash burn of your application



Simplicity

Ensure that your application is modular

What is architecture? What are trade offs?

AWS Shared Responsibility Model

Pizza As A Service

Serverless	Managed Services with Autoscaling	Managed Services	Running on EC2	On Prem
Dining Table	Dining Table	Dining Table	Dining Table	Dining Table
Soda	Soda	Soda	Soda	Soda
Electric/Gas/Oven	Electric/Gas/Oven	Electric/Gas/Oven	Electric/Gas/Oven	Electric/Gas/Oven
Olive Oil	Olive Oil	Olive Oil	Olive Oil	Olive Oil
Pizza Dough	Pizza Dough	Pizza Dough	Pizza Dough	Pizza Dough
Tomato Sauce	Tomato Sauce	Tomato Sauce	Tomato Sauce	Tomato Sauce
Cheese	Cheese	Cheese	Cheese	Cheese
Toppings	Toppings	Toppings	Toppings	Toppings

Dining in a Restaurant	Takeout	Boxed Dinner	Hello Fresh	Making from Scratch
------------------------	---------	--------------	-------------	---------------------

Your Responsibility	Someone else's responsibility
---------------------	-------------------------------

Software as a Service

Serverless	Managed Services with Autoscaling	Managed Services	Running on EC2	On Prem
Scaling Criteria	Scaling Criteria	Scaling Criteria	Scaling Criteria	Scaling Criteria
Provisioning	Provisioning	Provisioning	Provisioning	Provisioning
Server Reboots	Server Reboots	Server Reboots	Server Reboots	Server Reboots
High Availability	High Availability	High Availability	High Availability	High Availability
Software Upgrades	Software Upgrades	Software Upgrades	Software Upgrades	Software Upgrades
Software Installation	Software Installation	Software Installation	Software Installation	Software Installation
Network Connectivity	Network Connectivity	Network Connectivity	Network Connectivity	Network Connectivity
Physical Security	Physical Security	Physical Security	Physical Security	Physical Security

Dining in a Restaurant	Takeout	Boxed Dinner	Hello Fresh	Making from Scratch
------------------------	---------	--------------	-------------	---------------------

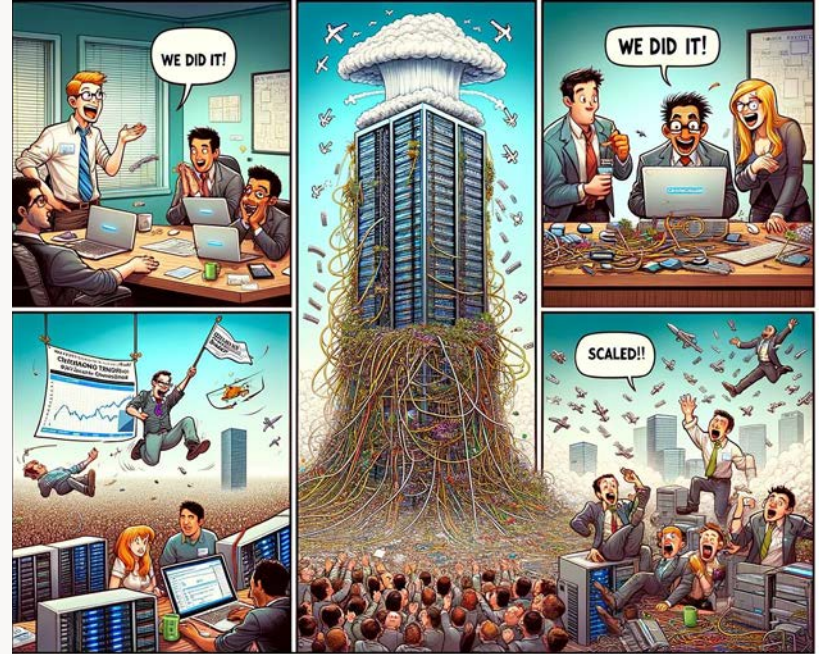
Your Responsibility	Someone else's responsibility
---------------------	-------------------------------

Ref - Albert Barron

As AWS architects, what tools do you have to in your toolbox?

Scalability vs Efficiency

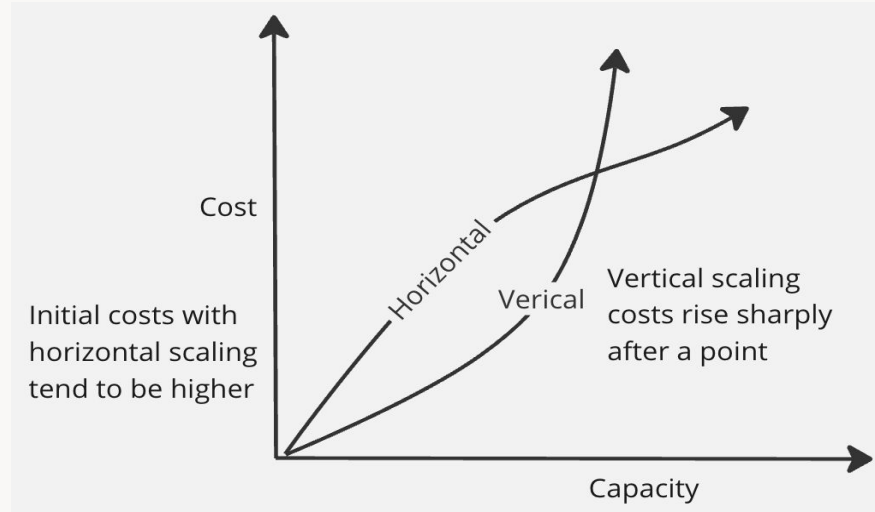
- What is the common problem that both are trying to solve?
- What is the difference between both of these approaches?
- When do you need scalability?
- What do you get as a result of scalability?
- What do you lose as a result of scalability?



What is scalability? How is it different from efficiency?

Types of Scalability

- **Types of Scalability**
 - **Vertical**
 - **Horizontal**
- **When to use vertical scalability**
- **When to use horizontal scalability**
- **Practical rule of thumb**
- **How cloud makes it easy**



What is scalability? What are the broad types?



Examples of Vertical Scalability

- Resizing an EC2 instance from a small to a large instance type, which results in either a higher RAM or CPU or throughput.
- Resizing your instance from a general purpose instance to a memory optimized instance (or a CPU optimized instance) where you get more RAM (or CPU) for a similar cost
- Moving your database from a shared database cluster to a dedicated database cluster so it can get dedicated processing power

What is scalability? What are the broad types?



Vertical Scalability

- Larger Instance Sizes
 - RAM
 - CPU
 - Throughput / IO / Storage
- RAM vs CPU
 - T4g micro vs t4g small
 - 0.008 cents price
 - 1 GB RAM. Same CPU
 - T4g medium vs t4g large
 - 0.0336 price
 - 4 GB RAM. Same CPU
 - T4g large vs T4g xlarge
 - 0.1344 price
 - 8 GB RAM 2 CPU.
- 0.008 Per GB. CPU increases follow

Instance name ▲	On-Demand hourly rate ▼	vCPU ▼	Memory ▼
t4g.nano	\$0.0042	2	0.5 GiB
t4g.micro	\$0.0084	2	1 GiB
t4g.small	\$0.0168	2	2 GiB
t4g.medium	\$0.0336	2	4 GiB
t4g.large	\$0.0672	2	8 GiB
t4g.xlarge	\$0.1344	4	16 GiB
t4g.2xlarge	\$0.2688	8	32 GiB

What is vertical scalability?

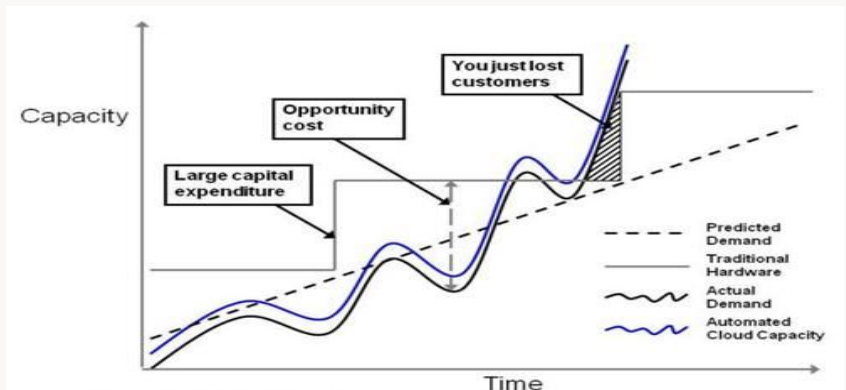
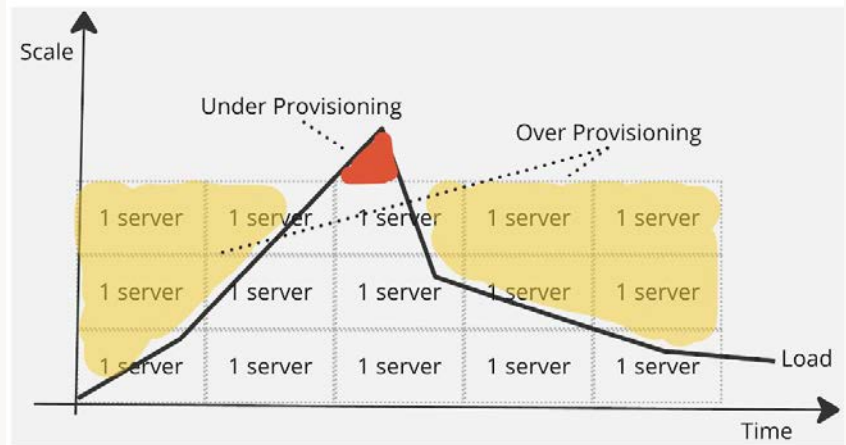


Horizontal Scalability

- Demand projection and planning
- What does it even mean to have software that is horizontally scalable?
- What happens if scalability is too rigid?
- Elasticity vs Scalability

Ref - <https://www.liquidweb.com/blog/horizontal-vs-vertical-scaling/>

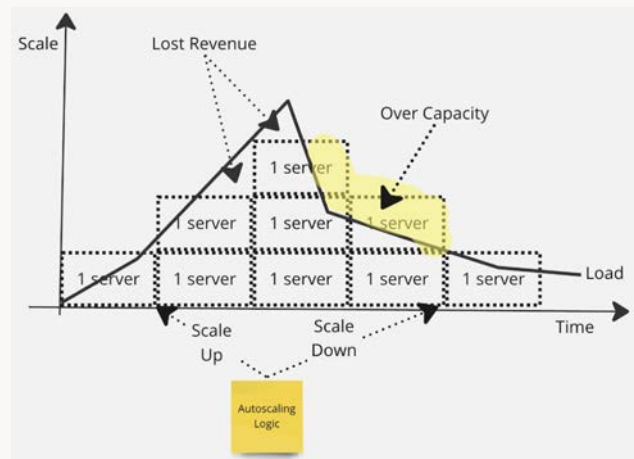
What is horizontal scalability?



Based on illustrations by Amazon Web Services

Auto scaling Applications

- The application can add servers when certain events are triggered
- You have to define when a new resource needs to be added
- In some cases, you may need to add custom scripts to scale efficiently or scale down efficiently

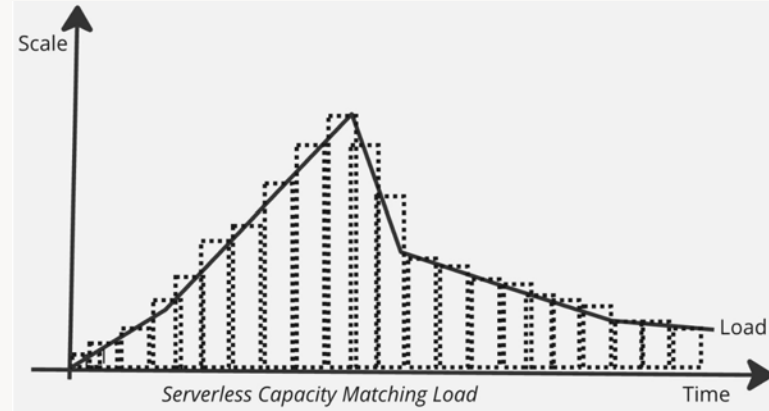


What is scalability?



Serverless Applications

- If your application performs a specific task that is supported, AWS handles the scalability up to a reasonable level as long as you conform to the AWS specs
- Turnkey solutions for specific tasks:
 - AWS Lambda. AWS only provisions the resources based on your load
 - AWS Postgres Serverless will scale as CPU or Connections pile up.
 - AWS DynamoDB autoscales but is designed for key-value lookups



What is scalability?



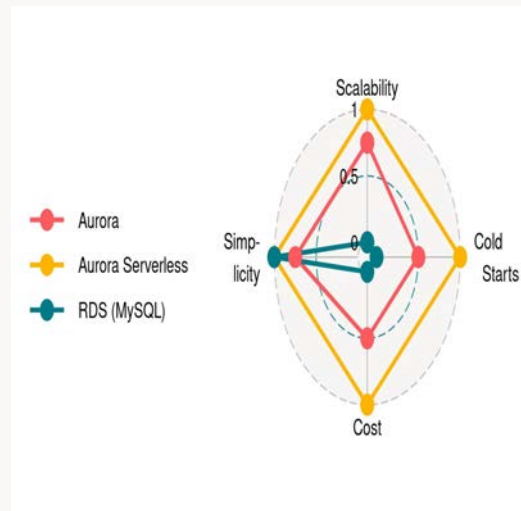
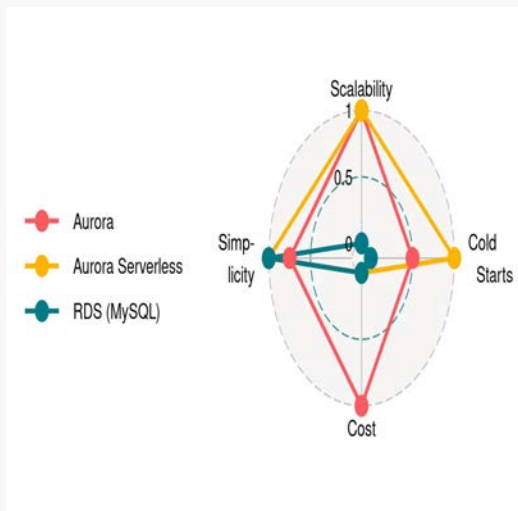
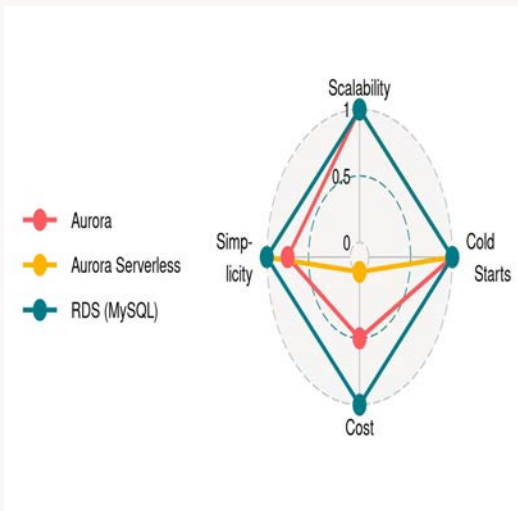
Comparison of Databases



Demand is sustained and high

Demand is low except during lunchtime when it spikes

Demand is bursty



Thank You



Key Takeaway 1

Scalability means your software can handle more work (requests) when you give it more resources (like servers or better hardware). It's different from efficiency, which focuses on doing the same work with fewer resources.

Key Takeaway 2

Scalability can be achieved in multiple ways. In almost all situations, there are tradeoffs that need to be made. Finding what is most important will dictate the optimal path

If I had to remember two things from this talk, what would they be?