



Introduction to Serverless

Beyond Traditional Infrastructure







On-Premises

What media should I use to keep backups?

What size of servers should I buy?

How do I deploy new code to my servers?

How can I scale my app?



Which packages should be on my server?



Do I need a secondary Network connection?

Who has physical access my servers?

Who monitors my servers?

How many servers do I need?

Do I need a UPS?

It takes how long to provision a new server?

What happens in case of server hardware failure?

Who monitors my apps?



How often should I backup my server?



Are my servers in

How can lincrease

server utilization?



How do I keep the operating system up to date?



What happens in case of

server hardware failure?

How often should I patch my severs?

What storage do I need to use?

How can I dynamically configure my app?







laaS

What media should I nar uneque packups;

What size of servers should I buy?

What happens in case of What is the right size of servers for myseusiness needs? case of hardware failure? How can increase server utilization?

my apps?

How do I deproject willight to do I need?

How can I scale. How can I **scale** my apprecation? How can lincrease



Which packages should be on my server?



How often should I backup my server?





Do I need a secondary Network connection?

How often should I patch my servers? How of ten should I backup my server?

Servers Which packages should be on my server? Do I need a UPS?

Who monitors my servers?

How do I keep the **operating**system up to date deploy new code to my server?

How do I keep the **operating system** to date?

How do I keep the **operating system** in need to my server?

What happens in case of server hardware failure?

How can I dynamically configure my app?

It takes how long to provision a new server? How often should I patch my severs?







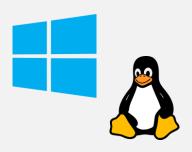
PaaS

What is the right **size** of **servers** for my business needs?

How can I increase **server** utilization?

How **many** servers do I need?

How can I **scale** my application?







How often should I **patch** my **servers**?

How often should I **backup** my **server**?

Which **packages** should be on my **server**?

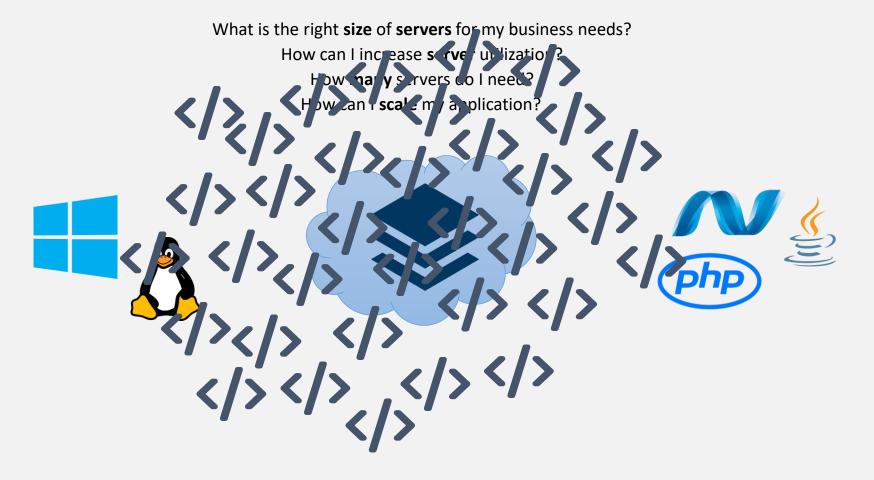








Serverless









What is Serverless?

Serverless computing is a cloud computing model where the cloud provider dynamically manages the allocation and provisioning of servers.







"What's in a name?"

Not there isn't servers

Just, you can think about the servers less

Server Configuration

Server Scaling







Types of Serverless Architecture

Function as a Service (FaaS)

Backend as a Service (BaaS)







Event-Driven







Event-Driven

Short-Lived







Event-Driven

Short-Lived

Automatic Scaling







Event-Driven

Short-Lived

Automatic Scaling

Pay-Per-Execution







Event-Driven

Short-Lived

Automatic Scaling

Pay-Per-Execution Abstraction of Infrastructure







Cost Efficiency







Cost Efficiency

Auto-Scaling







Cost Efficiency

Auto-Scaling

Reduced Operational Overhead







Cost Efficiency

Auto-Scaling

Reduced Operational Overhead

Faster Time-to-Market







Cost Efficiency

Auto-Scaling

Reduced Operational Overhead

Faster Time-to-Market

Trigger-Driven Architecture







Cost Efficiency

Auto-Scaling

Reduced Operational
Overhead

Faster Time-to-Market

Trigger-Driven
Architecture

High Availability







Cost Efficiency

Auto-Scaling

Reduced Operational Overhead

Faster Time-to-Market

Trigger-Driven
Architecture

High Availability

Micro-Billing







Loss of Control







Loss of Control

Cold Starts







Loss of Control

Cold Starts

Usage-Based Pricing







Loss of Control

Cold Starts

Usage-Based Pricing

Provider Lock-In







Loss of Control

Cold Starts

Usage-Based Pricing

Provider Lock-In

Testing and Debugging





















































































Conclusion

- Serverless Architecture offers numerous benefits like cost efficiency and easy scalability.
- However, it presents challenges like vendor lock-in and the cold start problem.
- Despite these challenges, serverless is becoming popular for many organizations due to its numerous advantages.







Overview of Azure Serverless

Harnessing the Power of Microservice Azure



