Serverless Vs Containers Comparison Between Top Two Cloud Services

In modern-day technology, Serverless (Function as a service (FaaS)), and Containers (Container Orchestration) are two of the biggest DevOps buzzwords which signify efficient apps deployment. Touted as cost-effective architectures, they do have some common features but are not interchangeable technologies. While they are different in several aspects, both the models reduce overhead for cloud-hosted web applications.

Serverless and containers are two of the best technologies used by the **DevOps** team. While DevOps team requires extensive resources to dole out the deliverables, containers, and serverless infrastructure are picking up pace in the market. According to the IBM survey, serverless adaptation is increasing at a rapid pace.

Nearly 68 % of techies who are not using serverless technology admitted that they are ready to adopt a serverless approach in the upcoming years. A recent analysis by the

MarketsandMarkets, the serverless architecture experienced a growth of USD 7.6 billion in 2020, and it will grow at USD 21.1 billion in 2025.

Useful Link: Business Benefits of Cloud Computing Migration

Echoing similar sentiments for containers are other surveys. Gartner predicted that 70% of organizations are going to run containerized applications in 2023. An analysis by IBM revealed that 61% of users are using containers for half of their new applications in the past two years.

Based on the unique requirements, your DevOps team can decide on which one to go with. It is a very tough decision to choose between a serverless infrastructure or containers.

Our Veritis team will guide you to the best solution for your application in the cloud platform. Before delving deep into the details, let’s know about serverless computing, and container services in detail.

Serverless

The term serverless might be a bit confusing for some and it doesn't imply that your code shall operate without a server. It is all about building and running applications without having to manage their servers. Simply put, users don't need to get involved with the server management.

In serverless architecture, enterprises don't have to worry about the underlying infrastructure. The organization can focus their efforts on other areas as serverless setup wicks away a variety of bothers which may range from maintenance to capacity constraints as cloud vendor provides the back-end services.

Any enterprise that acquires serverless for their backend services is charged based on their requirement. They don't have to pay extra money for idle resources or unused space. Simply put, this payment model is based on 'pay-as-you-go 'system. With this model, it reduces the cost of deploying applications, instead of providing an auto-scale mode in serverless computing.

Containers

Containers are the standard unit of software that packages the application code, along with its libraries, dependencies, and configuration files which turns into a single object. In container services, the application runs quickly at anywhere, at any environment, be it on the cloud platform, desktop, or traditional IT.

Container technology is the best way for developers to focus only on application logic, while the operations team can concentrate on management and deployment of the application, instead of focusing on specific details like configuration, and software versions.

In the current era, containers are one of the most trending topics in the tech world, especially in security. The top tech companies such as Microsoft, and Facebook are using containers regularly. For instance, Google is relying on containers (using ‘cgroups’ containers) for most of everything. The US tech giant runs 2 billion containers per week (3000 started per second). From Search and YouTube to Gmail, almost everything runs in a container at Google.

Containers first made an appearance over decades ago, with versions like Solaris Containers, AIX Workload Partitions, and FreeBSD Jails. Most developers in 2013 will relate to Docker when containers gained popularity.

Useful Link: Cloud Automation: Services, Tools and Business Benefits

---

Let’s dig deep into the major differences between serverless and containers

Details	Serverless	Containers
Supported Host Environment	Mostly clouds like Azure functions, and AWS Lamba	It can run on certain Windows versions, and Linux servers

Supported Languages	Supports a few programming languages like C#, Node, Java, and Python	Any Virtual language
Isolate Workloads	Yes	Yes
Running Locally	Hectic to operationalize outside a public cloud platform.	It can run easily on a developer's workstation or local data center
Cost	Based on the respective Cloud platfom's billing model	Orchestrators, and container engines are open sources, and they can be used for no cost in a local environment
Stateful functions/app	No (in most cases)	Yes
Support Persistent services	No	Yes
Availability	It can run for a long period of time	It can run for a short period of time
Pros	Reduces maintenance, and administration burden, zero cost for idle time, microservice nature, autoscaling, zero administration, pay per execution, and faster time to market	Reduces maintenance, and administration burden, zero cost for idle time, microservice nature, autoscaling, zero administration, pay per execution, and faster time to market
Cons	Cold starts, Black box environment, complex apps are	Required manual intervention, running costs, hard to get the

	hard to build, no standardization, vendor lock-in	start, administrative work, scaling is slower
Maintenance	The vendor will take care of software updates, and management of servers.	Teams will take care of hardware updates, patches, and maintenance.
Testing	It is difficult to test the function on a serverless environment as backend services are complex to replicate on local machines	It can done easily on local machines to test the function
Deployment	The vendor will take the responsibility of deploying a function	The developer will take the responsibility to deploy the container services
Development Times	The vendor will take care of all things, organization only needs to write the code	Additional work is required for containers like building the image, and moving to the registry
Monitoring	Vendor tools will provide the monitoring capabilities to the serverless	We need to install a monitoring tool in containers
Scaling	Serverless scaling is done by the Vendor	Container scaling is done by the application team

Conclusion

On a concluding note, in the current market scenario, both serverless, and containers are considered as kings in the IT field. To get the best results for adopting serverless, IT Pros need to decide what platform they need to run their application, and what programming language they need to choose for their teams. Most of the application development environments can work with different types of programming languages.

Containers can deliver the best quality software product to their customers by streamlining remote deployment process and local development process, which will help to execute more efficiently. Serverless has its own advantages which are enticing enough for clients who have their hands full.

Useful Link: Cloud Monitoring: Services, Solutions and Tools

When comparing between serverless and containers, it’s best to think that both technologies complement each other. Instead of undertaking a tightrope walk, it may be wise to get the best of both worlds. Most tech industries are likely to use both trending technologies to deliver the same application.

However, this may not always be required as the unique needs of an organization decide which way should one go. This maybe tough for many cloud and DevOps solutions are comprehensible to most. This is where **Veritis** steps in. The trusted tech partner for companies of various scales, Veritis provides cost-effective solutions which are amped up on quality and robustness. Become future-proof by partnering with Veritis.

Services