Creating cost-effective and cloud-independent designs Solutions Native to the Cloud

Sunil Kumar Pandey[1], Dr. Akram Khan[2]
[1]*Research Scholar, Madhav University, Pindwara, Sirohi, Rajasthan*
[2]*Professor, Madhav University, Pindwara, Sirohi, Rajasthan*

Abstract- Cloud Computing has crossed the chasm and it is now for the adopters to avoid being laggards. They must choose it before losing market share to their competitors. Across all the Industry verticals, be it Automotive, Aerospace, Telecom, Life Sciences, Manufacturing or Retail; Cloud has disrupted the traditional business models. IT teams of these verticals are under tremendous pressure to move their solutions to cloud. What makes it complicated is the multiple cloud offerings available to make this happen and every choice with its own trade-offs. Another dimension that makes Infra and IT Architects nervous about cloud journey is the outcome expected from it. Not only that business should run as usual post cloud journey, operating expenditures must come down, returns on investments must be visible and user experience must improve. It makes the decision to design it right the first time paramount. This paper attempts to introduce Cloud computing and its impact, highlight the need to embrace it and an approach for cost-sensitive adopters in implementing Cloud Native Applications.

Keywords- Cloud Native Applications, Cloud Agnostic, Open Source Technologies, Containerization, Microservices, Public Cloud, Private Cloud.

I. CLOUD AND IT'S ADOPTION

It would not be a hyperbole to refer "Cloud" as biggest disruption after "Internet" in the technology world. A quick reference to some statistics in Fig 1 below suggests apparently so.

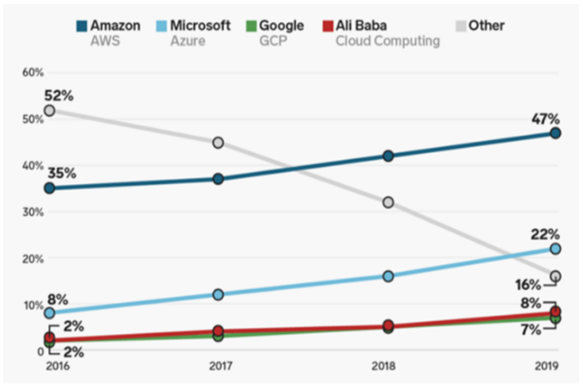


Fig 1: Public Cloud Market Share (*Source – Gartner, Goldman Sachs*)

According to Right Scale's annual State of the Cloud Report (2019) 94% of enterprises use cloud today, Gartner (2019) predicts that global public cloud service market will reach $354.6 billion by 2022 at an average growth rate of ~16% year-on-year, Forbes predict that 83% of enterprise workloads shall be deployed in cloud by 2020.

In simple terms, Cloud refers to a remote location where requested resources reside. These resources are used for a purpose on-demand, over internet, as and when needed by a consumer on agreed commercial terms with service provider. These service providers are known as Cloud Providers and there are some prominent players in the market namely AWS (Amazon), Azure (Microsoft) and GCP (Google). According to Gartner (2019) AWS, Azure, GCP and Alibaba share 84% of global market shares amongst them.

II. JOURNEY TO CLOUD

Organizations can no longer attain a competitive advantage by moving to cloud rather develop a parity ground. Broadly, cloud journey could be of two type – Migration or Modernization. Migration is a strategy where an organization leverages Infrastructure services (IaaS) offered by a Cloud Provider and shifts its application workloads to cloud Infrastructure instead of local one. This approach is also referred as Re-host. This is relatively a simpler approach with no code changes involved and Infra team alone from an organization should be able to perform this task most of the time. Applications moved using this approach would be called Cloud Infrastructure-Ready Applications.

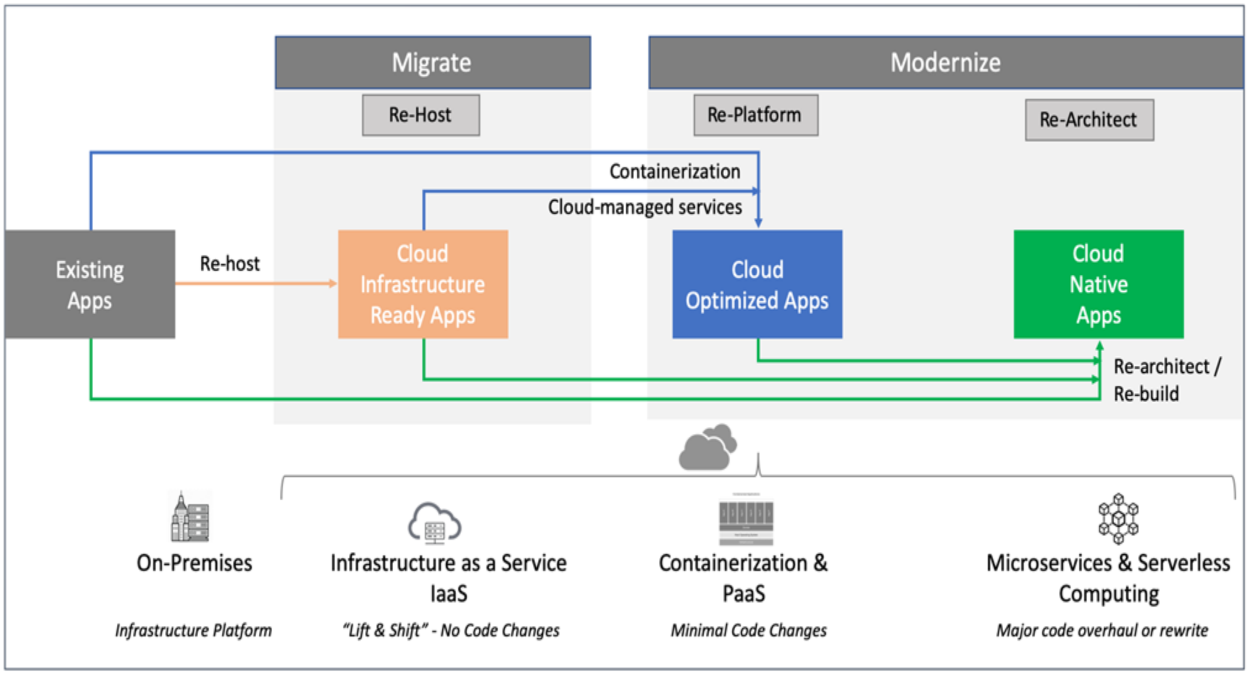
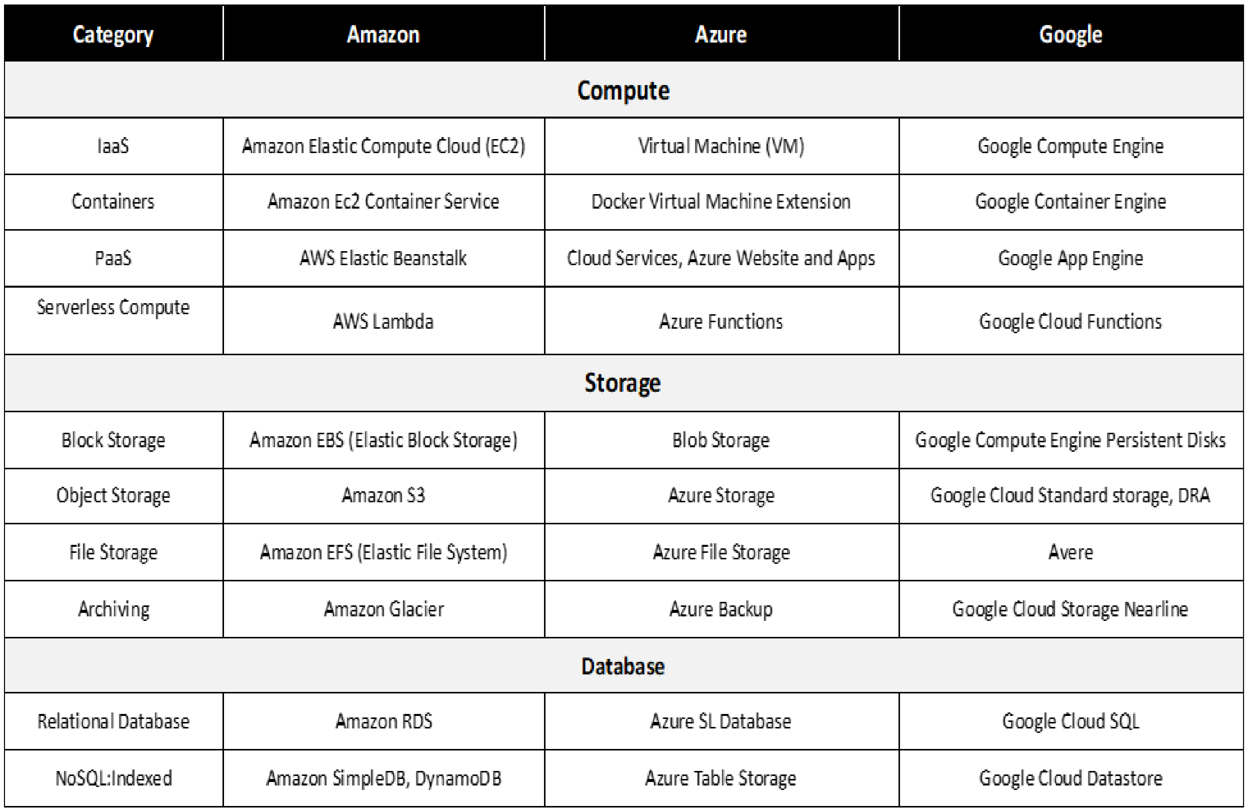


Fig 2: Cloud Journey Approaches - Migration and Modernization

Modernization, however, could be achieved using Re-Platform or Re-Architect choices. In Re-Platform, an organization modernizes its applications by using Platform services (PaaS) offered by a Cloud Provider or by containerizing them. Cloud PaaS services offer a vast range of choices like Managed Database Services, App Services etc. enabling organization to go lean on capex investments. Containers are a prime choice these days over virtual machines because of their simplicity and a suit of benefits offered. Containers are offered as PaaS services by all prominent cloud providers these days such as Amazon EKS, Azure AKS and Google GKE. With these services, organizations no longer need to own and maintain infrastructure required to run application workloads. Applications such modernized, would be called Cloud Optimized Applications. Re-Architect (aka Re-build) will be a chosen approach when existing applications are not tuned to offer required agility, performance and operational cost efficiency or in case when new business needs to be addressed which can be offered in cloud only. This is a resource-intensive and time-consuming approach of all and at the same time has to offer most of all other approaches.

III. CLOUD AGNOSTIC SOLUTIONS

Given that there are so many cloud offerings, an organization has to make choice wisely because this is a long-term engagement and there are enough lock-ins to make the exit quite difficult. For instance, an organization which is collecting GBs/TBs of data from thousands/millions of devices daily into AWS S3 bucket, cannot shift its operations to Azure/GCP Storage in a small span of time and without incurring significant cost. All Cloud providers are offering the similar services under different names with different implementations which makes the architect's job challenging to design a solution that can be deployed to any (or most) cloud. A quick look at Figure 3 below gives an insight into the challenges being discussed and when the other cloud providers like Alibaba, Adobe, VMWare, IBM Cloud, Rack Space, RedHat, Salesforce, Oracle Cloud are not even considered for the sake of argument. A single solution that support multiple cloud environment with no or minimal changes can be claimed as Cloud-agnostic solution. In today's time no true cloud-agnostic solution that can be achieved as such but still with some intelligent solution considerations and technology choices, multiple cloud environments can be supported.



Category	Amazon	Azure	Google
Compute			
IaaS	Amazon Elastic Compute Cloud (EC2)	Virtual Machine (VM)	Google Compute Engine
Containers	Amazon Ec2 Container Service	Docker Virtual Machine Extension	Google Container Engine
PaaS	AWS Elastic Beanstalk	Cloud Services, Azure Website and Apps	Google App Engine
Serverless Compute	AWS Lambda	Azure Functions	Google Cloud Functions
Storage			
Block Storage	Amazon EBS (Elastic Block Storage)	Blob Storage	Google Compute Engine Persistent Disks
Object Storage	Amazon S3	Azure Storage	Google Cloud Standard storage, DRA
File Storage	Amazon EFS (Elastic File System)	Azure File Storage	Avere
Archiving	Amazon Glacier	Azure Backup	Google Cloud Storage Nearline
Database			
Relational Database	Amazon RDS	Azure SL Database	Google Cloud SQL
NoSQL:Indexed	Amazon SimpleDB, DynamoDB	Azure Table Storage	Google Cloud Datastore

Fig 3: High Level Cloud Services Comparison

IV. CLOUD NATIVE APPLICATIONS

As an analogy, consider the realm of Mobile apps where applications were initially developed keeping in mind the operating systems such as Android and iOS and such apps were called as Native apps. Native, because they optimally harness the underlying platform capabilities. Then came hybrid apps which can work on multiple mobile operating systems but in doing so some native capabilities cannot be leveraged. Similarly Cloud Native applications are the application which are built and run to capitalize benefits of cloud computing model. The Cloud Native Foundation exists to promote best practices and community engagement. Their definition of Cloud Native is, "Cloud native computing uses an open source software stack to be:

- Microservices oriented. Applications are segmented into microservices. This significantly increases the overall agility and maintainability of applications."
- Containerized. Each part of the application (applications, processes, libraries, etc.) is packaged in its own container. This facilitates reproducibility, transparency, and resource isolation.
- Dynamically orchestrated. Containers are actively scheduled and managed to optimize resource utilization.

V. COST OPTIMIZED CLOUD NATIVE SOLUTION

Designing solutions is an art of balancing investments with returns while making right technology choices and meeting business requirements. Organizations choose for their cloud native journey keeping in mind the Low Capex investments, Agility to deliver business features, High Availability, Highly Scalable and Performant solution and this all must come at a minimal cost. This makes it paramount for designing the solution right in first place and hence bringing all the building blocks together calls for failproof architectural decisions.
Under Cloud Native landscape today, hundreds of different options are available under different categories required in designing a Cloud Native Solution. For Infrastructure Provisioning, Container

Runtime, Orchestration & Management, App definition and development there are many commercial and open source available.
Open source software development has immensely changed the technology industry and digital transformation paradigm e.g. Hadoop's genesis lies on the whitepapers published by Google, Kafka was developed by LinkedIn and then donated to Apache Software Foundation and there are many other similar examples. And it comes for free!! But availability of technical experts, community support, regular patches and updates is an upfront item in checklist which an organization must tick before finalizing an open source solution. Consider a scenario when for example, Cassandra/MongoDB/Postgres is chosen as an open source database solution and operations team need administrators to support issues in production, but organizations have access only to developers not administrators. PaaS services on the other hand are complete managed services where organizations need not worry about installation, low level configurations, updates, scaling and monitoring. With PaaS service most of the focus can be shifted on solving business problem instead of operations. But then these services come at a cost and create dependencies and vendor lock-ins.
Architects must perform a thorough assessment of these products and pick the appropriate tools keeping in mind cost, availability of community support, availability of skills in market. Architects have a sincere responsibility to choose a right mix of commercial and open source solutions keeping the challenges above discussed in mind along with business needs.

VI. RECOMMENDED CLOUD NATIVE SOLUTION

An enterprise level Cloud Native Application requires multiple aspects to be considered and make all the components work together like a well-oiled machinery to perform a task at all the times, with a desired performance, be ready to handle peak traffic hours and adaptable to continuous upgrades and extensions. In the following text, we attempt to describe our point of view for designing such a solution.

a. Infrastructure

Choose a public cloud such as Amazon, Azure, GCP or a private cloud such as openstack depending on the business requirements. For example, an e-commerce organization could go for a Public cloud option, but a banking organization would prefer a private cloud option. Choosing cloud managed infrastructure would be recommended as it would enable organizations to focus more on application management. Infrastructure provisioning could be automated using opensource tools like Chef, Puppet, Saltstack, Ansible, Terraform. While Chef, Puppet and Ansible are primarily used for configuration management, Salt and Terraform are used for infrastructure provisioning. Saltstack would be a preferable choice when multiple client machines need to be upgraded at once from a server using agent-based approach because terraform does not support client server architecture.

b. Containerization and Orchestration

While docker remains a hardly challenged player for container image creation there are many options available for container orchestration. If organization does not face challenge in hiring right talent in this area, Kubernetes is a great choice over other options like Mesos, PCF etc. given its adoption rate in industry and community support available. Its acceptance is so dominant that almost all cloud providers today provide managed Kubernetes services such as Elastic Kubernetes Service by Amazon, Azure Kubernetes Services by Azure, Google Kubernetes Engine by Google and others under the umbrella of PaaS services. However, these PaaS services come at cost, and a choice can be made to create and manage Kubernetes cluster yourself or take these services. If Kubernetes is chosen as an open source container orchestrator, Rancher could be an excellent choice for creating and managing Kubernetes clusters.

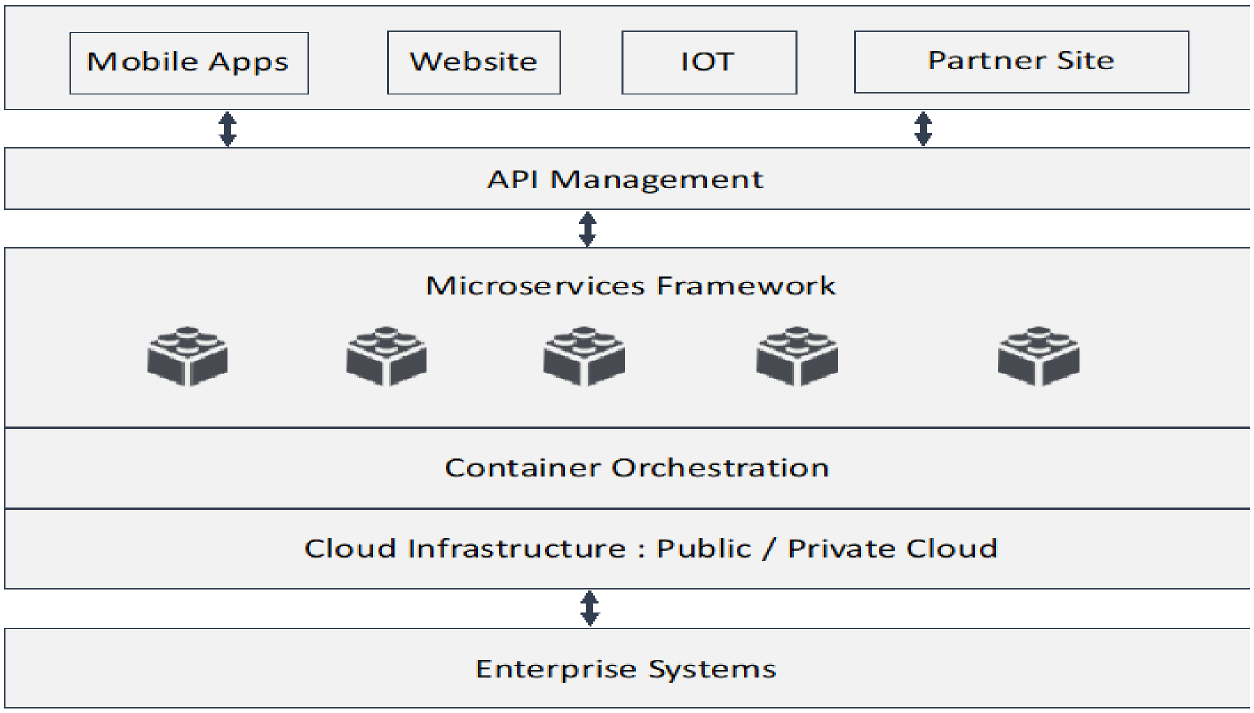


Fig 4: Conceptual Cloud Native Architecture

c. Microservices Framework

Organizations must choose microservices development language keeping in mind the comfort level with the choice and nature of problem being solved. Also, with the polyglot microservices it is no longer a constraint to develop application in a single language. It is on developers to comply with best practices for microservices development. Important aspect would be to provide a framework which enables the management of these microservices by providing services such as service discovery, dynamic routing, security, monitoring, logging, tracing, caching. Istio is a great open source, independent service mesh choice that provides above discussed fundamentals to successfully run a distributed microservices architecture. Rabbit MQ and Kafka are other open source choices for inter microservices communication.

d. API Management

Every microservices needs an API but not the other way around and in order to open communication channel outside, microservices are exposed via APIs. All cloud providers offer great API Gateway choice which provide features like SSL offloading, Analytics, Quota limit implementation, billing, rate plans, routing, caching, load balancer services. Apigee from Google, AWS API gateway, Azure API Management are such credible options. At the same time there are popular open source API gateway choices like Kong and Tyk.

e. Integration with Enterprise Systems

It is highly likely that organizations would like to integrate these cloud native solutions with their existing enterprise application such as CRM, OSS/BSS systems. Mule is a great such commercial option which not only offers integration but many other capabilities alongside. WSO2 and Apache Camel are well accepted when implementing an open source ESB for integration.

VII. CONCLUSION

In this paper we have discussed the rise of cloud computing, its swift dominance on industry, architectural challenges faced in building Cloud Native Solutions and our point of view on approaching this step by step. While the rise of open source technology has given choices for organization to cut on the costs, it always comes with its own set of challenges and there are times when PaaS services prove to be better choices. In our view, this will always be the case and it will be up to the designers to make best of all available options and approach the problem in a systematic way. Our recommended solution along with a conceptual architecture is a point of view only and does not solve a specific problem but it is applicable to specific business cases.

We believe that future belongs to cloud computing and with the more competitive offerings in this area, organizations will be at the beneficiary end. Solution designers will have to keep a close eye on various offerings and continue to evaluate and pick the best for their specific problems. Microservices based applications seem to be the appropriate harnessing of cloud computing at the moment, to be followed by serverless computing.

REFERNCE

[1]. Balalaie, A., Heydarnoori, A., Jamshidi, P., 2016. Migrating to cloud-native architec- tures using microservices: an experience report. Advances in Service-Oriented and Cloud Computing: Workshops of ESOCC 2015, Taormina, Italy, September 15-17, 2015, Revised Selected Papers. Springer International Publishing, pp. 201– 215. doi: 10.1007/978-3-319-33313-7_15.

[2]. Familiar, B., 2015. Service fabric. Microservices, IoT, and Azure: Leveraging DevOps and Microservice Architecture to Deliver SaaS Solutions. Apress, Berkeley, CA, pp. 165–182. doi: 10.1007/978-1-4842-1275-2_8.

[3]. Kratzke, N., Peinl, R., 2016. ClouNS - a cloud-native application reference model for enterprise architects. In: 2016 IEEE 20th International Enterprise Distributed Object Computing Workshop (EDOCW), pp. 1–10. doi:10.1109/EDOCW.2016. 7584353.

[4]. Martin Fowler, 2014. Microservices - a definition of this new architectural term. Last access 2016-05-27. http://martinfowler.com/articles/micro services.html

[5]. RightScale State of the Çloud Report (2019) from Flexera, https://resources.flexera.com/web/media /documents/rightscale-2019-state-of-the-cloud-report-from-flexera.pdf?elqTrackId=372b6798c7 294392833def6ec8f62c5c&elqaid=4588&elqat= 2&_ga=2.75255952.381106268.1556868633-1445624021.1556868633

[6]. CNCF Cloud Native Interactive Landscape, https://landscape.cncf.io/

[7]. IBM (2020). Cloud_learnhub_Cloud-Native-A-Complete-Guide. Retrieved from https://www. ibm.com/cloud/learn/cloud-native

[8]. RedHat (2020). Understanding cloud-native apps. Retrieved from https://www.redhat.com/en/ topics/cloud-native-apps

[9]. Microsoft Azure Documentation, Start your cloud migration process https://azure.microsoft. com/en-us/migration/migration-journey/#migrate