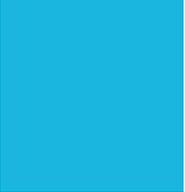




CASE STUDY FOR
CLOUD.BLUE TRANSFORMATION

Proven Cloud Transformation methodology from Blue Guava



CONTENTS

Introduction	3
Build your own	3
Virtualization	4
Cloud	4
AWS Cloud	5
Today Trends	6
Background	7
About Us	7
Perspectives	7
Challenges	9
Infrastructure	9
Development	9
Security	9
Solutions	10
Define the Goals	10
Architecture Consideration	10
Operation Excellence	13
Performance Efficiency	14
Cost Optimization	14
The Transition	14
About Blue Guava	15

Cloud Transformation is an evolution of infrastructure management. Many ingredients for success already existed before. It is a huge opportunity for any organization to go in a direction in simplicity vs. the very high available on a reliable cost.

A short historical overview of the computing business from the last two decades.

BUILD YOUR OWN

So many years ago, everybody built their server or server farm or maybe their owned datacenter. Why? Because this is the only way you want to run an application and control your way.

The limitation was the server's size and time, and effort to build and had a limitation of the scale-up. If you have only a few servers, then it was not easy to buy another one. Have to wait a week to be able to put it into production. And also, you have no way to take care of redundancy. You have only a way to buy more physical servers and set up everything by us.

PROS

- Everything under your controls

CONS

- Complicated to build a High Availability Environment
- Expensive because of the physical items
Complicated maintenance
- More outage expected



VIRTUALIZATION

The Virtualization brought the option to the engineering to run multiple OS on the same machine. It got some HA options to the maintenance team, but setting up the excellent balance and efficiently maintaining the environment was also challenging.

This step started a trend in the IT business and opened millions of options to the IT staff.

PROS	CONS
<ul style="list-style-type: none">• Everything under your controls• Optimize the server size• Isolate the services• HA option was available• Option to automate processes with CLI	<ul style="list-style-type: none">• Overprovisioning the infrastructure• The sophisticated configuration needed for the high performance• Many software adoes not tolerate well when the HA function was active. For example, SQL Server performance degraded when the vmMotion was in action• Many HA settings needed to consider where will be placed on the guest or the host level• Overall management still required a qualified engineer to set up and maintain the hardware and software

CLOUD

The cloud brought a new concept called IaaS (Infrastructure as a Service). It means that we need a server or service that was still on our hand to decide to build it, but via a web-based wizard, which started to simplify the provisioning, scale, and load balancing. We did not need to buy expensive HW; we could use the cloud provider predefined Virtual machine based on our requirements.

PROS	CONS
<ul style="list-style-type: none">• Optimize the server size• Isolate the services• HA option was available easier like before an easier way to manage it• Infrastructure as a Code to automate the scale up• Cost-Efficiency	<ul style="list-style-type: none">• Lost the control over the HW• Possible to pay more because of the wrong usage

AWS CLOUD

Like to mention AWS in a separate section. The specialty of AWS is: doing the cloud on a **professional** level.

Concept:

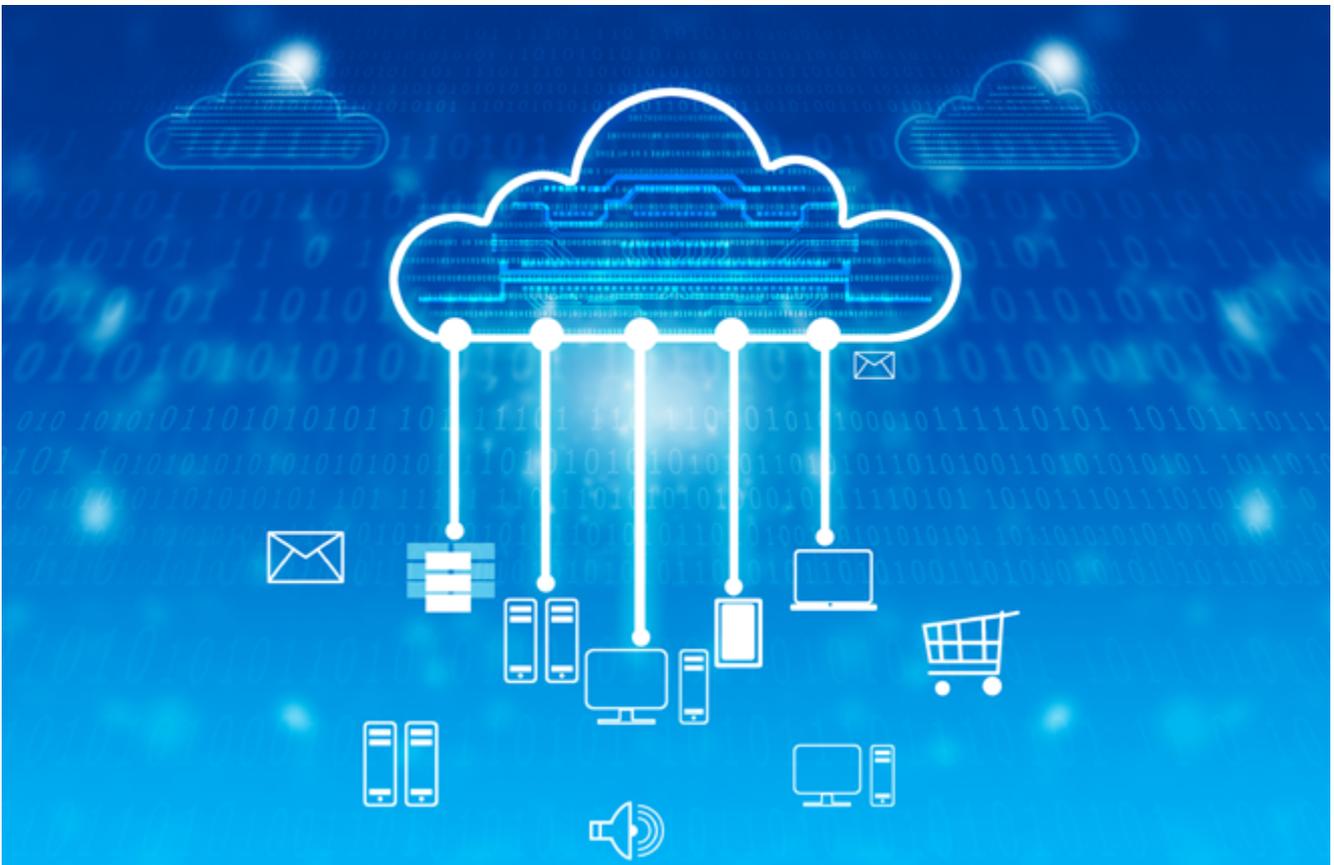
1. Add more HA to the infra layout.
2. Automate the whole ecosystem.
3. Provide a straightforward way to manage it.
4. Use manage service where AWS takes the underlying management and configuration.

PROS	CONS
<ul style="list-style-type: none">• Trade upfront expense for variable expense• Stop spending money to run and maintain data centers• Stop guessing capacity• Benefit from massive economies of scale• Increase speed and agility• Go global in minutes• Fault-tolerant option almost every place• HA by default• The ecosystem approach from infra direction• No need for an IT team to set up everything• Many scalable, HA technology exists there which has never existed on the on-premise way	<ul style="list-style-type: none">• Build by purpose!• Use what you have to use!• Requires more maintenance at the beginning

TODAY TRENDS

CIOs, CTOs and the organizations they work for will face increasing challenges to not only remain competitive in this ever-changing cloud computing environment, but ensure they stay on the right side of both existing regulations and new ones as and when they emerge. It's safe to say that the next decade of cloud computing will be just as eventful as the last.

1. The Number of Cloud-based Services and Solutions Will Continue to Rise
2. Hybrid and Multi-Cloud Strategies Will Play a Key Role
3. Security and Compliance Will Be Critical



ABOUT US

Blue Guava engineers spent more than 20 years in the IT business.

Managed 300+ on-demand and cloud servers across the globe.

Supported more large audience events on-premise and also in the cloud.

Deployed physical and virtual servers around 20 neutral data centers across the Globe (Asia, Europe, United States).

Create LAN and WAN direct connection via L2 / L3 lines.

Worked with all major CDN providers.

Build Multi Tiers Custom CDN in Europe (Origin, HDD, SSD, NVMe, CDN federation).

HA and Fault-Tolerant SQL Servers.

PERSPECTIVES

The Business, People, and Governance Perspectives focus on business capabilities, whereas the Platform, Security, and Operations Perspectives focus on technical capabilities.

1. Business

The Business Perspective helps you move from different business and IT strategies to a business model that integrates IT strategy. Agile IT strategies are aligned to support your business outcomes, and they can adjust to business needs or technical capabilities as they change.

2. People

The People Perspective helps Human Resources (HR) and personnel management prepare their teams for cloud adoption by updating staff skills and organizational processes to include cloud-based competencies.

3. Governance

The Governance Perspective integrates IT Governance and Organizational Governance. It guides on identifying and implementing best practices for IT Governance and on supporting business processes with technology.

4. Platform

The Platform Perspective helps you design, implement, and optimize AWS technology architecture based on business goals and objectives. It helps provide strategic guidance for the design, principles, tools, and policies you will use to define AWS infrastructure. The Platform perspective also includes principles and patterns for communicating your target state environment, implementing new solutions on the cloud, and migrating on-premises workloads to the cloud.

5. Security

The Security Perspective helps you structure the selection and implementation of controls. Following this guidance can make it easier to identify areas of non-compliance and plan ongoing security initiatives.

6. Operations

The Operations Perspective helps you run, use, operate and recover IT workloads to level that meet your business stakeholders' requirements. Insights gained through the Operations Perspective define your current operating procedures and process changes and training needed for successful cloud adoption. Well-managed IT operations support the operations of the business from planning and sustaining through change and incident management. It guides on identifying and implementing best practices for IT Governance and on supporting business processes with technology.



INFRASTRUCTURE

- They have to move from the Infra Engineer direction to the DevOps.
- Many configurations managed differently from how they worked before on the on-premise world.
- The network configuration and network isolation is the first task in every project—no way to change it without shutting down the system.
- No access to the hardware, no access to the OS.
- Many services have on the way to login to the server and check what's going on.
- Need to learn how to read the logs.
- The whole system could be more distributed, harder to find which components failed.
- The services communicate asynchronously.
- Maintenance jobs are mostly a bunch of automation tasks.
- Have to use the Terraform/CloudFormation to build an infrastructure. No click Next, Next in the future.
- Have to pay for almost everything in the cloud. Consider the services size and instance numbers.
- Do a cost analysis and set up the thresholds.
- Understand the scale-up strategy.

DEVELOPMENT

For cloud development, it is not required to learn a programming language but has to learn new techniques to approach the problem.

- Asynchronous communication.
- Suppose you have a monolithic architecture, then worth considering how to decoupling the system. For example, frontend and backend if the architecture supports it.
- Consider the Logging.
- Debugging in the Cloud is requiring more analytics skills.
- Optimize your code as you can.
- How to run a service in multiple instances.

SECURITY

Security is a new task for many organizations. The right cloud settings require the right roles and security configuration for a cloud account.

DEFINE THE GOALS

Ready to Go

The Go-to-market steps will require architectural changes on the code and database level also.

We would recommend a minimum of a short CI/CD pipeline to deploy what we improve in the future!

Maintainability

Create a Dev, QA, and Stage Environment for the testing.

When the environment is ready, we have to extend a pipeline.

Improve the CI/CD pipeline for the development.

Scalability

Allow the environment to scale together with the demand.

As a second step, we make the environment available for the go-to-market and enable room for development and configuration.

Sustainability

Set up the monitoring of the system from multiple aspects.

ARCHITECTURE CONSIDERATION

For the AWS scalability, we recommend making some changes on the current code side to support it.

The plan here:

1. Scalable Environment
2. Identified the bottlenecks
3. Understanding the current code or business logic limitations
4. Add more automation.

Elasticity Concept

When we decide to go in the manage-services direction and want to configure autoscale and global presents to the application, we have to consider a few things.

- Configuration: How to configure my application? What types of the parameter needs to set up for the right environment.
- How heavy is my database usage, how to optimize it?
Avoid heavy write database operation. Please build a buffering approach for tracking purposes where the data is going to a queue and process it later. This technique can drain the massive traffic and process the data with expected performance.
- What is the known bottleneck of the system, or what could it be?
For example: database limitation, database size, heavy queries, etc. Optimize the Query performance with the right JOIN operator and try to cache the query result.
- Application configuration strongly recommended being managed from the outside, not from the inside. Please take a look at the Parameter Store Service from AWS.
<https://aws.amazon.com/blogs/mt/the-right-way-to-store-secrets-using-parameter-store/>
- Debugging Strategy for the Environment.
- How do you plan to monitor the system?
- IAM roles to access the right services!
- Suppose you have a monolithic architecture, then worth considering how to decoupling the system. For example, frontend and backend if the architecture supports it.

Use AWS CDN

The S3 is the perfect place to store and manage files in AWS. S3 is also good to use the S3 for a distributed file system between the services/instances.

If we plan to publish it on the website, then not recommend using the direct S3 access because the CloudFront allows us to cache the content on the AWS CDN and reduce the latency!

Next step: Have to review the code and change the URL from the internal S3 path to a CloudFront supported way!

More information:

<https://aws.amazon.com/cloudfront/>

Debugging and Logging

An important point to understand the logging option in AWS.

The CloudWatch is by default an option for anyone but has to consider how to use the log message; otherwise, the log will not be usable!

The X-Ray helps developers analyze and debug production, distributed application, and improve performance bottlenecks. It is optional and future-ready; no necessary to use it now!

More information:

<https://aws.amazon.com/cloudwatch/>

<https://aws.amazon.com/xray/>

Database Optimization

We have to think of a more cloud-optimized database server to improve the scalability.

Under the AWS umbrella, you have more options. We would like to highlight two from the wide range.

AWS Aurora

Recommend to use Aurora service, which is a combination of MySQL and Postgres. It is a managed service from AWS! The migration cost for this is lower but provides multiple benefits also.

HIGHLIGHTS

Performance and cost	High Availability	Compatibility
Get 5X the throughput of standard MySQL and 3X the throughput of standard PostgreSQL. This performance is on par with commercial databases, at 1/10th the cost.	Amazon Aurora is designed to offer greater than 99.99% availability, replicating six copies of your data across 3 Availability Zones and backing up your data continuously to Amazon.	MySQL and PostgreSQL Compatible.

More info:

<https://aws.amazon.com/rds/aurora/>

How to connect to Aurora Cluster:

<https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/Aurora.Connecting.html>

Why Aurora?

Aurora's storage is fault-tolerant by design.

Aurora's performance is higher and more consistent.

Failover time is faster on Aurora!

You can scale your read queries on Aurora by creating 15 read replicas, whereas 5 in regular RDS.

DynamoDb

It is a NoSQL database with many advantages! One of them is **multiple WRITE nodes and high-speed replication between the datacenter.**

Going in this direction could be beneficial for many prospective, but it requires a more significant architectural change and understanding of the Dynamo advantage and also the disadvantages.

Recommend if need to distribute the database traffic between data centers and the business logic designed for NoSQL logic.

More information:

<https://aws.amazon.com/dynamodb/>

CI/CD pipeline

Create and implement release pipelines for deployment and management. We recommend a simple PROD flow for the launch and an DEV, QA, STG, and PROD stages for the future.

OPERATION EXCELLENCE

First Step

If we want to go in this direction, we need an AWS master account to set up the environment!

Monitoring

For monitoring purposes, we can offer two things.

Regular monitoring via the AWS Dashboard System - covers the physical parameters of the environment like CPU, Memory, and Request/seconds.

The second is primarily a service KPI metrics-based where you have to monitor the businessn KPI.

Recommend to take a look at the Prometheus/Grafana Integration.

More information:

<https://aws.amazon.com/prometheus/>

<https://aws.amazon.com/grafana/> (preview phase)

PERFORMANCE EFFICIENCY

Choose the right configuration size

Need to define or measure what could be the ideal size of the containers. Based on the forecast, we have to choose the right container size in AWS.

Scale-Up/Down Strategy

All applications have their behavior, and AWS also could not scale immediately. Have to configure the right number for the container. These settings save a cost and also provide the same performance during the various loads.

COST OPTIMIZATION

Cost Structure - Tagging

AWS can categorize the cost based on businesses. It means the price can divide into four domain names. It called tagging,

Organization

AWS also supports the structure of the services and even in the organization. It means the four different domain names can use the same architecture but additional services. This will include multiple times the cost and allow you to increase the load and demand per business and not everything together.

Cost Monitoring

Optional. AWS also supports many cost exploring options. We recommend configuring the AWS Budget, which allows the business to notify when the AWS cost reached the limit of the pre-configured threshold level.

THE TRANSITION

Review

Well-architected review of completed environments and services with remediations.

Documentation

Documentation for infrastructure, Terraform templates, and pipelines.

Training

Handoff and training sessions for stakeholders.

Support

Any post-project support.

Our goal is to become the best long-term partner that any of our clients could wish for. With more than ten years of state-of-the-art software development, streaming, and testing solutions, we have helped market-leader partners increase their revenue and the efficiency of their IT operations while cutting costs and time. Simultaneously, the software products we developed for them streamlined and optimized the streaming experience for millions of their customers across more than 50 countries on three continents.

At Blue Guava, we believe in exceptional customer service. Our passion is to provide our clients with nothing but the highest quality services that are guaranteed to meet their needs and help them in their quest to produce excellent software solutions. Our content delivery, content management software solutions, and quality assurance services will help you maximize customer engagement, ultimately empowering your business's customer adoption and retention capabilities.

[Contact Us](#)