



WHITE PAPER



Cloud Migration Checklist:

how to successfully move to the Cloud

The cloud is one of the key digital transformation trends at the moment, utilized as an effective tool to leverage for projects that aim for highperformance, scalability, availability, and global presence. One of the most important value propositions in the cloud is the ability to transform your CAPEX to OPEX, freeing up budget space and increasing growth and development capabilities, even more so when utilized fully. We have prepared this guide to offer you vital information to help you dig into your business's untapped technological potential.

First, we will discuss one of the newest trends in the industry: Cloud Native. After that, we will explore a few scenarios using the cloud migration checklist (we encourage you to try it when modeling your cloud journey!).



| Contents

<u>Cloud Native</u>	3.
<u>Cloud Migration Checklist</u>	4.
<u>Cases</u>	5.
<u>Case #1: Simplifying Global Expansion</u>	6.
<u>Case #2: Improving Online Player Experience</u>	7.
<u>Case #3: Accelerating Streaming Services</u>	8.
<u>Cloud Migration for Businesses</u>	10.

| Cloud Native

In a nutshell, being cloud native is the concept of building, deploying, and managing applications that are adapted or even designed specifically for the cloud from their inception onwards. Companies who start their businesses in the cloud are called Cloud Native companies. These companies want to have highly scalable, flexible, and resilient applications that are able to answer both the expected and unexpected needs and challenges of customers (i.e., endure high loads or mitigate security problems smoothly).

The cloud gives you:

- Increased resilience and agility of your business. This means that you can use resources when you need them and deallocate when you don't.
- Optimized costs via moving to the pay-as-you-go model. By using the budgeting practices that are emerging as an answer to this type of cloud model, you can use as many resources as you need when you need them!
- Protection for your infrastructure, innovations, and data. Cloud platforms are optimal adopters of all possible best practices, compliance frameworks, and tools that are available on the market, including a team that is available to advise you specifically about security and migration.
- The ability to focus on innovations instead of investing resources both upfront and continuously to keep flexible infrastructure turned on.
- New business models and leveraging of application innovation practices.
- Decreased risks, whether compliance related, geographical, political, or otherwise.

In this guide, we focus on the migration of an existing project to the cloud, which can be a challenge. For example, how do you approach migrating a legacy project that was written decades ago? Or a project that was created more recently, but was deployed to a local hosting or on-premise infrastructure? What steps should we take in order to make the migration process smoother?

Migration to the cloud is essentially a process of moving your business apps and data from one location to a cloud-hosting platform.

This migration can be done in various ways:

- Full migration: We move all systems to the cloud platform.
- Partial: When you move one service or a bulk of data as an extension (in which you use the cloud as an extension to your datacenter, taking resources whenever they are needed). For example, partial migration may be used when you need resources for testing an application or during peak times.
- Using the cloud platform as a reserve site, in case of incidents at the current location, be it the on-prem, hosting, or even cloud. The system switches to a reserve site and continues to serve customer requests.

To sum up, if your company is looking for easy global scalability and simplified infrastructural management, then the cloud is worth exploring! Even if you have an old application that was written decades ago, there are ways to migrate it step by step, slowly, looking at how the application behaves in this hybrid model.

| Cloud Migration Checklist

Any solid decision, and the decision to migrate is a solid one indeed, should be thoroughly planned and executed. We have prepared this checklist and will go through a few scenarios in this document.

These scenarios are anonymized, yet based on real case studies we have worked on.

Please be advised that the way you use this checklist will depend heavily on why you're looking to assess cloud migration options.

Essentially, there are two types of triggers: planned and unplanned. Planned means that you as a business have some time to prepare. These are the kinds of scenarios we'll explore below. We won't go over the unplanned type, mainly to avoid going too deep into details. However, we are working on an article that will go into the logic behind processes during unplanned situations.

1. SITUATION ASSESSMENT:

Do an assessment of the application's ecosystem and business value. This stage requires a general assessment of the business from a higher level.

2. PROBLEM:

What problem do you want to solve by migrating? We need to choose carefully, particularly to avoid a situation in which one major issue is mitigated, but a new one is created. For example, if we want to increase the system's fault tolerance by introducing a new reserve site, do we understand how it will impact latency?

3. NEED-PAYOFF:

If we choose fault tolerance over a slight decrease of system speed, how will it impact your business KPIs? Some systems can cope with seconds of added latency, but if we have a delivery where each added second means losing customers, we need to acknowledge that and work on solving it another way.

4. PLANNING:

Each migratory process starts with a plan. This involves taking into consideration feedback from all parties involved in the business processes of the company. The plan should cover not just technical issues but also legal aspects, etc. This is especially important if our aim is, for example, to comply with external data laws. At this stage, we do a detailed assessment of our architecture components (applications, databases, storage, network, etc.), choose the type of migration, choose the specific cloud model (IaaS, PaaS, SaaS), map our architecture services to the cloud services model, and create a document that outlining the migration team as well as roles and responsibilities.

5. PREPARATION:

Gather our team and tools. Synchronize our plan with the cloud vendor's team.

6. TESTING AND PILOTING:

It is always a good idea to test the “light” version of your system in the cloud. Before piloting, it is important to agree on the most important technical metrics that the team should measure or monitor.

The list below is not exhaustive, as some businesses have their own specific metrics, but can be used as a baseline:

Application behavior: This includes the initial adaptation to cloud monitoring instruments, as well as understanding how the application behaves under heavy loads. Some applications (like SAP) have their own deployment and licensing rules that should be taken into consideration while deploying to the cloud. Checking with the cloud vendor before deploying is a good idea.

Latency: This includes the overall latency, as well as more detailed metrics like the latency between internal system components (e.g., between the database and frontend). The average latency can be calculated over time.

Throughput: This includes overall throughput between the cloud and end customer, as well as internal throughput between system components. It is not limited to the network, as cloud storage disks and database services often provide different tiers with different throughput numbers.

7. ESTIMATION:

The cloud is dynamic by nature. Moving from the CAPEX to OPEX model means that if you've ever calculated resources in this model, there is a chance of unexpected bills.

To avoid this, plan a pilot deployment for at least a few weeks to understand how your system behaves in the cloud and what costs are incurred.

8. IMPLEMENTATION AND MIGRATION:

This stage is where the migration actually happens.

Let's take a look at how to apply this checklist to specific scenarios. We brainstormed and chose a few situations that we think are applicable.

Cases →

I Case #1: Simplifying Global Expansion

Situation: There is a company called SpeechEngine. SpeechEngine is working on a speech recognition service, providing it in a SaaS model with the GPUs on the backend. SpeechEngine is a startup, so they have limited human resources. They have just received investments from venture investors for their global expansion plans. Currently, they provide their services in the US, but plan on targeting European countries.

Problem: SpeechEngine's goal is to launch a new region. After a thorough assessment, the internal team decided to launch their second region in the cloud datacenter in Europe.

Needed payoff: Launching the solution in the cloud allows SpeechEngine to quickly test their hypothesis and delegate a subset of legal questions regarding running the business in other jurisdictions. Without capital investments, SpeechEngine gets a reliable path to a global expansion and quick way to understand if the solution is ready to be launched globally.

Planning: Architecture and migration teams met with business, legal, and operational teams to understand how launching a new region will affect different aspects of the business. During the meeting, it became clear that the question was much more complex than just taking what SpeechEngine created in their original region and replicating it.

A few concerns were raised:

- The operations teams worried about how to update both regions at the same time.
- Architects were concerned about how this move would affect the performance of the project, since the team decided to maintain core functionality in its original region, keeping proprietary technologies closer to the team.
- Both the business and legal teams were interested in how to approach legal questions, such as hosting and processing personal data.

The dedicated migration team took notes and started the next planning phase.

As core functionality was left “as is” in the original region, the remaining components left to migrate were: storage, website, and useful PaaS functionality to assess in the cloud like a load balancer. Based on this, the team decided to take a PaaS approach and deploy the website to Kubernetes, as well as storage and the database to corresponding PaaS services.

Preparation. Choosing PaaS services minimized the deployment team's human resource needs and reduced the number of tasks to complete. The SpeechEngine team had the experience of using Kubernetes. The challenge was moving data to S3 storage and DBaaS. Solution Architects from Gcore helped to prepare a migration plan, and advised how to use cloud services. They also helped create a mapping between self-managed tools and cloud services, including data migration from the local database to DBaaS.

Testing and piloting. SpeechEngine decided to allow only a subset of features to be migrated to the new region in order to understand how it would impact core functionality that was left in the US datacenter. The testing period lasted for 1 month. During the piloting, the R&D team identified new opportunities. The Cloud platform had Graphcore, an innovative hardware specialized in AI workload. Graphcore is a powerful platform that provides faster time to results. CPUs and GPUs aren't always the best choice for some tasks like high-accuracy computer vision, natural language processing, or graph ML. Intelligence Processing Unit (IPU) systems from Graphcore, designed from the ground up for AI, looked like an interesting option to explore. The R&D team notified the planning team.

Implementation. SpeechEngine successfully launched the project in the secondary region, minimizing the financial risks of launching in a brand new region. After the R&D team notified the planning team about Graphcore hardware, it was decided to leverage this for a subset of premium customers, adding a brand new business model for the company.

The vendor's technical team was helpful by following up and consulting the company during the migration process.

I Case #2: Improving Online Player Experience

Situation: The company is called GlobalGames. They already have their own global infrastructure that serves online gaming services. GlobalGames's goal is to expand their business and scale smoothly, as well as to cut costs by moving some of their resources to a cloud provider. It should also reduce developer overhead so they can allocate their time to more important business tasks. Running the system on their own infrastructure was enough until GlobalGames entered a phase of rapid growth that required a change in their approach to infrastructure. This growth has been driven by the pandemic, which has given a significant boost to the gaming industry.

Problem: Online game players demand seamless, low-latency performance, as well as the best possible experience, and that requires something that stands out. GlobalGames's goal was to launch two new regions: in Europe and Asia (Tokyo, specifically). However, there were several challenges. Firstly, there was a need for fast deployment to the Asian market, taking into consideration language barriers and complex legal and bureaucratic laws. Secondly, they needed to ensure the low latency and performance that are required to deliver a seamless digital gaming experience.

Needed payoff: GlobalGames, as a company with extensive experience in running global projects, understood that moving even part of its infrastructure should be accompanied by thorough testing, piloting, and analysis. This means that the team would need to invest a lot of time and money into this process. However, their momentum is strong. According to Statista, worldwide revenues from cloud gaming amounted to \$1.48 billion in 2021, and are expected to rise to around \$6.3 billion in 2024. Making slower progress is much more risky, as there are always competitors waiting to take advantage of delays.

Planning: The planning team gathered all of the stakeholders, including legal and analytics departments, to recap the current state of business and decide how to move forward with migrating their project to new regions. Fortunately, from discussions with the cloud vendor team, it was clear that legal questions would be addressed by the compliance certificates.

Architects were encouraged by the new challenges of introducing PaaS services into their architecture, as the current GlobalGames architecture was composed of mostly VMs and IaaS services.

Mitigating legal risks allowed the team to focus on the technological side of the discussion, and they decided that the company would replicate the infrastructure using the Infrastructure-as-a-Code approach while also introducing a few PaaS services to the architecture, like DDoS protection, and testing the newest AI technologies that were in a backlog for some months.

Preparation. During the preparation stage, GlobalGames asked Gcore to support them through architectural consultation as well, to understand how to introduce new services and leverage the latest application modernization practices to extract as much as possible from PaaS services.

Testing and piloting. GlobalGames chose an unusual way to test and pilot the new regions. The team decided to launch two versions of their software. 80% of their customers used the original version of their system, while the remaining 20%, the experimental group, used the service with added PaaS services and AI-as-a-Service.

Implementation. After the migration and launch of the new version of the platform, GlobalGames gained unique insight. Firstly, the team launched two new regions in less than two months. Secondly, during the migration, a completely new business model was identified and the architectural team was successful in introducing services that, in the end, optimized both the new and old version of the project, all while maintaining their low latency and high performance and scalability.

I Case #3: Accelerating Streaming Services

Situation: MegaSolutions has been on the market since 2000. MegaSolutions's core service is to provide productivity solutions for event organizers. Since 2000, MegaSolutions has had a rich history of exploring optimization, and now they provide services for virtual and hybrid events.

As with many other companies around the world, MegaSolutions had a high load spike during the very beginning of the COVID-19 pandemic. Before COVID, MegaSolutions customers were content with the infrastructure and private cloud deployments, but the new situation demanded a new approach.

MegaSolutions sought to complete a platform-as-a-service migration to cope with thousands of new users.

Problem: The major challenge here was that the MegaSolutions system was always a single-tenant solution. There was no multi-tenant solution, since there had never been major demand for one. To offer a reliable service to thousands of new users, MegaSolutions needed to re-architect their solution, test it, and understand the cost-effectiveness of migrating to the cloud.

Needed payoff: MegaSolutions's C-level team had a clear understanding that they were lagging behind the needs of their potential customer base. They were ready to invest human hours and money into rapid research and development to learn how they could start providing services to new customers.

Planning: The planning team consisted mainly of technical specialists, engineers, and technical support. **Here were a few of the questions raised:**

- What will the delivery model be? Should it be IaaS or PaaS?
- Are there SaaS services that can help with the event organization process?
- How can we optimize the project's functionality in new regions while keeping the high degree of reliability needed to plan and run events seamlessly?

During the architecture brainstorm with the Gcore solution architects team, it was decided that the product's core would continue to use a big cluster of virtual machines and work slowly on moving from the old codebase to new, with the help of a partner.

There was one task that looked small, but was actually critical. The MegaSolutions project used a streaming function that worked smoothly, but the team did not test how it would work on a global level. While deploying the project to the VMs did not look difficult, streaming posed a real challenge. Fortunately, Gcore has Streaming as a Service, a very convenient solution for the MegaSolutions developers.

Preparation. The vendor team was helpful in understanding the preparation and migration process. During the preparation stage, MegaSolutions asked vendors to support them with architectural consultation as well.

Testing and piloting. During the testing and piloting process, MegaSolutions had a major challenge in figuring out how the streaming pricing model would affect their financial situation. The C-level team dedicated additional budgetary resources to have their life-long event organizing partners to put together a few test events.

Implementation. MegaSolutions was able to go global without huge upfront investments. As they had a lot of legacy code, they decided to migrate on a step-by-step basis. The major challenge was to deploy a reliable streaming service, a challenge that Gcore's product easily solved. With the help of their partners, the necessary tests for the migration process, including budget and stabilization, were completed.

This is a quite common situation when moving to the cloud or extending resources, especially if you are a startup.

The cloud allows for faster migration.

- **Choose the type of migration (lift-and-shift, partial, extending to the cloud)**

During this step, it is important to understand what you want to achieve via migration.

Lift and shift can be difficult in cases where there are specific requirements, such as hardware replacement. However, with a successful migration, you will open up virtually unlimited scalability and resources.

Partial migration is much easier, as we move only some parts of the solution towards cloud management. In this case, it could be a website's functioning, or storage solutions that allow the company's architecture to maintain innovative algorithms. Keeping storage in a host country allows it to comply with local laws. Additionally, in this scenario, migration to the cloud will be instrumental in mitigating load spikes.

- **Choose the cloud model (IaaS, PaaS) and create a mapping between your project components and cloud services**

Choosing a specific cloud model is vital for a future integrated with cloud-managed service. Typically, company architects understand how the system will work in different cloud models, but you can also contact your cloud vendor, as they may have specialized teams built just for this purpose.

- **Synchronize the plan and be in contact with the cloud vendor team**

Constant communication is important here: how will you effectively use cloud resources in a way that produces successful outcomes? In this sense, it is best to relay your plan to your cloud vendor team and ensure that it is optimal and will produce good outcomes. This can help not only in terms of successful migration, but by preventing wasted resources too.

- **Do an assessment of your applications, databases, and storage, and refactor if needed**

Does your application need to be migrated in the way you are trying? By reanalyzing your applications, databases, and storage, you are able to ensure that your initial target is still fulfilled, and that you haven't strayed too far from the goal set out. Often, refactoring is a part of the initial process that can ensure optimal migration techniques, particularly when partial migration is carried out, to ensure increased productivity.

- **Create a migration team and define roles and responsibilities**

Without a team dedicated to successful migration, the process risks being put on the back burner. However, as a big step in moving towards the future, dedicated and experienced team members should ensure that the process is a priority.

- **An e-commerce company wants to move from a monolith application to being cloud native and achieving hyper-growth without sacrificing their high degree of security and compliance quality.**

Cloud Migration for Businesses

1. SITUATION ASSESSMENT

- Set goals
- What do we want to achieve with the migration?

2. PROBLEM STATEMENT

- Determine the problem
- What problem do we want to solve?

3. NEED-PAYOFF

- Map results and risks
- If we choose X over a slight decrease of Y, how will it impact our business?

4. MIGRATION PLAN

- Create a migration roadmap
- Choose the type of the migration
- Choose the specific cloud model
- Create a migration team and define specific roles and responsibilities

5. MIGRATION PREPARATION

- Allocate resources and delegate tasks to team members
- Synchronize the migration plan with the cloud vendor team

6. PILOT

- Understand system behavior and gather data
- Agree on the most important technical metrics

7. ESTIMATE

- Making an estimation based on the pilot

8. MIGRATION