Step 1: Establish the migration-architect role

Before you begin your cloud migration, establish the migration architect role to lead the effort. The migration architect is a system architect-level position responsible for planning and completing all aspects of the migration; their core responsibility should include defining necessary refactoring required to make the migration successful, designing strategies for data migration, defining cloud-solution requirements, and determining migration priorities and production switchover mechanisms.

During a large migration project, there are many decisions and technical plans that must be made and having a migration architect who is responsible for all aspects of the migration is critical to the success of the project.

Step 4: Cloud KPIs

The best KPIs for a cloud migration show how you migration is doing, illuminating visible or invisible problems that may be lurking within your application. Most important, perhaps, cloud migration KPIs can help you determine when the migration is complete and successful.

|  |  |
| --- | --- |
| Category | KPIs |
| User experience | Page load time |
|  | Lag |
|  | Response time |
|  | Session duration |
| Application performance | Error rates |
|  | Throughput |
|  | Availability |
|  | Apdex |
| Infrastructure | CPU usage % |
|  | Disk performance |
|  | Memory usage |
|  | Network throughput |
| Business engagement | Cart adds |
|  | Conversions and conversion % |
|  | Engagement rates |

Step 5: Establish performance baselines

Baselining is the process of measuring the current (pre-migration) performance of your applications to determine if its future (post-migration) performance is acceptable. Baselines help you determine when your migration is complete and provide validation of the post-migration performance improvements you expected. You can also refer to baselines during a cloud migration to diagnose any problems that arise.

Set a baseline metric for each KPI that you have decided to measure. Determine how long you will collect data to determine the baseline. Choosing a short baseline period (such as a day) lets you move faster, but you risk not collecting a representative performance sample.

You also need to determine if you want to collect only baseline data that’s average or representational, or if you want to include data collected over “peak” or “critical” periods. For instance, do you want to collect data over a day with a big news event, or do you want to avoid such days?

Step 6: Prioritize migration components

It often makes sense to start with the services that have the fewest dependencies.

Step 7: Perform any necessary refactoring

You may want to do other work on your applications and services before you migrate them, so they work as effectively and efficiently in the cloud as possible. For example, you may want to refactor your application:

So, it works effectively with a variable number of running instances to allow dynamic scaling, potentially saving you money on cloud service costs.

So, your resource utilization can better take advantage of dynamic-cloud capabilities, such as the ability to dynamically allocate and de-allocate resources as needed, rather than you statically allocating them ahead of time.

To move to a more service-oriented architecture before the migration, so that you can more easily move individual services to the cloud.

Step 8: Create a data-migration plan

Migrating data is one of the trickiest parts of a cloud migration. The location of your data can significantly impact the performance of your application. Moving your data to the cloud when the data-access methods are still primarily on-premises can significantly impact performance. The same holds true if the data is still on-premises but the service accessing it resides in the cloud.

Options for data-migration include:

Using a bi-directional syncing mechanism between your on-premises and cloud databases. Once you’ve moved all consumers of the data to the cloud, remove the on-premises database.

Use an on-premises database with one-way synchronization to a cloud-based database, and allow consumers to connect only to the on-premises version. When you’re ready, disable access to the on-premises version so the cloud-based version becomes the main database, and enable cloud-based consumers access to the new database.

Use a cloud data-migration service, such as those available from Amazon Web Services.

Don’t underestimate the complexity and importance of data-migration planning. Not paying close attention to your data-migration plan before you begin a cloud migration can cause migrations to fail, or at least fail to meet expectations. Your migration architect should be very involved in the data-migration planning process.

Step 9: Switch over production

When and how do you switch over the production system from the legacy on-premises solution to the new cloud version? The answer depends on the complexity and architecture of your application, and especially the architecture of your data and datastores.

There are two common approaches:

Do it all at once. Wait until you’ve moved the entire application or service over to the cloud and validated that it works there, and then switch traffic from the on-premises stack to the cloud stack.

Do it a little bit at a time. Move a few customers over, test that things are still working, and then move a few more customers. Continue this process until you’ve moved all your customers to the cloud-based application.

Step 10: Review application resource allocation

Even after you’ve finished migrating everything to the cloud, there are a few more things to consider. Most important is resource optimization. The cloud is optimized for dynamic resource allocation, and when you allocate resources (servers, for example) statically, you’re not taking advantage of the cloud’s strengths. As you move into the cloud, make sure your teams have a plan for distributing resources to your application. When you need to allocate additional resources to an application in the cloud, they are usually available from the vendor in virtually any quantity in a moment’s notice. This means that you can typically trust that you can scale as needed to meet demand, assuming your teams have the application architecture in place to support dynamic scaling.

Other considerations for your cloud migration

The 10 steps in this cloud migration checklist cover a lot of ground, but there are definitely other things you should consider during your cloud migration. Creating a safe and secure cloud environment, for example, is obviously a critical part of any cloud migration. Fortunately, the major cloud providers offer significant tooling and resources to help you build and maintain a secure system.

When it comes to cloud costs, there are two rules of thumb about cloud pricing: the cloud is cheaper than on-premises, and the cloud is more expensive than on-premises. Both can be true or false, depending on the situation.

It is certainly possible to start using the cloud and find out that your infrastructure bill has, in fact, increased compared to what you were spending on your physical data center. There are a couple reasons this could happen:

First, there are hidden costs in all infrastructure systems, and you may not be considering all the costs involved in running your own data center, while the monthly bill you get from your cloud provider makes the costs very clear. The result? Sometimes you end up comparing apples to oranges, making the cloud solution appear more expensive than the on-premises one.

Second, the costs of an on-premises infrastructure are mostly composed of capital expenditures (CapEx), while a cloud-based infrastructure usually comes out of operating expenses (OpEx). Depending on how your business manages its books, CapEx may be easier to come by than OpEx, or vice versa. Understanding how paying for cloud-based infrastructures differs from an on-premises infrastructure, and making sure your company’s financial models support the distinctions, is critical to recognizing cloud cost improvements.