



Migrating Legacy Applications to the Cloud: How to Pick the Right Strategy

Introduction

Organizations migrate their legacy applications to the cloud for reasons as diverse as the companies themselves. The cloud has become synonymous with progress in the business world. It allows companies to get closer to their customers than ever before, move with the speed of the market, focus their staff on innovation, and much more. Every migration takes hard work and finding the right way to move applications to the cloud makes that work pay off.

We have identified four primary types of cloud migration, based on the way different companies want to use the cloud to accomplish their goals. The most important part of any cloud migration is making sure the migration gets your company where it needs to be

The four types of cloud migration are called lift and shift, shift to Software-as-a-Service (SaaS,) application refactoring, and replatforming.

Lift and Shift: Get Out of the Datacenter

The lift and shift approach is for organizations looking to get out of the datacenter and stop managing hardware.

Lift and shift provides the same software that your company used in the data center, but now in the cloud. There isn't any learning curve for the cloud applications since they work exactly the same as before. This is the fastest method for migrating applications to the cloud and the one that causes the least disruption. It only requires the involvement of the infrastructure and security teams, leaving everyone else free to pursue their work uninterrupted. It's also the option with the least upfront cost. Moving the application to the cloud allows it to handle peak performance, without your company having to pay for it.

Lift and shift comes with its drawbacks, however. This cloud migration can't take full advantage of the speed and versatility the cloud can provide. Since the process doesn't change the application – it just moves the code to a new location – the shifted version of the application doesn't usually have better performance than the original. It's also unlikely to lead to long-term savings. This model is best suited to companies with a regular peak schedule and slow, predictable changes in the market. Food delivery companies with their regular peaks on Monday, Saturday, and Sunday are a good example of the former. Tax companies are a good example of the latter since tax rules change on a yearly basis and new rules are released a year in advance, giving the companies plenty of time to prepare.

“You don't move to the cloud to save money. You do it to save time.”

However, tax and food delivery companies aren't the only good candidates for this strategy. We worked with a construction company to lift and shift its huge logistics application to the cloud. The construction company's goal was to get out of the data center. The application was a mixture of custom and off-the-shelf code that the CIO had neither the time nor the money to break up – something that would be required for other methods of migrating to the cloud. Lift and shift accomplished the company's goals with low upfront costs and no need to make changes to the application.

Shift to SaaS: Save Time and Trouble

Companies that want to stop devoting time and resources to applications outside their core business should think about shifting to SaaS.

Shifting to SaaS means outsourcing one or more applications to a cloud services company that specializes in managing those applications. Companies do this on an application by application basis and only shift the applications they need to. Static applications can remain on-premises. Shifting to SaaS frees employees up to focus on core competencies and the things that make a business unique and competitive. Outsourcing applications to SaaS also means that you need fewer licenses for business tools. However, it's extremely important when shifting an application to SaaS that you pick the right service.

The main drawback of shifting to SaaS, while you can personalize it, is that customizing it can lead to problems. Interjectory code added to a SaaS tool can cause you to lose the support model and update model provided by the SaaS company. This means that if you start shifting applications directly related to your industry, you can lose the customizations that give your business a competitive edge. Shifting to SaaS should only be used for routine functions – not for anything that needs to be unique.

Email is a good example of a routine business function that can be shifted to SaaS. For example, we worked with a manufacturing company that built engine components. The company was sick of managing its own email. While it needed to have an email for all its employees, the customers didn't care what the email service looked like – so long as it worked. We helped them find a hosting service that freed the company of needing to worry about their email so it could focus on manufacturing.

Application Refactoring: App Modernization

App modernization is a preferred approach for organizations that have specific applications which could benefit from the cloud.

With refactoring, organizations can replicate their legacy applications whole and intact onto a cloud platform. What makes it low risk is that legacy applications can run in parallel while new applications are constructed, with the immediate benefits of agility and speed to market. This approach focuses on the applications that benefit the most from a cloud platform.

Refactoring is about prioritization; it provides lots of opportunities to save over time by minimizing spending on things you won't need once you're in the cloud. You can save money on the platform itself as well by switching to cloud-native services that cost less than the ones you use on-premises. For example, switching from Oracle WebLogic to cloud-native JBoss can cut costs.

Refactoring isn't just about cutting costs. It also allows you to make changes to your enterprise very quickly, which means that you can keep up with your customers. Refactoring lets you respond faster and prioritize updates. One big box store we worked with started out with its applications so hard coded that it took months to do simple things like changing the font or background color. Refactoring got them moving.

“This is when you make a decision to develop in a whole new paradigm.”

With refactoring, you move a select group of your applications to the cloud. Picking the right platform for the applications is extremely important. Look for a tool that addresses speed to change and can fulfill all of your needs, not just some of them. It's also vital that you refactor your applications in the right order.

Refactoring usually requires outside help. We worked with a company that wanted to refactor its entire application suite into modern technology. The project would have taken four years for the company working on their own, but we were able to cut that time down to eighteen months. We provided the company making the migration with its unique model for retraining.

Rather than building the solution in a silo, as most consultancies do, we partner with the company doing the upgrade and training by introducing their development team to the project as domain experts. When introducing the new technology, we rotate through everyone who needs training. Slowly, over time, we dial back how many consultants they have helping until the local IT people are able to handle everything on their own. This means that the IT staff learns by doing, so there is no awkward learning curve. At every point in the process, there is someone trained to handle difficulties on-site.

Replatforming: Develop Apps in the Cloud

Replatforming is for organizations looking to embrace benefits of the cloud, enterprise-wide.

These companies want their core competencies to be scalable, elastic, robust, resilient, redundant, and available. This is the hardest option to implement, requires the most planning for the future, and comes with the most upfront cost, but it's the only option that lets you utilize the full strength and flexibility of the cloud. Replatforming is replacing the application at the code level to make it cloud-native. This is a complete reimagining of the application and usually requires a complete rewrite. However, for those looking to get into containers or microservice architectures, this is the way to go.

“It is the market speed response.”

When considering replatforming, think about how fast your company can change. Then think about how fast it needs to change to keep up with customers and the market. By making applications truly cloud-native, they can be updated and those updates pushed out at the speed of the cloud. This boosts the speed to change across the board for all aspects of the business. Replatformed applications can also be designed to be more modular and thus easier to maintain. Refactoring the application can save development time down the road since modules of code from your first refactored application can be used to extend the capabilities of new applications.

Unlike refactored applications, refactored platforms can work across multiple cloud providers. This makes it easier to port from one mobile platform to another, which positions replatforming as the ideal strategy for those looking to develop mobile applications. We worked with a healthcare company building a net-new mobile application for patient management, which included sending push notifications on appointments and subscriptions. The company wanted to build this application as a cloud-native company. However, they already had internal patient managing applications for the nurses and doctors that complemented the new applications. The company used replatforming to get their legacy patient management application to the cloud, so the application would be ready and waiting while it developed its net-new mobile application.

Determine your goals before you select your approach

Each of the four primary methods we identified comes with its share of advantages and disadvantages. Finding the method that matches your organization's goals and needs is the first step to a successful migration.

When considering replatforming, think about how fast your company can change. Then think about how fast it needs to change to keep up with customers and the market. By making applications truly cloud-native, they can be updated and those updates pushed out at the speed of the cloud. This boosts the speed to change across the board for all aspects of the business. Replatformed applications can also be designed to be more modular and thus easier to maintain. Refactoring the application can save development time down the road, since modules of code from your first refactored application can be used to extend the capabilities of new applications.

Unlike refactored applications, refactored platforms can work across multiple cloud providers. This makes it easier to port from one mobile platform to another, which positions replatforming as the ideal strategy for those looking to develop mobile applications. We worked with a healthcare company building a net-new mobile application for patient management, which included sending push notifications on appointments and subscriptions. The company wanted to build this application as a cloud-native company. However, they already had internal patient managing applications for the nurses and doctors that complemented the new applications. The company used replatforming to get their legacy patient management application to the cloud, so the application would be ready and waiting while it developed its net-new mobile application.

Determine your goals before you select your approach

Each of the four primary methods we identified comes with its share of advantages, but also with its disadvantages. Finding the method that matches your organizations goals and needs is the first step to a successful migration.

About Shadow-Soft

Shadow-Soft is a systems integrator with a rich open source history. Our team of strategists and engineers help organizations navigate their technology journey with confidence.

 Call Shadow-Soft: 770-546-0077