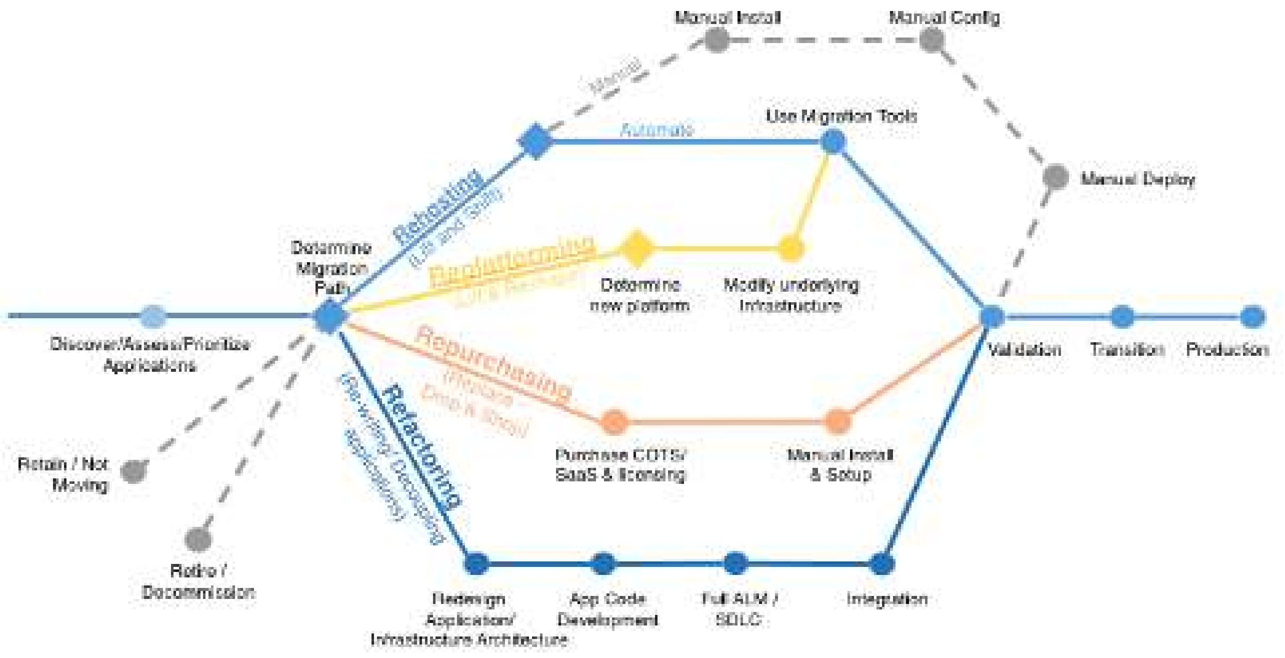
Application Migration Strategies

Learn the common cloud migration strategies.



There six different migration strategies organizations use to migrate applications to the cloud. These strategies build upon the 5 R's that Gartner outlined here in 2011.

The most common application migration strategies we see are:

Re-Hosting

We find that many early net-new development initiatives like startups build cloud-native applications, but in a large legacy migration scenario where the organization is looking to scale its migration quickly to meet a business case, we find that the majority of applications are rehosted. GE Oil & Gas, found that, even without implementing any cloud optimizations, it could save roughly 30 percent of its costs by rehosting with a cloud provider. White paper here.

Most rehosting can be automated with tools available from cloud providers like AWS, Azure, Oracle etc although some customers prefer to do this manually as they learn how to apply their legacy systems to the new cloud platform.

Applications are easier to optimize/re-architect once they're already running in the cloud. Partly because organizations will have developed better skills to do so, and partly because the hard part designing the network topology, migrating the application, data, and traffic—has already been done.

Replatforming

This is what is called "Lift-and-Shift". Here you might make a few cloud (or other) optimizations in order to achieve some tangible benefit, but you aren't otherwise changing the core architecture of the application. You may be looking to reduce the amount of time you spend managing database instances by migrating to a database-as-a-service platform like Amazon Relational Database Service (Amazon RDS), or migrating your application to a fully managed platform like Amazon Elastic Beanstalk.

A large media company migrated hundreds of web servers it ran on-premises to AWS, and, in the process, it moved from WebLogic (a Java application container that requires an expensive license) to Apache Tomcat, an open-source equivalent. This media company saved millions in licensing costs on top of the savings and agility it gained by migrating to AWS.

Repurchasing

Moving to a different product. Moving a CRM to Salesforce.com, an HR system to Workday, a CMS to Drupal, and so on.

Refactoring / Re-Architecting

Re-imagining how the application is architected and developed, typically using cloud-native features. This is typically driven by a strong business need to add features, scale, or performance that would otherwise be difficult to achieve in the application's existing environment.

Customers using this approach are looking to migrate from a monolithic architecture to a service-oriented (or server-less) architecture to boost agility or improve business continuity. This pattern tends to be the most expensive, but, if you have a good product-market fit, it can also be the most beneficial.

Retire

Once there is an audit done to discovered everything in your environment, you might ask each functional area who owns each application. It has been

found that as much as 10% to 20% of an enterprise IT portfolio is no longer useful, and can simply be turned off. These savings can boost the business case, direct your team's scarce attention to the things that people use and lessen the surface area you have to secure.

Retain

Usually this means "revisit" or do nothing (for now). Maybe you're still riding out some depreciation, aren't ready to prioritize an application that was recently upgraded, or are otherwise not inclined to migrate some applications. You should only migrate what makes sense for the business; and, as the gravity of your portfolio changes from on-premises to the cloud, you'll probably have fewer reasons to retain.