

# Migration Strategy Checklist

Before moving ahead with your cloud migration planning, you need to determine your migration strategy. This is simply how best to handle your existing apps. (Your previous app profiling should have already informed you as to which apps must be retained, must stay on premise due to security, compliance or other issues, or can be moved to the cloud.) You can:

- Retire/replace.** With this option, you remove an app that has no or low business value and then determine if it should be replaced by new cloud-native app.
- Retain.** You simply keep the app where it is, how it is. Not all apps can or should be moved to the cloud for a variety of reasons.
- Rehost, also known as “lift and shift.”** This entails moving apps to the cloud as-is. It’s the easiest, least expensive strategy. However, not all apps will perform optimally in the cloud without some modification.
- Refactor.** This requires rearchitecting an app to improve its performance, agility and ability to leverage the benefits of the cloud.
- Replatform, also referred to as “lift, tinker, and shift” or “lift and reshape.”** You move the app to the cloud, introducing some level of optimization to increase the app’s efficiency, reduce costs, and/or take advantage of cloud capabilities.
- Rebuild.** This requires redeveloping an app from scratch using cloud-native technologies and services. Doing so allows you to modernize the app so it can take full advantage of cloud capabilities.
- Repurchase.** In some instances, it may make sense to purchase a SaaS app to replace an existing app.

## The New Environment

Another critical “to-do” prior to the actual migration is architecting and testing the new environment(s). A cloud architect will need to design the appropriate cloud environment(s).

This entails assembling virtualized compute, storage, and networking instances, along with services such as databases, logging/monitoring tools, security, and event-driven computing. Adequate bandwidth will be critical for optimal application performance.

Make sure testing is done to validate the architecture and ensure that the workload will function properly once deployed and cut over for production. This may entail a proof-of-concept with iterations and refinements before an actual migration/cutover takes place.