7 STEPS TO SIMPLIFY MIGRATING ARCHITECTURES TO THE CLOUD

Face it: most IT architectures are complicated. And if you're considering migrating to the cloud, you're right to be concerned about the vast changes that will be required of your architecture—and your organization—as you make your transition.

The good news is that if you're like most companies, you've done this before. Many times. About every three to five years you overhaul your core architectures. You adjust how you deliver applications. You strive to increase performance, enhance security, and reduce costs.

The bad news is that with cloud, things will be even more complicated. You might not have control over services. You may not be able to hard code connections or do things the old way. There will be some pain. But, like they say, "No pain, no gain," right? You'll have to plan for user behavior, connectivity, and appropriate bandwidth.

To make the transition just a little bit smoother, we've compiled seven important steps to get you started.

1. ASSESS WHAT YOU HAVE

What is the state of your applications? How many do you have? How important are they to your business? What sorts of data do they hold, and—most importantly—what are the dependencies between them?

Start thinking about the categories your apps will fall into. You have four options:

1. Adopting SaaS

2. Migrating to the cloud

3. Adopting a hybrid environment

4. Keeping them where they are

2. DECIDE WHICH APPS ARE RIPE FOR OUTSOURCING TO SAAS

Do the easy part first. Identify the apps in your portfolio that are virtual commodities. You're likely to have a lot of them. Do you really need to support your own Exchange server, your out-of-date HR system, or your homegrown sales automation tools? Are they worth the efforts of your team or the OpEx you incur? If not, save yourself a lot of trouble by subscribing to a sales, HR, productivity, or other appropriate solution. Let third parties do your heavy lifting. You'll get obvious, quick wins with SaaS.

3. ANALYZE AND DECIDE ON THE REST

Next, you'll need to assess your remaining apps and decide which you will move to the cloud, which you'll refactor for the cloud, and which to keep as-is.

Ask yourself the following questions:

- If we migrate app X, how many things will break?
- Where are the data stores?
- What are the dependencies?
- What network services are they using?
- Which apps require workarounds to normal procedures and protocols to make them work?

You'll have answers to those questions for many of your apps. For others, you may not know the answers until you actually try to move them. The greater the risk of breakage and the more complicated and less known the dependencies are, the more likely you are to keep an app where it is.

As you map out these dependencies, document them. This will be useful even if only a few of your apps end up in the cloud.

4. STANDARDIZE

Examine your app delivery policies and look for opportunities to standardize and automate. You should have a limited number of standard load balancing policies—say, 10—rather than hand-tuned configurations for every app. Determine standardized storage tiers. Define standardized network services. Talk to your developers about the benefits of standardization and gain their commitment. Make templates to help them deploy things quickly and easily.

5. SIMPLIFY AND SECURE ACCESS

Ask yourself who is going to be accessing each app and from where. You have to plan for user behavior, connectivity, and appropriate bandwidth. Many of the applications that you seek to migrate to the cloud—whether private or public—may need to be more readily accessible from anywhere. Moving them to the cloud will place less stress on the infrastructure.

There are also authentication and security issues; most businesses have traditionally used network rather than app controls to determine access. In a public cloud, you may need to adopt new identity and access management technologies that you didn't have before.

6. PLAN YOUR ARCHITECTURE

When migrating to the cloud, the architecture will be different because the constructs aren't static. For monolithic applications like databases, the mechanisms that were formerly tied to specific IP addresses or other constant constructs won't work in the cloud. You may need additional load balancers or proxies that will help provide consistency in an environment that is always changing. Make additional points of control so you can ensure that everyone can access your apps consistently and without disruption.

7. REMEMBER THAT IT'S NOT EASY

This is hard stuff. As we said at the beginning, IT architectures are complicated.

While it may not be easy, it's worthwhile—for the cost savings (OpEx and CapEx) and scalability alone. And some enterprises have achieved massive savings just by preparing for cloud. By assessing your existing app inventories, analyzing dependencies, documenting everything, and standardizing and simplifying as much as possible, you'll be in the perfect position to decide what to move and how to do it.

Find out more at f5.com/solutions/deploy-consistent-policies-across-clouds