

Managing Tech Migration and Replatforming

Grant Zukel

My experience in Tech Migration and Replatforming



What are the different sub-components of a migration?

Infrastructure



The systems your software runs on, including your servers, operating system, and database platform.

Software Stack



Changes could include language choices, frameworks and architecture, and tooling of the software you have built.

Data



It's usually crucially important to your organization to migrate this successfully from one system to another.

Skills



If you have a specialized team, migrating any of the fundamental pieces of your platform or software can result in a big shift in talent needs.

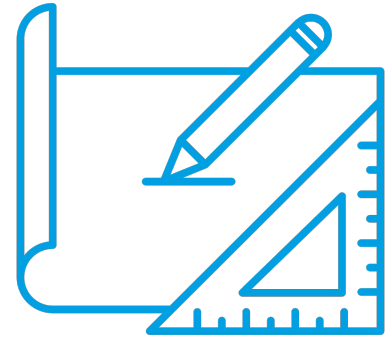
What are the different migration paths you might take?



Rehost



Replatform



Rearchitect

What approach you take depends upon:

The benefits you're trying to realize

Current state of your platform

Engineering resources available

What are the different migration paths you might take?

Rehost

“Lift and Shift”, taking exactly what you have in your existing situation and rebuilding it as closely as possible in a new environment

Advantages & Disadvantages

- ✓ It's frequently the easiest way to migrate quickly
- ✓ There's a lower chance of outages and customer impact in the short term
- ✗ You get the least benefit from the migration and likely miss a lot of available tools using a lift and shift.
- ✗ It costs the most in the long run.
- ✗ There's a higher risk of outage or data loss in the long term

What are the different migration paths you might take?

Replatform

Shifting your approach to accommodate the new platform but not committing to fully invest right out of the gate

Advantages & Disadvantages

- ✓ It may not lock you into investing in a specific vendor and you can still invest to get optimizations.
- ✓ If you don't know how your application works, this approach gives you the chance to document how it works as you migrate each piece
- ✗ You don't get the full benefit of investing in the cloud when you have one foot in and one out.
- ✗ You need to coordinate more teams, and it goes more slowly (generally avoid this approach in enterprise environments)

What are the different migration paths you might take?

Rearchitect

Committing to a full rewrite of your application

Advantages & Disadvantages

- ✓ You can get the most benefit from investing in the platform you're moving to
- ✓ If you keep your original platform running in parallel, it lowers your chances of data loss
- ✓ If you're moving to the cloud, you likely need to redesign for the cloud
- ✗ It takes a lot of work and there is a high risk of failure
- ✗ The need for new skills (vs. your existing team) can be significant

When should (and shouldn't you) replatform?

Reasons to Replatform

- + Enhance your ability to automate
- + Expand your talent pool for recruiting
- + Cost savings (depending on your traffic pattern)
- + Security improvements
- + Performance improvements
- + Merger or acquisition

Risks of Replatforming

- ! The potential for heavy investment without the results you want
- ! Security risks
- ! Damaging relationships with your engineering team
- ! Switching to a new platform that ultimately doesn't survive
- ! Data loss or outages

What are some common myths about migration?

Myth: that cloud setups are less secure

Myth: that more control = better performance

Myth: that replatforming will always save you money (or cost more)

What engineering process changes might you change alongside a replatforming?



What can you do to change the way your application is structured to save on costs?



Take advantage of elasticity



Dynamically leverage "spot" compute



ARM servers are typically less expensive



Don't overprovision underutilized or abandoned resources

When should you consider switching to a different cloud provider to save on costs?

Savings aren't usually significant enough to justify switching costs, unless:



You're a small startup and have been offered significant credits to switch



You have a huge infrastructure and can secure a significant discount by signing a long-term contract with another cloud provider



Your infrastructure is significantly cloud-agnostic (i.e., doing everything in Kubernetes), making migration a simpler undertaking

What are some common pitfalls?



Over or under-planning



Being overly optimistic



Questions