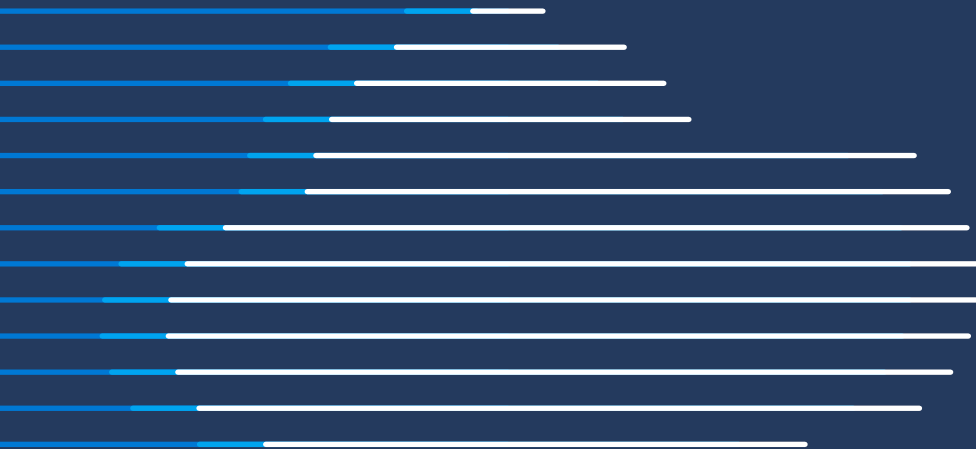


Cloud Rationalization: The Five Rs

The “five Rs of rationalization” are a great way to label a potential future state for any workload that's being considered as a cloud candidate. Here we describe the most common options for rationalization.





Contents

01

Rehost

02

Refactor

03

Rearchitect

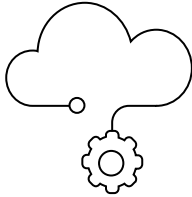
04

Rebuild

05

Replace

01



Rehost

Also known as a *lift and shift* migration, a rehost effort moves a current state asset to the chosen cloud provider, with minimal change to overall architecture.

Common drivers might include

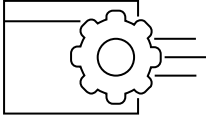
- Reducing capital expense
- Freeing up datacenter space
- Achieving rapid return on investment in the cloud

Quantitative analysis factors

- VM size (CPU, memory, storage)
- Dependencies (network traffic)
- Asset compatibility

Qualitative analysis factors

- Tolerance for change
- Business priorities
- Critical business events
- Process dependencies



Refactor

Platform as a service (PaaS) options can reduce the operational costs that are associated with many applications. It's a good idea to slightly refactor an application to fit a PaaS-based model.

“Refactor” also refers to the application development process of refactoring code to enable an application to deliver on new business opportunities.

Common drivers might include

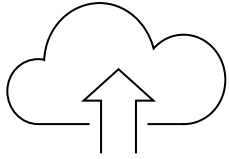
- Faster and shorter updates
- Code portability
- Greater cloud efficiency (resources, speed, cost, managed operations)

Quantitative analysis factors

- Application asset size (CPU, memory, storage)
- Dependencies (network traffic)
- User traffic (page views, time on page, load time)
- Development platform (languages, data platform, middle-tier services)
- Database (CPU, memory, storage, version)

Qualitative analysis factors

- Continued business investments
- Bursting options/timelines
- Business process dependencies



Rearchitect

Some aging applications aren't compatible with cloud providers because of the architectural decisions that were made when the application was built. In these cases, the application might need to be rearchitected before transformation.

In other cases, applications that are cloud-compatible, but not cloud-native, might create cost efficiencies and operational efficiencies by rearchitecting the solution into a cloud-native application.

Common drivers might include

- Application scale and agility
- Easier adoption of new cloud capabilities
- Mix of technology stacks

Quantitative analysis factors

- Application asset size (CPU, memory, storage)
- Dependencies (network traffic)
- User traffic (page views, time on page, load time)
- Development platform (languages, data platform, middle tier services)
- Database (CPU, memory, storage, version)

Qualitative analysis factors

- Growing business investments
- Operational costs
- Potential feedback loops and DevOps investments

04



Rebuild

In some scenarios, the delta that must be overcome to carry an application forward can be too large to justify further investment. This is especially true for applications that previously met the needs of an organization but are now unsupported or misaligned with the current processes. In this case, a new code base is created to align with a [cloud-native](#) approach.

Common drivers might include

- Accelerate innovation
- Build apps faster
- Reduce operational cost

Quantitative analysis factors

- Application asset size (CPU, memory, storage)
- Dependencies (network traffic)
- User traffic (page views, time on page, load time)
- Development platform (languages, data platform, middle tier services)
- Database (CPU, memory, storage, version)

Qualitative analysis factors

- Declining end-user satisfaction
- Business processes limited by functionality
- Potential cost, experience, or revenue gains



Replace

Solutions are typically implemented by using the best technology and approach available at the time. Sometimes software as a service (SaaS) applications can provide all the necessary functionality for the hosted application. In these scenarios, a workload can be scheduled for future replacement, effectively removing it from the transformation effort.

Common drivers might include

- Standardizing around industry-best practices
- Accelerating adoption of business process-driven approaches
- Reallocating development investments into applications that create competitive differentiation or advantages

Quantitative analysis factors

- General operating cost reductions
- VM size (CPU, memory, storage)
- Dependencies (network traffic)
- Assets to be retired
- Database (CPU, memory, storage, version)

Qualitative analysis factors

- Cost benefit analysis of the current architecture versus a SaaS solution
- Business process maps
- Data schemas
- Custom or automated processes