

doit

Common AWS Migration Acceleration Program (MAP) Mistakes

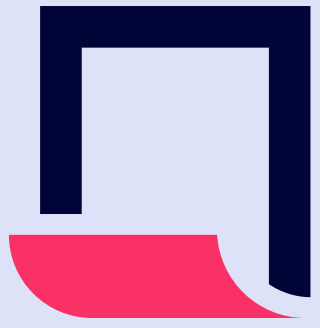
Phase 1: Assess

Phase 2: Mobilize

Phase 3: Migrate & Modernize

doit

Phase 1: Assess



Using handwritten/ outdated inventory sheets

Why it's bad

Relying on outdated or inaccurate inventory sources leads to TCO calculations that are way off, which can impact business decisions and result in underfunding for MAP.

Do this instead

Add multiple, high-quality data sources to improve TCO accuracy and forecasting.



Matching on-prem server sizes 1:1 to AWS

✘ Why it's bad

On-prem servers perform differently than cloud instances. Simply matching specifications without performance testing often results in over-provisioning and unnecessary costs.

✔ Do this instead

Conduct thorough load testing and performance analysis to right-size instances, and consider using AWS instance types optimized for your specific workload patterns.



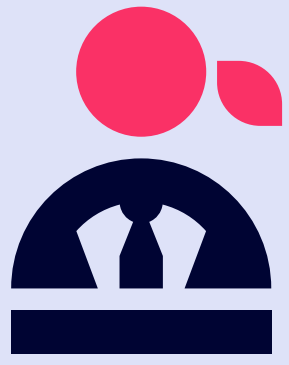
Excluding security-related service costs from your TCO calculation

Why it's bad

Many companies only calculate core service costs (like EC2) but forget the required security, disaster recovery, and monitoring services that need to be part of a Well-Architected deployment, leading to significant budget surprises post-migration.

Do this instead

Include all required security, monitoring, and operational tools in your TCO calculations. Work with your security team and partner to map out complete cloud security architecture and associated costs.



No CISO involvement

Why it's bad

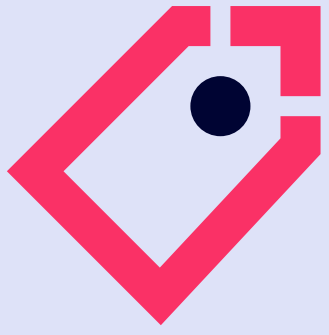
Security teams must be involved early to ensure compliance requirements, and security controls are properly mapped to cloud services and included in planning. Otherwise, you risk misalignment with AWS Well-Architected security principles.

Do this instead

Include security stakeholders from day one and integrate their requirements into your migration strategy, architecture decisions, and TCO calculations to ensure architecture follows AWS security best practices and compliance requirements.

doit

Phase 2: Mobilize



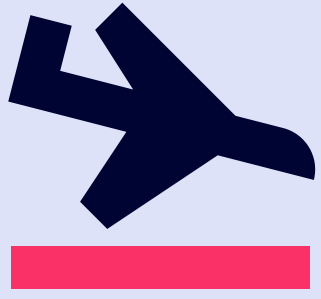
Missing MAP tags on migrated resources

Why it's bad

Incomplete or inconsistent tagging means resources won't be counted towards MAP funding calculations, making tracking difficult. This becomes especially challenging with complex architectures involving many interconnected services.

Do this instead

Implement regular tag compliance checks with clear processes. For non-taggable resources (like Amazon Connect) or services not in AWS's calculator (like non-Titan Amazon Bedrock models), ensure proper manual submission procedures.



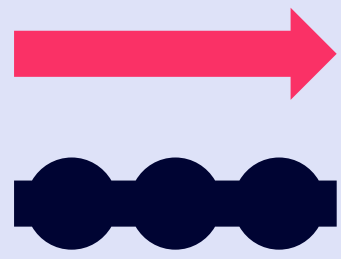
Deploying full landing zones before validating needs

Why it's bad

Building out complete landing zones too early leads to unnecessary costs from unused features, and potentially inappropriate architectural decisions

Do this instead

Start with a minimal viable landing zone, validate your needs through pilot migrations, and incrementally build out your landing zone based on validated requirements.



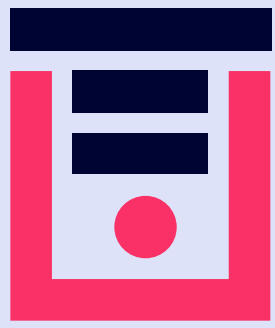
Skipping bandwidth capacity calculations

✘ Why it's bad

Not properly calculating data transfer requirements can result in network bottlenecks during migration and ongoing operational issues between regions or hybrid setups.

✔ Do this instead

Calculate bandwidth requirements upfront, including peak usage patterns, and plan for both migration and steady-state bandwidth needs so your landing zone design can support expected traffic patterns.



Attempting migration with deprecated runtimes

✘ Why it's bad

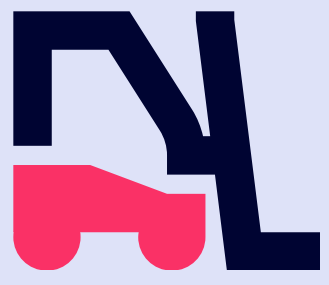
Moving applications running on outdated runtimes (like Python 2.7) without modernization can block deployment since these versions aren't supported in AWS services. Even minimum supported versions may face deprecation soon, forcing urgent updates.

✔ Do this instead

Assess application compatibility early and target latest runtime versions rather than minimum supported ones. Create a modernization plan for outdated components before migration and include this effort in your timeline and budget.

doit

Phase 3: Migrate & Modernize



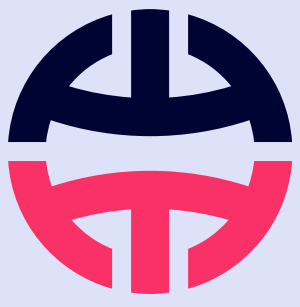
Treating lift-and-shift as the final state

Why it's bad

Simply moving applications “as-is” without planning for optimization misses the cost and performance benefits of cloud-native services and architectures.

Do this instead

Create a post-migration optimization roadmap that includes specific modernization goals, timeline, and budget for adopting cloud-native services and architectures.



Limiting deployment to a single region

Why it's bad

Single-region deployments become a major problem when business changes require global access, like during company acquisitions, market expansion, or when disaster recovery becomes necessary.

Do this instead

Design your architecture with multi-region capabilities in mind, even if you start with one region. Include DR requirements and global expansion possibilities in your planning.



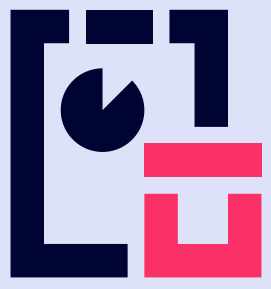
Underestimating hybrid connectivity needs

Why it's bad

During migration phases where applications run in both environments, insufficient bandwidth can cause slowdowns and performance issues across all applications.

Do this instead

Calculate total bandwidth requirements accounting for both migration traffic and applications running in hybrid mode, ensuring your landing zone and network planning can support the expected load



Omitting developer hours from modernization budget

Why it's bad

Cloud optimization shouldn't end once you migrate. Not budgeting for the development effort needed to optimize applications post-migration often leads to staying in an inefficient lift-and-shift state indefinitely.

Do this instead

Include modernization development costs in your initial budget and create a dedicated modernization team or allocate specific sprint capacity for ongoing optimization work.



Planning an AWS migration?

As a certified AWS MAP Partner, DoIT's solution architects have deep expertise in guiding successful cloud migrations — from comprehensive infrastructure assessment and capacity planning to post-migration optimization.

Reach out to talk about your migration strategy and how we can help ensure your successful transition to AWS.

[Book AWS migration consultation](#)