Erfolgreich in die Cloud migrieren? Geht das?

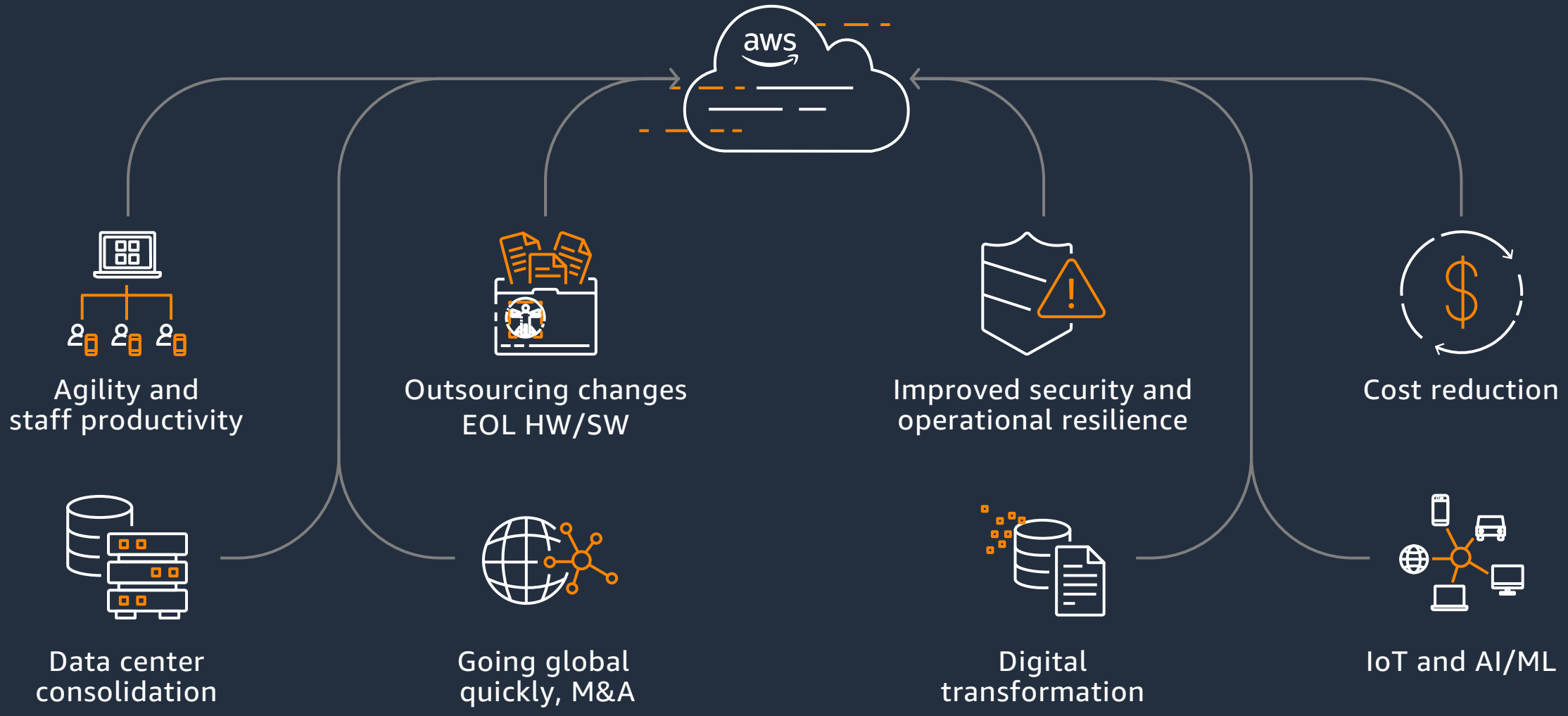
Ja, und zwar mit den richtigen Strategien und Werkzeugen!

David Surey

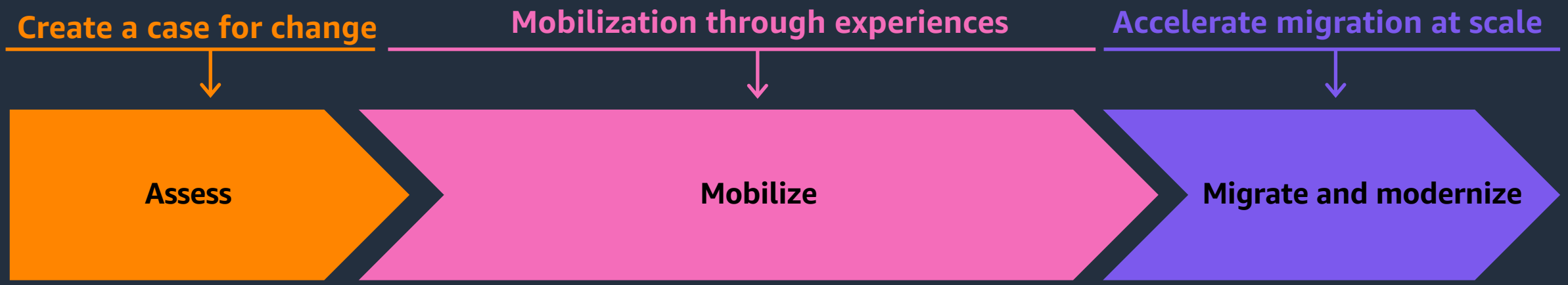
Senior Solutions Architect
Amazon Web Services

Strategy

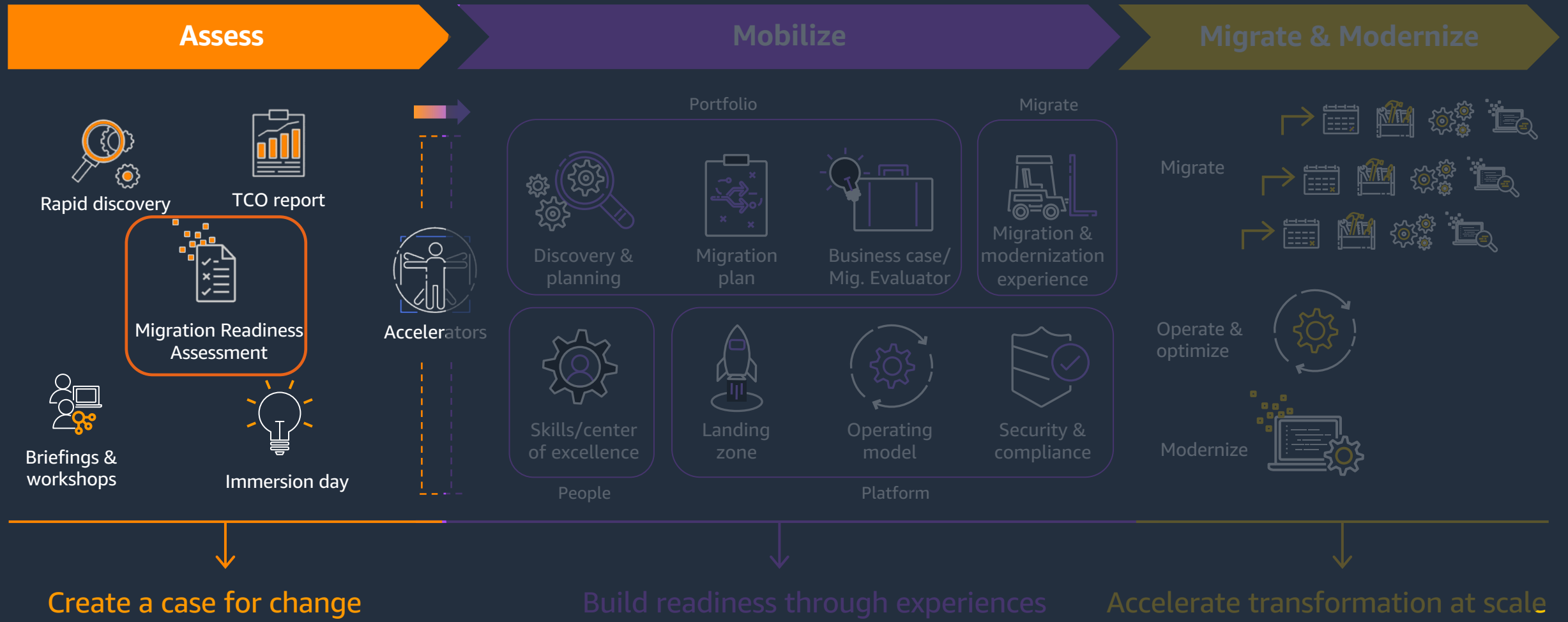
Business drivers for cloud migrations



AWS Migration Framework – migration phases



AWS Migration Framework – Assess



Using a proven framework greatly increases your odds of success and speed to market

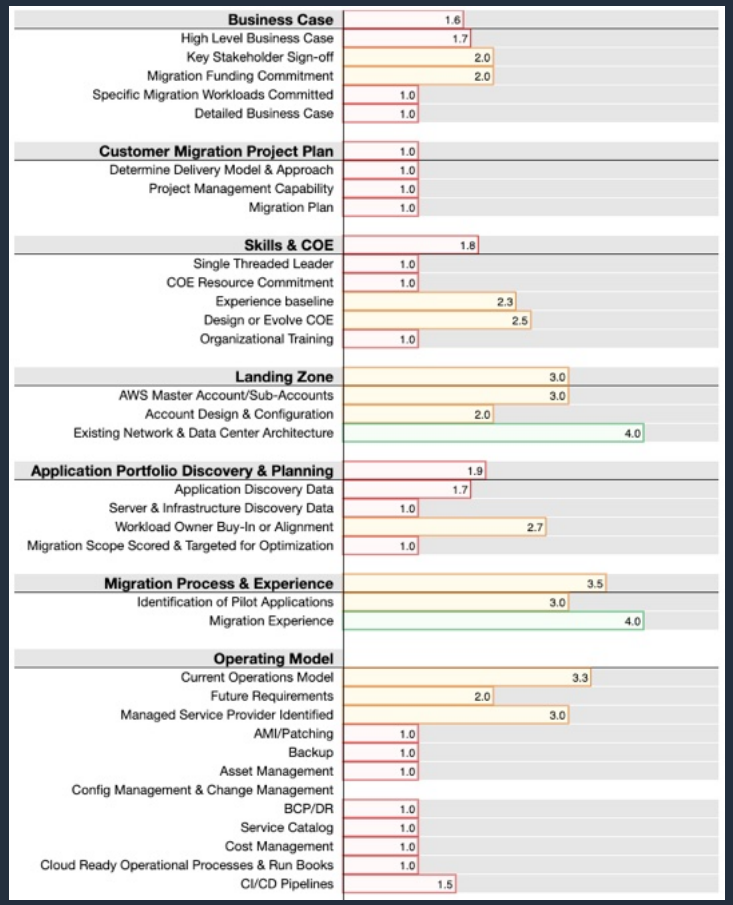
Migration Readiness Assessment – an overview

The assessment includes question survey and interactive activities conducted as a 1-day workshop:

- Alignment of leadership teams
- Consensus on decision-interdependency
- Identification of gaps and readiness

Action plan and proposal to close the gaps and accelerate to the next phase of adoption (e.g. Accelerators, Mobilize project)

Business Capability Focused	**Business** Value Realization
	People Roles and Readiness
	Governance Prioritization and Control
Technical Capability Focused	**Platform** Applications and Infrastructure
	Security Risk and Compliance
	Operations Hybrid and Dynamic

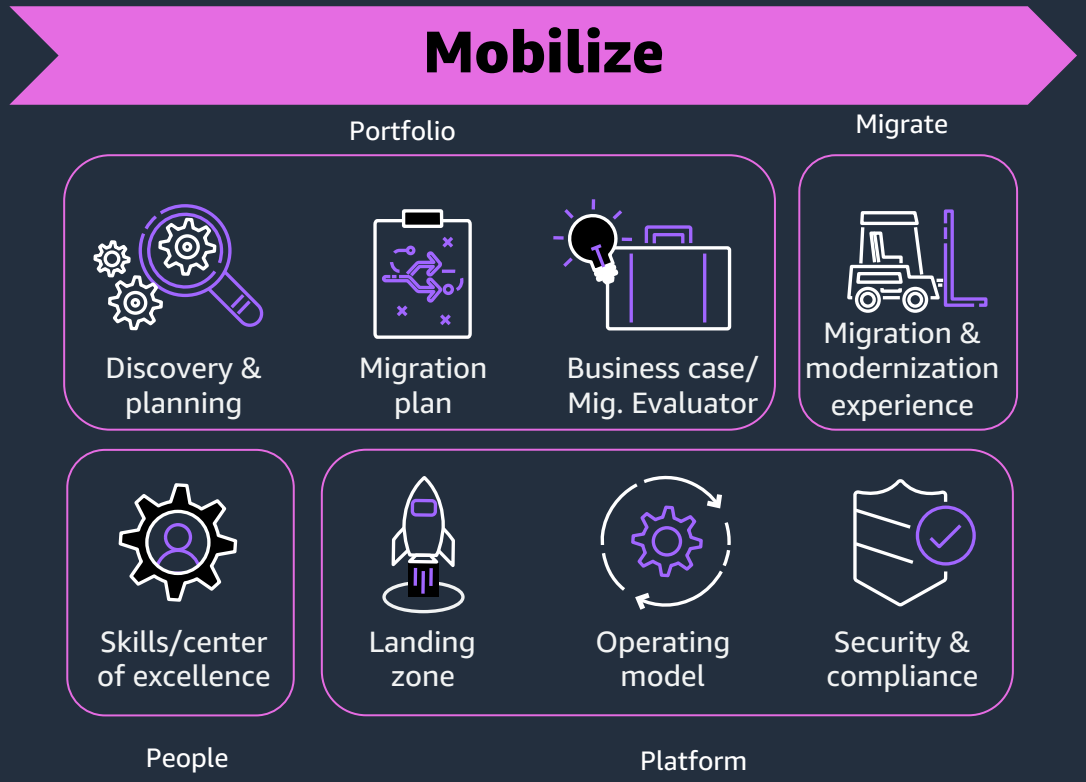


Diagnose migration readiness, align leadership teams, and build a case for change

AWS Migration Framework – Mobilize

Goals

- Build or validate AWS foundational environment to support targeted workloads and future initiatives
- Establish governance and security posture
- Scale enterprise operations to AWS
- Identify migration patterns
- Define team model and agile work streams
- Develop cost and resource model for the migration of a defined portfolio



Outcomes

Platform

- Landing Zone with security controls
- Adapted operational procedures

Portfolio

- 7R disposition & decision criteria
- App discovery & migration Wave Plan

Migration

- 3-5 applications migrated to AWS
- Hands-on migration experience & patterns for use to scale migrations

People

- CCoE structure, skills, & staffing
- Organization change management plan(s)

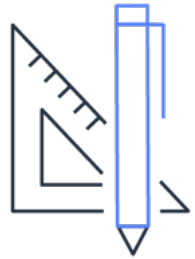
Enable readiness for a mass migration, modernization or greenfield

Establish Cloud Foundations

Guided path

Process

People



Technology

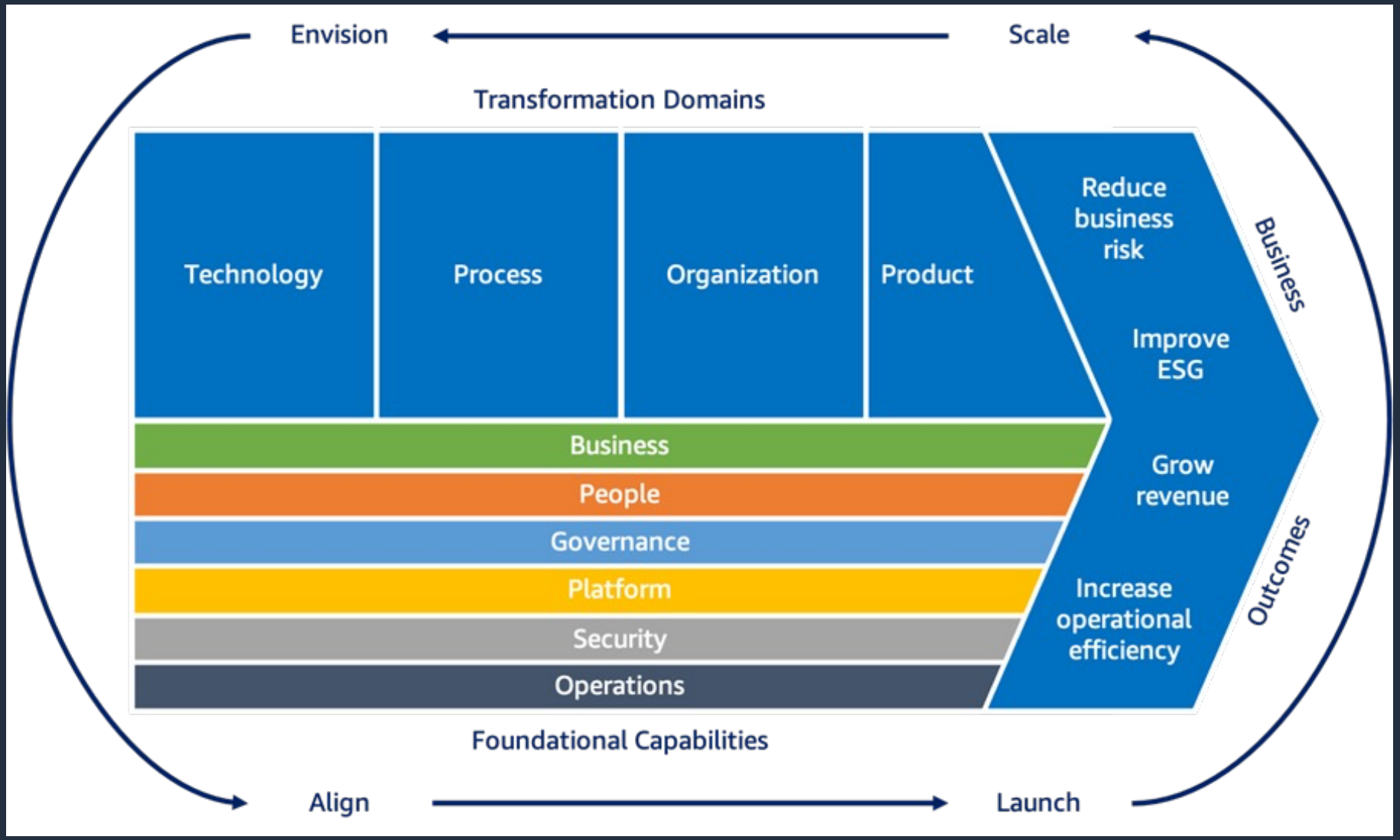
Deploy

Operate

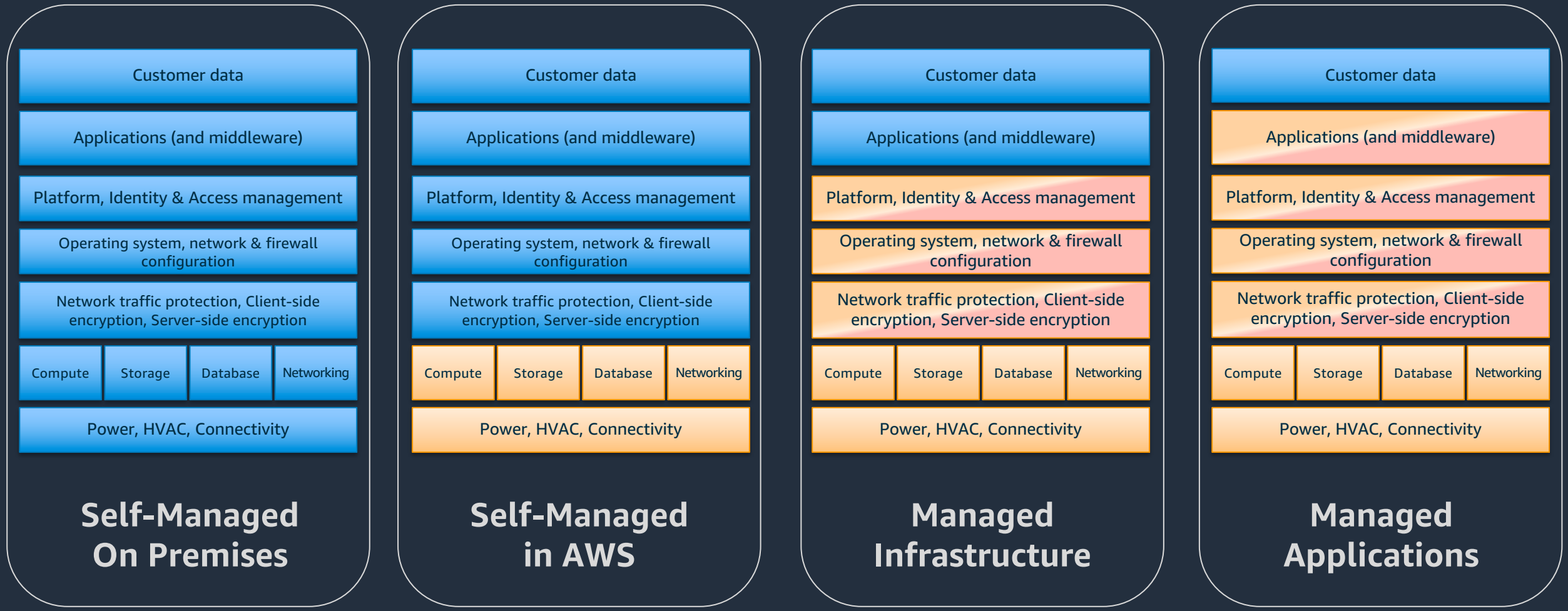
Govern

Production Workloads

The Cloud Adoption Framework

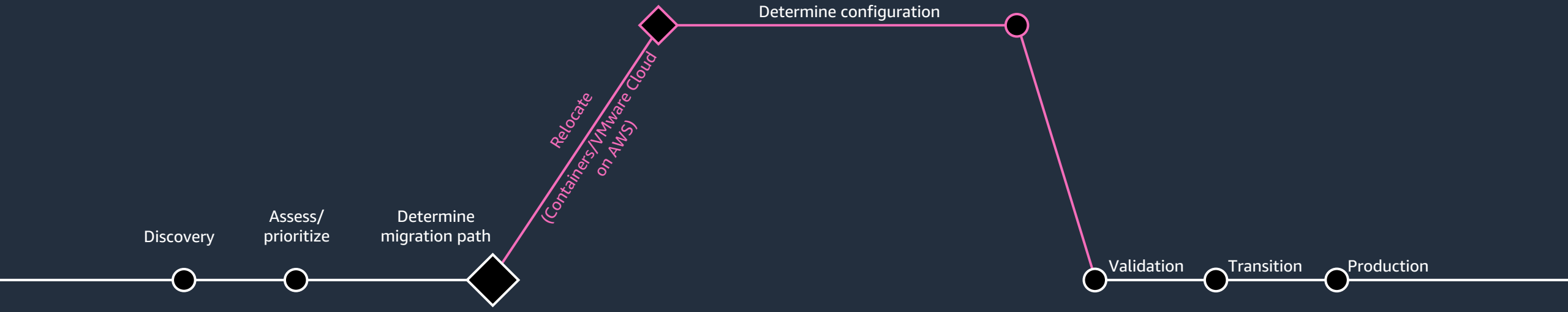


Cloud Infrastructure Operating Models

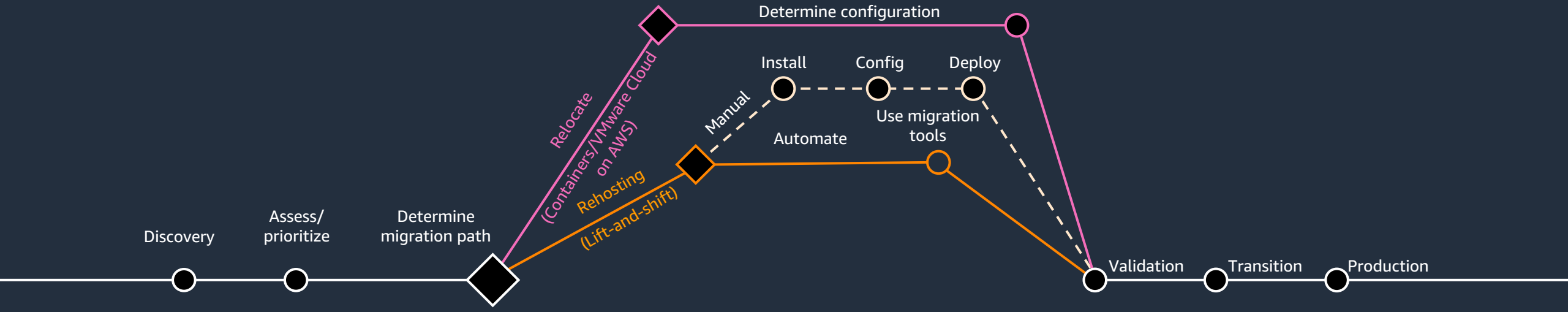


Customer | AWS | Partner MSP

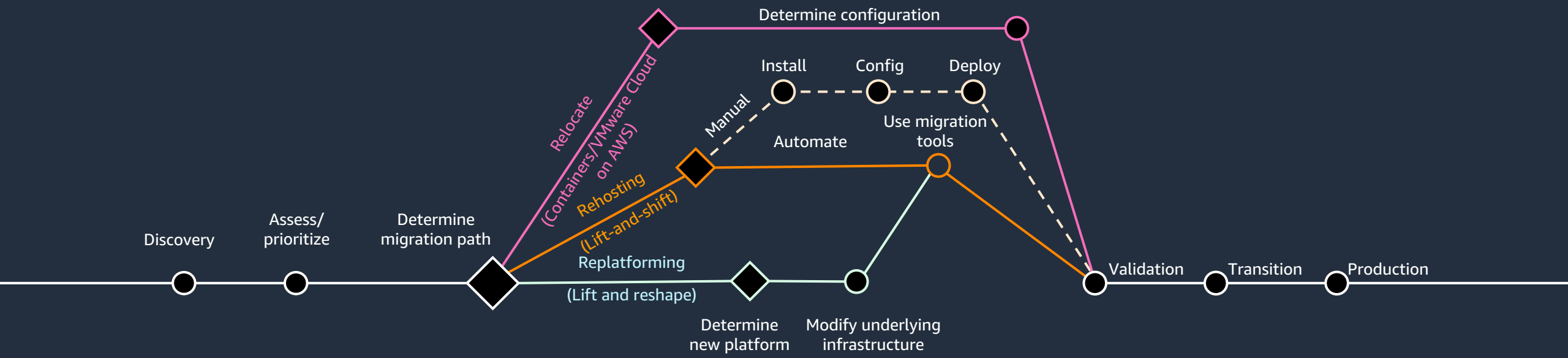
Application migration strategies: Relocate



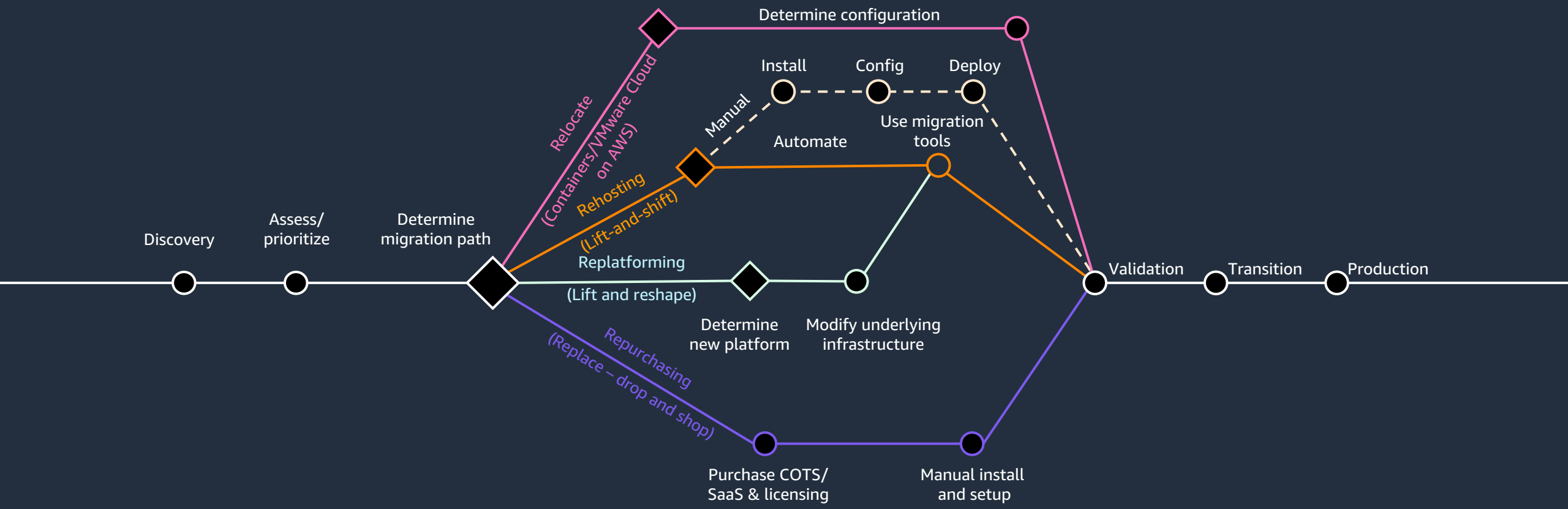
Application migration strategies: Rehost



Application migration strategies: Replatform



Application migration strategies: Repurchase



Cloud marketplaces deliver significant digital transformation advantages

- Transform the enterprise digital supply chain with software and data you know and trust
- Speed up procurement, improve governance, and optimize IT spend, all in one place

Software and data you know and love

Broad selection for app portfolio modernization

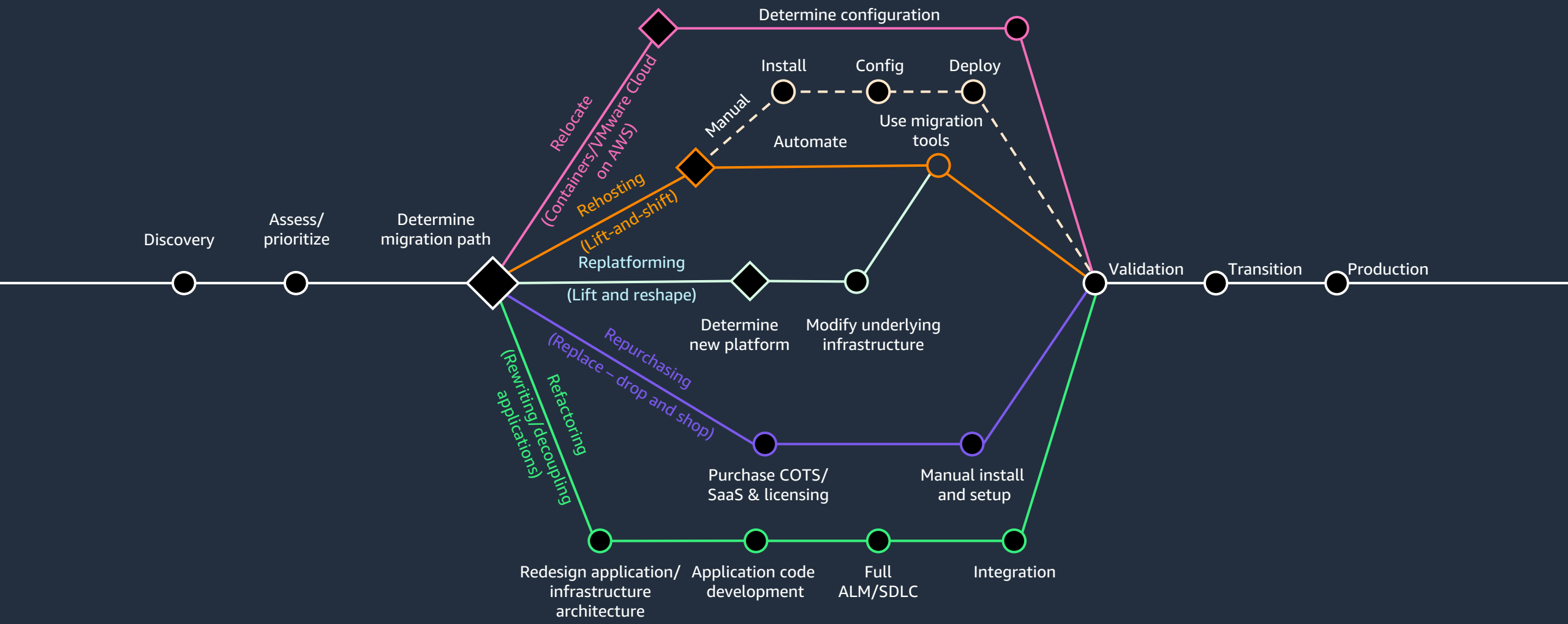
Make transactions & procurement processes easier

Enhance governance and control

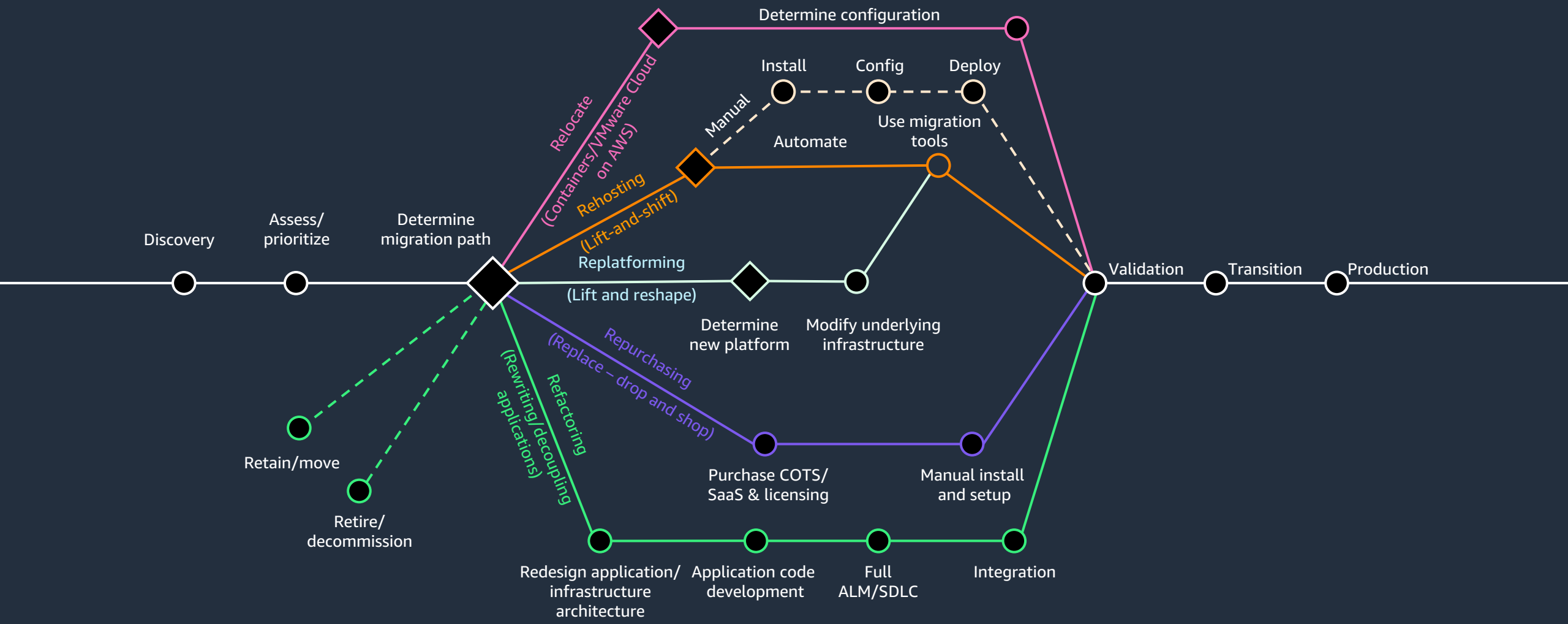
Incorporate professional support and expertise

Optimize IT spend

Application migration strategies: Refactoring



Application migration strategies: Retire and Retain



AVERAGE CUSTOMER ENVIRONMENT, BY MIGRATION PATTERN

(based on AWS experience)

REHOST
Lift-and-shift to quickly capture business benefits; transform once migrated for additional value

RELOCATE
Move VMware-based applications to VMware Cloud on AWS (VMC)

REPLATFORM
Small optimizations (e.g., to reduce OS/DB licensing costs)

REFACTOR
Reimagine application architecture and development

REPURCHASE
Move to SaaS

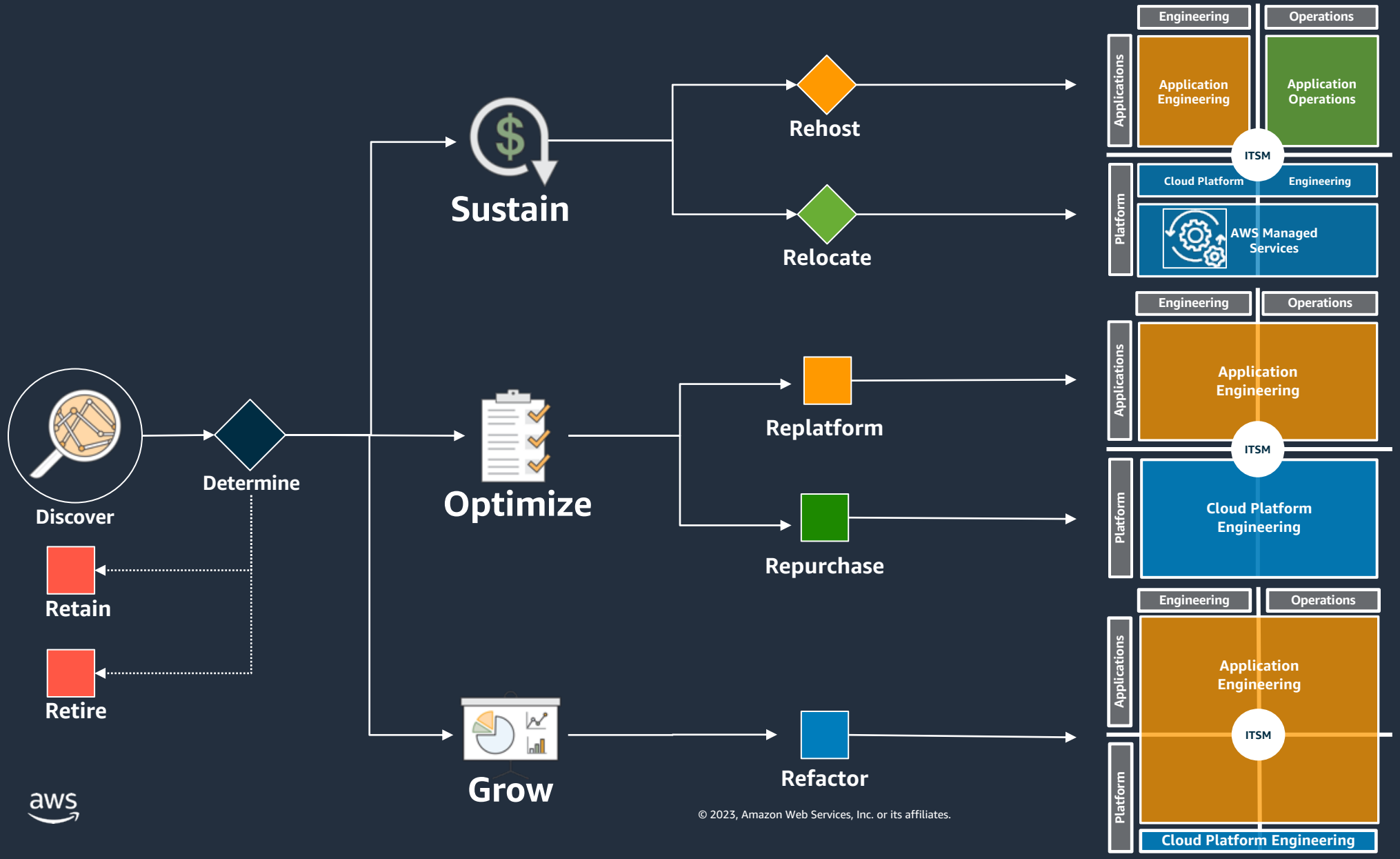
100%

RETAIN
Not prioritized for migration

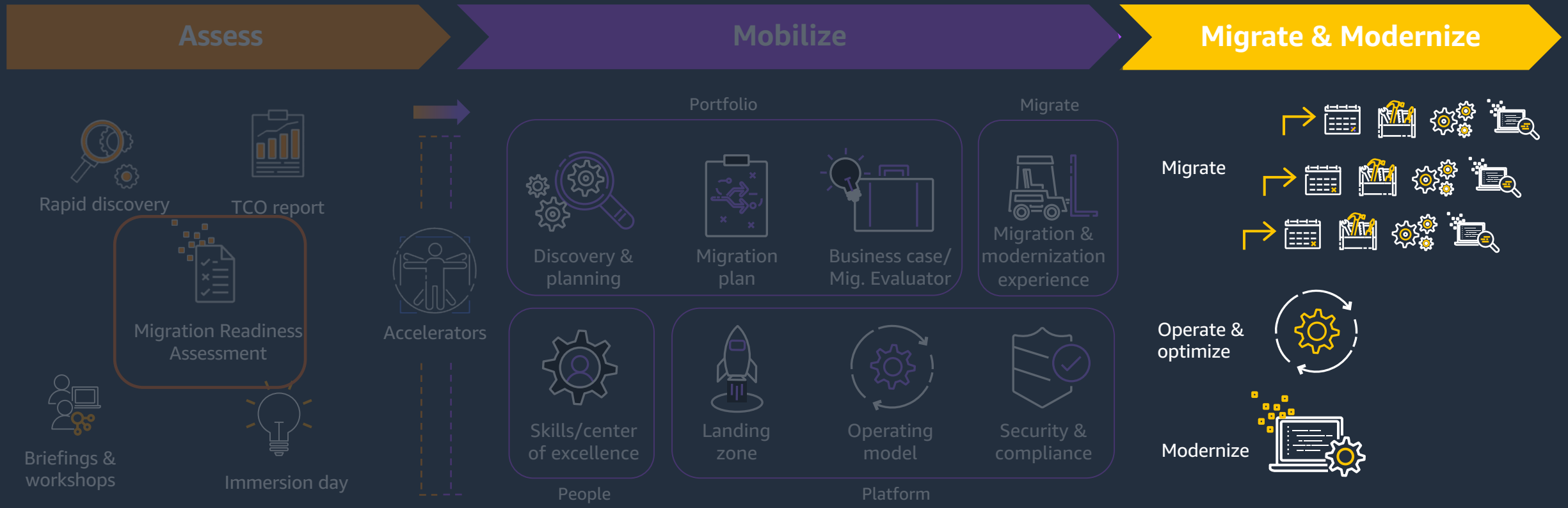
RETIRE
No longer needed

Full spectrum of patterns is important for transformation – but up to ~50% of typical environment can be *rapidly migrated*, freeing time and budget to focus on modernization

Operating Model Effect on Migration Strategies



AWS Migration Framework – Migrate & Modernize

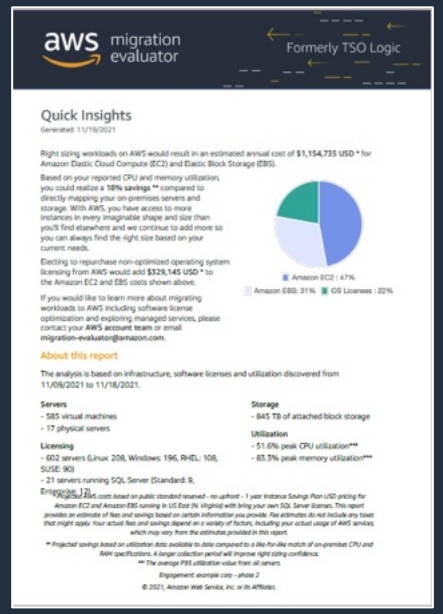


Using a proven framework greatly increases your odds of success and speed to market

Tooling

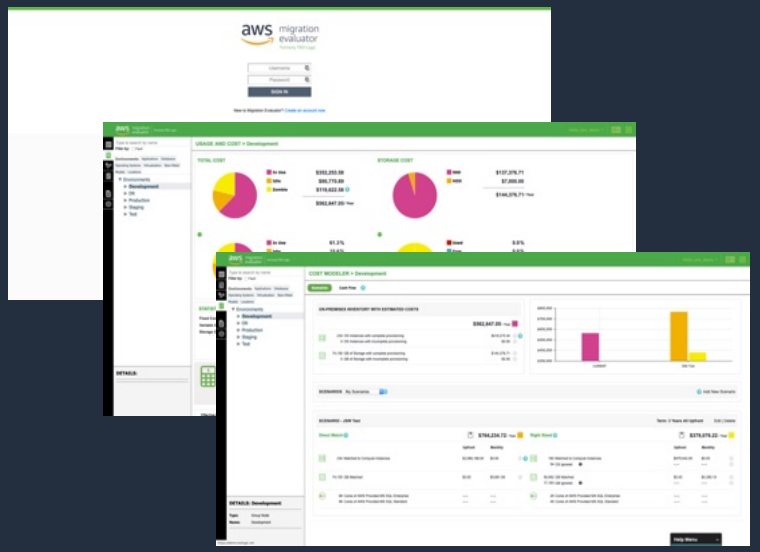
Migration Evaluator

Quick Insights



Directional Business Case

Analytics Engine



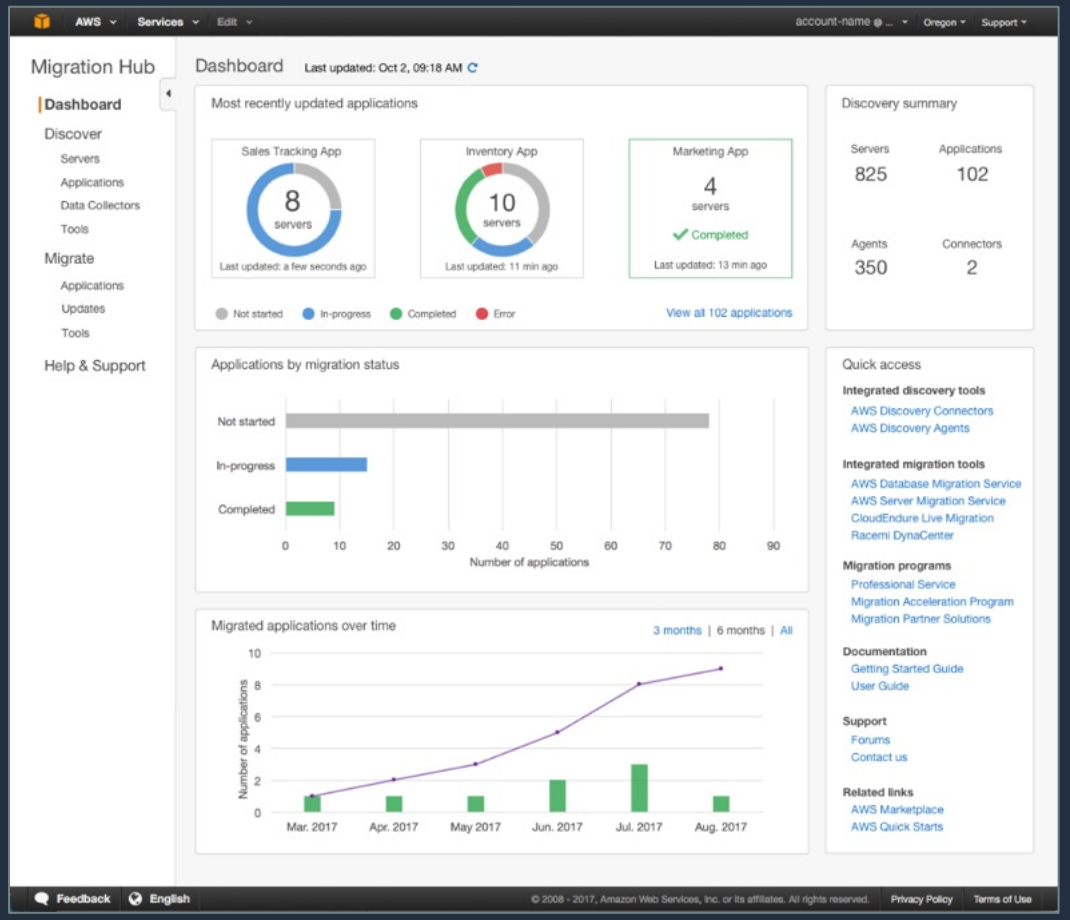
Automated (PDF & Export Available)

Available upon request

Migrate & modernize

AWS Migration Hub

- Track the migration status of your applications across multiple tools in one place
- Discover using AWS ADS or Import discovery data from existing sources
- Use the migration tools that best fit your needs—AWS or Partner tools integrated
- Available to all AWS customers at no additional charge



Click here to learn more

AWS Application Migration Service (MGN)

Flexible

Migrate from any source

Wide range of OS, application, and database support

Suitable for large-scale migrations

Reliable

Robust, predictable, non-disruptive continuous replication

Short cutover windows with minimal downtime

Highly secure

Ease of use

Minimal skill set required to operate

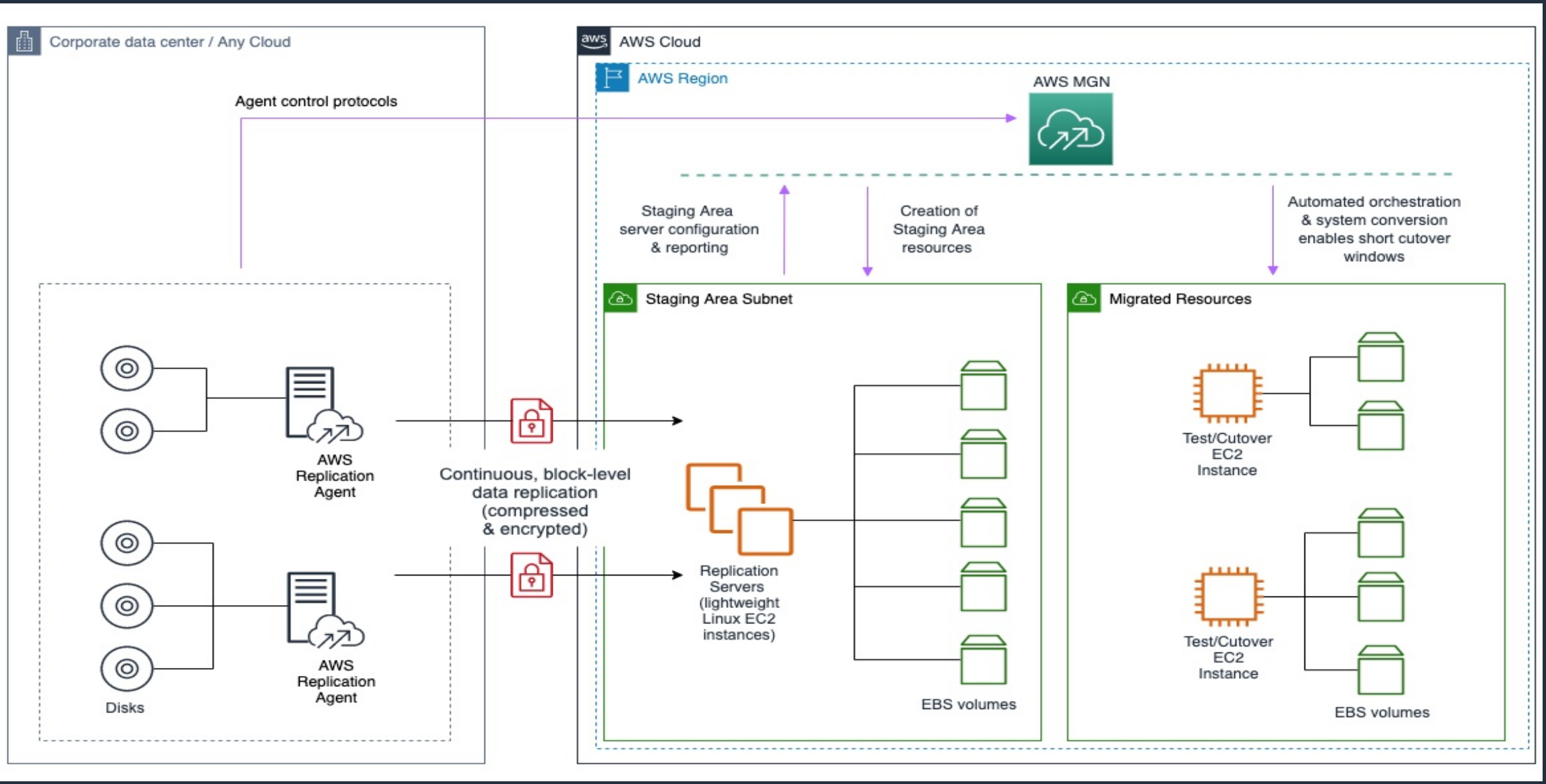
Easy, non-disruptive tests prior to cutover

Easily plugs into migration factories and cloud COEs

Free for 90 days from the installation of the agent on the source machine.

Charged per hour after that period for active replications.

How AWS Application Migration Service Works



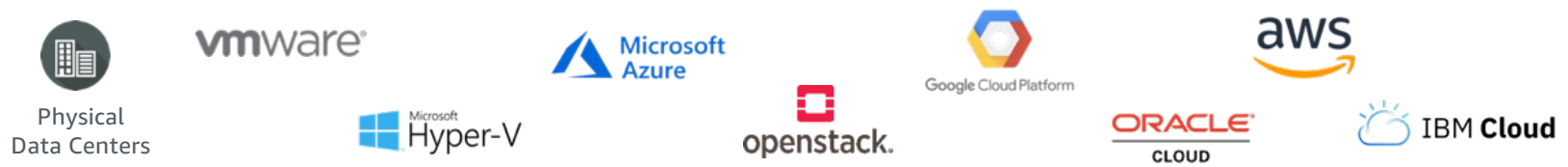
Wide Platform Support*

Any Application

Any Database

x86 Operating Systems

Source Infrastructure



* See documentation or contact Support for a complete list.

aws

Database migration tooling

AWS Schema Conversion Tool (AWS SCT) converts your commercial database and data warehouse schemas to open-source engines or AWS-native services, such as Amazon Aurora and Amazon Redshift

AWS Database Migration Service (AWS DMS) easily and securely migrates and/or replicates your databases and data warehouses to AWS

Infrastructure as Code

Why Infrastructure As Code?

Speed 

Security 

Share & Enforce Best Practices 

Consistency 

Pattern Re-use 

Automation 

Continuous Integration / Continuous Deployment 

Roll Back 

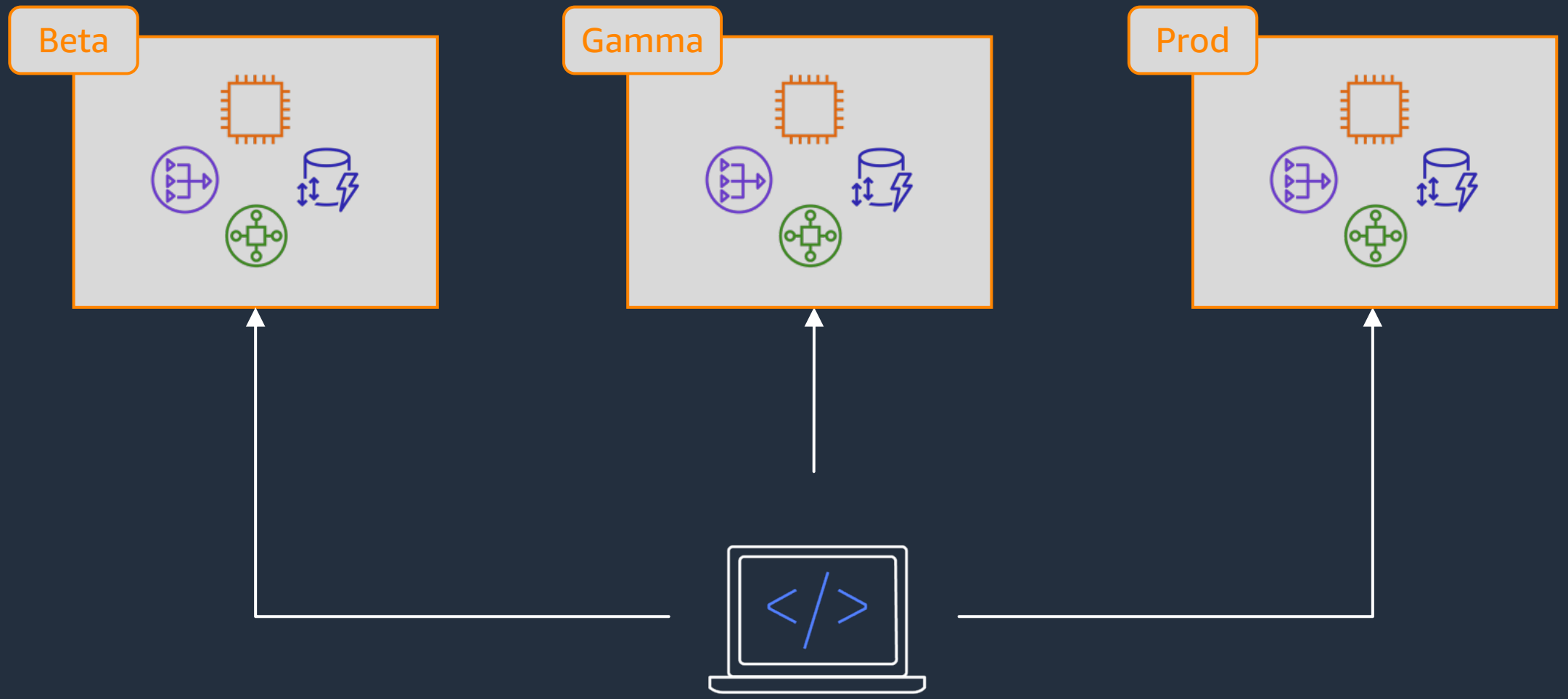
Single Source of Truth 

Lifecycle Management 

Version Control 

Drift Detection 

Consistent deployments



Infrastructure as Code (IaC) – Common Players

- Managing and provisioning of infrastructure through code instead of manual processes.
- Benefits


Self service


Safey


CICD validations


Version control

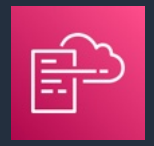

Reduce costs


Speed


Consistency

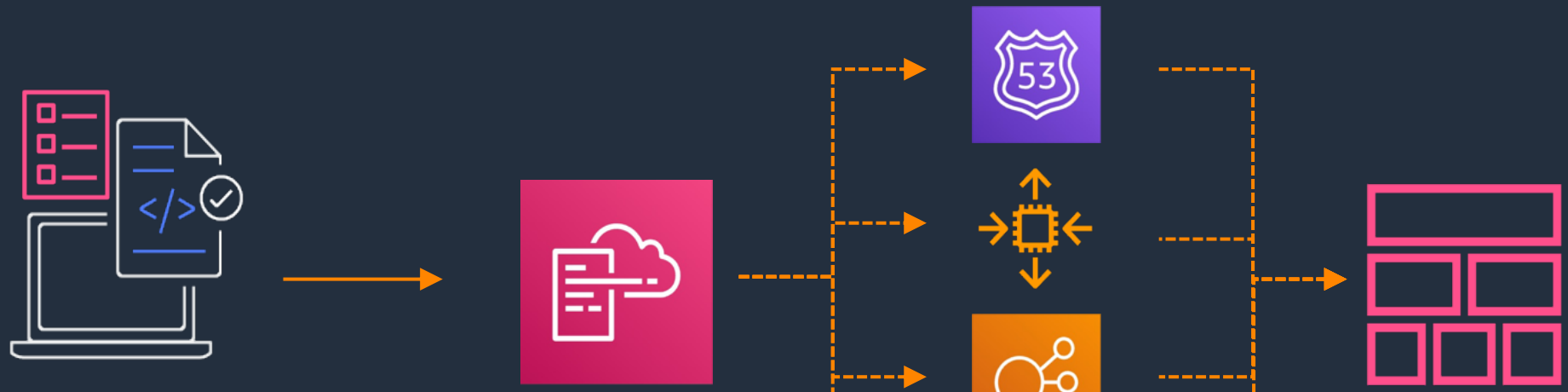

Reusability


Configuration drift


AWS CloudFormation

and many more…

AWS CloudFormation



Code in YAML or JSON directly or use sample templates

Upload local files or from an Amazon S3 bucket

Create stack using API via CloudFormation

Stacks and resources are provisioned as a running environment

AWS Cloudformation - Example

```
178   Pipeline:
179     Type: AWS::CodePipeline::Pipeline
180     Properties:
181       RoleArn: !GetAtt PipelineRole.Arn
182       ArtifactStore:
183         Type: S3
184         Location: !Ref ArtifactBucket
185       Stages:
186         - Name: Source
187           Actions:
188             - Name: GitHub
189               ActionTypeId:
190                 Category: Source
191                 Owner: AWS
192                 Provider: CodeStarSourceConnection
193                 Version: 1
194               Configuration:
195                 ConnectionArn: !Ref GitHubConnectionArn
196                 FullRepositoryId: !Ref RepoId
197                 BranchName: !Ref Branch
198                 DetectChanges: true
199               OutputArtifacts:
200                 - Name: extensions-source
```

AWS Cloud Development Kit (AWS CDK)

A MULTI-LANGUAGE SOFTWARE DEVELOPMENT FRAMEWORK FOR MODELING CLOUD INFRASTRUCTURE AS REUSABLE COMPONENTS

```
class UrlShortener extends Stack {
  constructor(scope: App, id: string, props?: UrlShortenerProps) {
    super(scope, id, props);

    const vpc = new ec2.Vpc(this, 'vpc', { maxAzs: 2 });
    const cluster = new ecs.Cluster(this, 'cluster', { vpc: vpc });
    const service = new patterns.NetworkLoadBalancedFargateService(this, 'sample-app', {
      cluster,
      taskImageOptions: {
        image: ecs.ContainerImage.fromAsset('ping'),
      },
      dom
    });
    // Setup AutoScaling policy
    const scaling = service.service.autoScaleTas
    scaling.scaleOnCpuUtilization('CpuScaling',
      targetUtilizationPercent: 50,
      scaleInCooldown: Duration.seconds(60),
      scaleOutCooldown: Duration.seconds(60)
    });
  }
}
```

domainName
domainZone

(property) patterns.NetworkLoadBalancedServiceBaseProps.domainName?: string | undefined

The domain name for the service, e.g. "api.example.com."

@default

- No domain name.

Familiar
Your language
Just code

Tool support
AutoComplete
Inline documentation

Abstraction
Sane defaults
Reusable classes




Java

Demo

Call to Action

7R decision tree

Cloud Adoption Framework

Large scale migration strategy

Application Migration Service Demo Content

Thank you!

David Surey
suredavi@amazon.de

Add me on Linkedin

Please fill out my survey!