AWS re:Invent

NOV. 29 – DEC. 3, 2021 | LAS VEGAS, NV

XNT307-R2

Strategies for deconstructing your .NET monolith

Hasnat Naveed (he/him)
Senior Product Manager
Amazon Web Services

Runeet Vashisht (he/him)
Senior Solutions Architect
Amazon Web Services

Brian Lewis (he/him)
Principal Solutions Architect
Amazon Web Services

aws

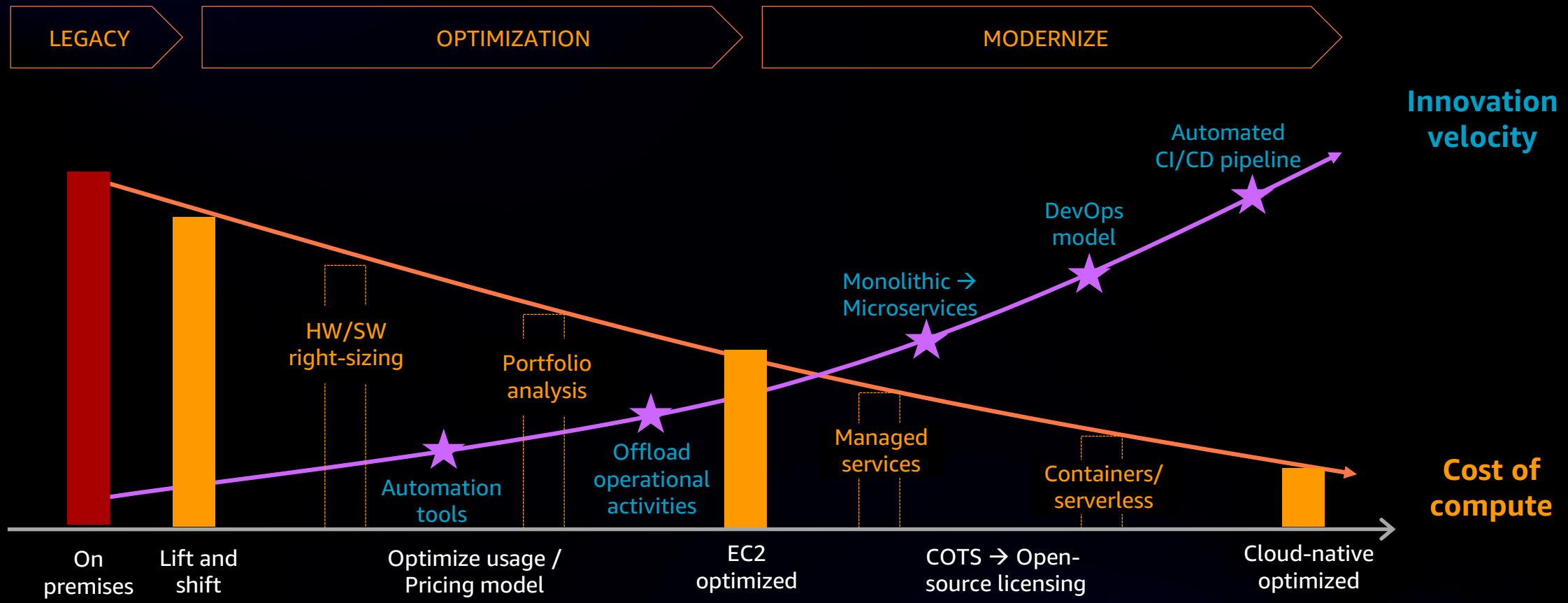
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

- PowerPoint – concepts for discussion (15 min)
  - Overview of modernization all up
  - Crash course on microservices
  - AWS tools that can help .NET application modernization
- Open discussion (45 min)
  - Chalk talk – whiteboard, demo, etc.

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Typical modernization journey overview



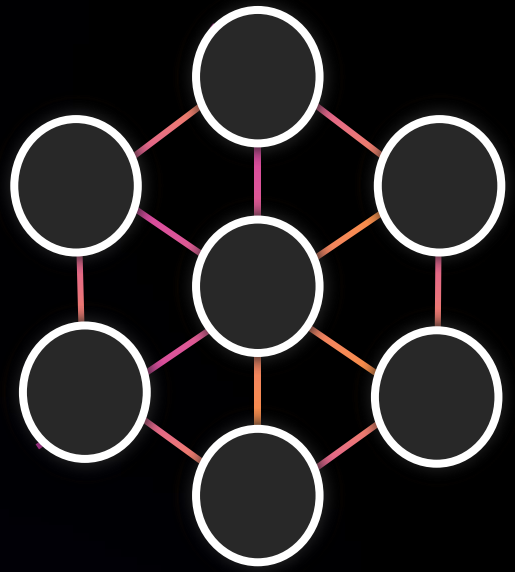
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

When the impact of change is small, release velocity can increase



Monolith

Does everything



Microservices

Does one thing

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Microservices architecture enables CI/CD

Teams that practice CI/CD ship more code faster, and with more confidence*

5x
Lower change failure rate

440x
Faster from commit to deploy

46x
More frequent deployments

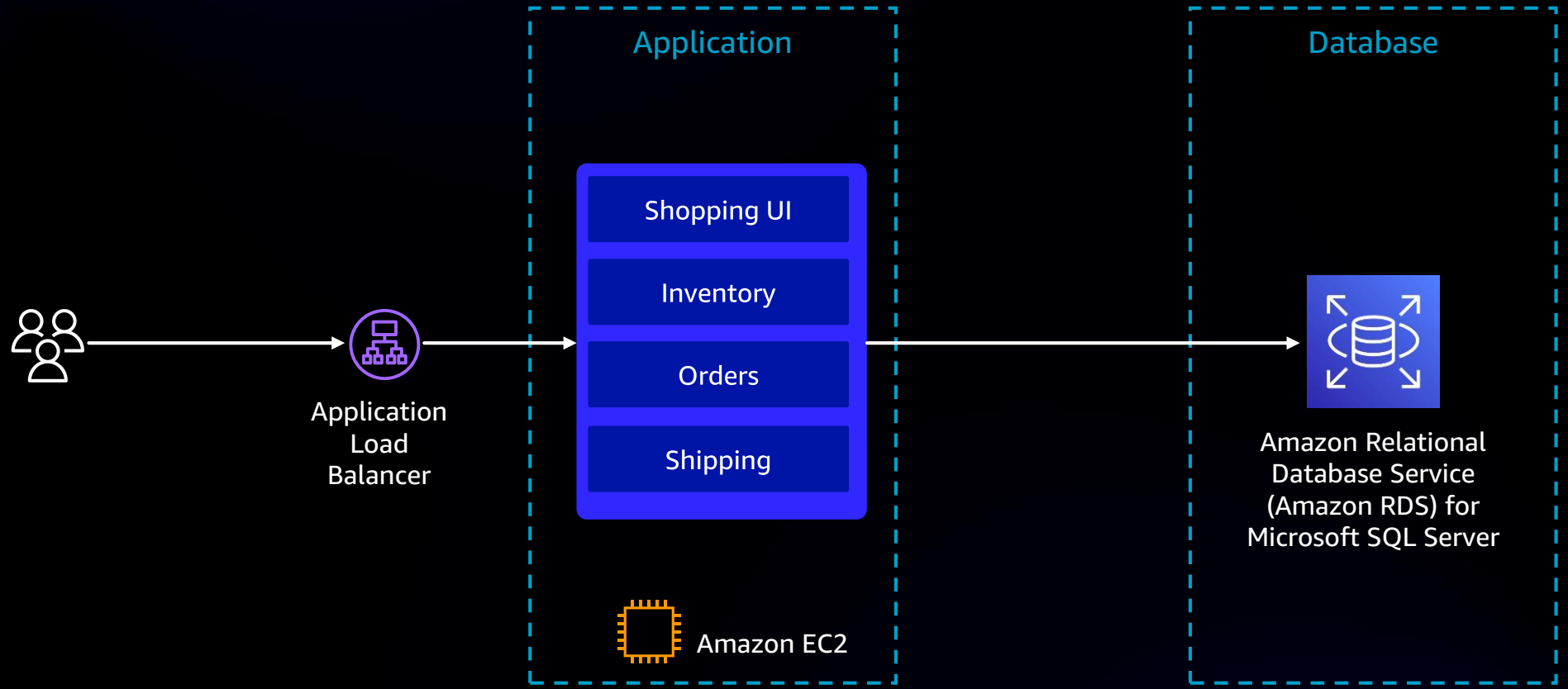
44%
More time spent on new features & code

* Puppet “2021 State of DevOps Report” https://puppet.com/resources/report/2021-state-of-devops-report

aws

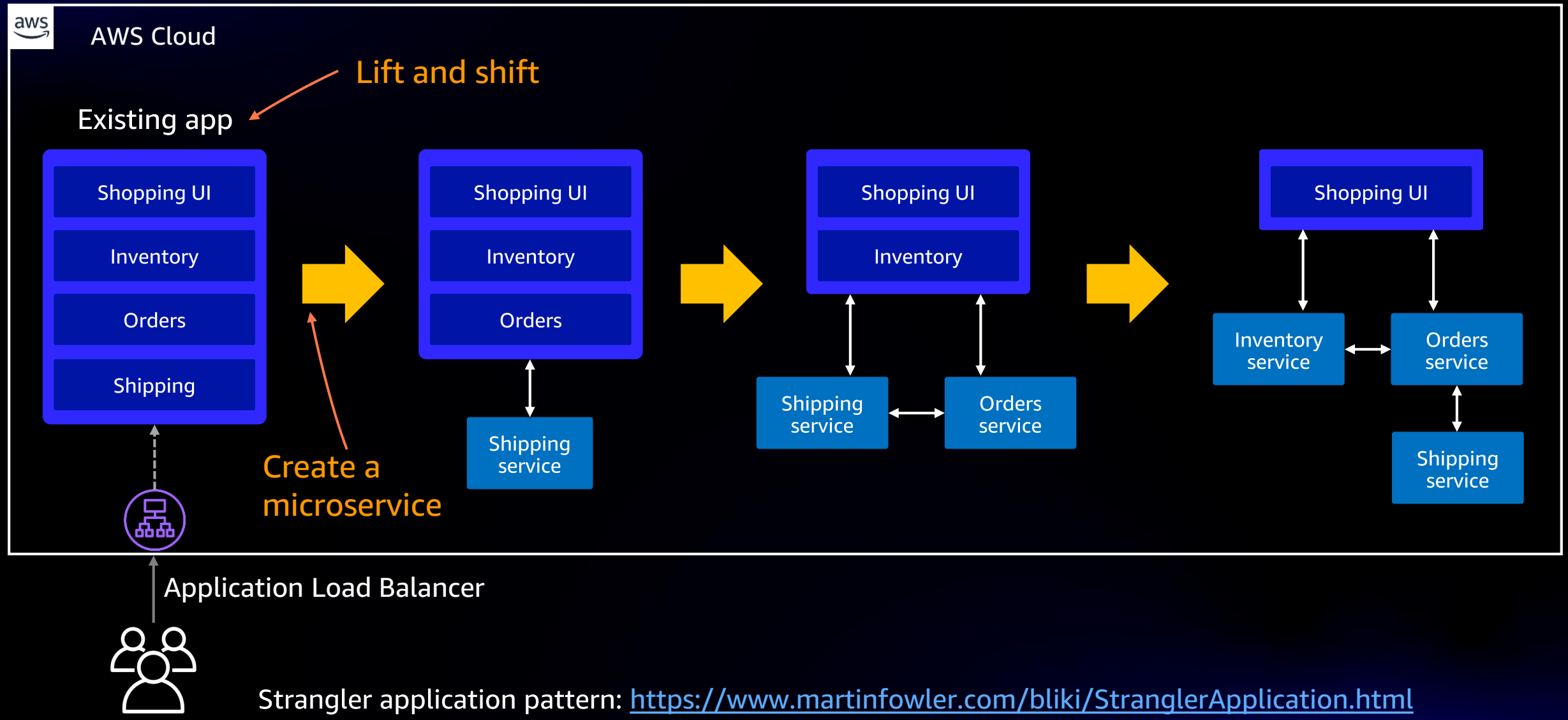
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Monolith architecture



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

How to move from monolith to microservices



Strangler application pattern: https://www.martinfowler.com/bliki/StranglerApplication.html

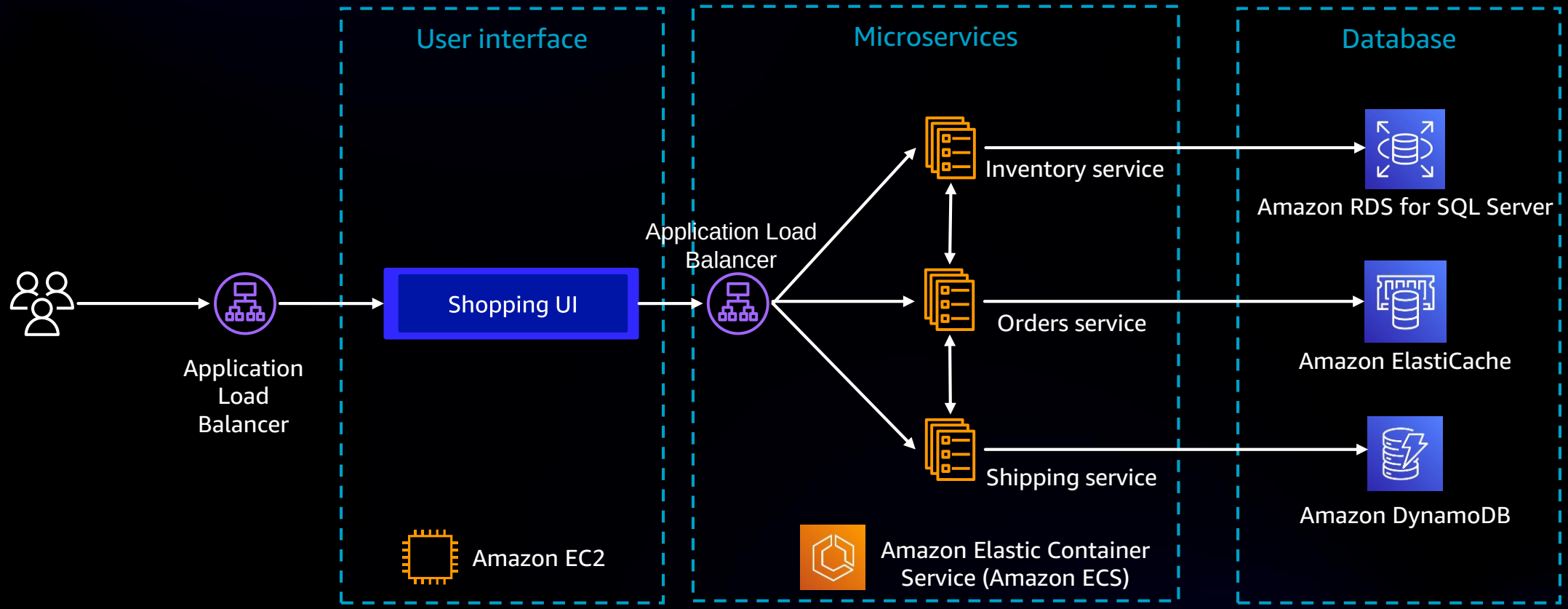
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Migrating from monolith to microservices

1. Identify existing components
2. Identify component dependencies
3. Group interdependent components – refactor
4. Expose component group as an API
5. Break the group as a microservice
6. Refactor – remove redundant code
7. Repeat

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Microservices architecture



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS tools to accelerate

New

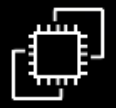
AWS Microservice Extractor for .NET
Extract parts of application as independent services

New

AWS Migration Hub Refactor Spaces
Streamline infrastructure for incremental refactoring

Rehost

MIGRATE
Self-managed
Apps/DBs run on VMs
No code changes



Amazon EC2 Windows instances

AWS Application Migration Service
Replicate VMs into AWS

Replatform

CONTAINERIZE
Automation and CI/CD
Portability and predictability
Minimal code changes





Amazon ECS AWS Fargate Amazon EKS

AWS App2Container
Faster containerization for existing applications

Refactor

.NET
License freedom
Perf & security
Cross-platform
.NET Core

SERVERLESS
Pay as you go
Build fast
Effortless scale



Lambda

Porting Assistant for .NET
Compatibility assessment and assisted refactoring

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What's on your mind?

Let's chat

Some discussion ideas:

- How to start?
- What path to take?
- < Insert your ideas here >



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Hasnat Naveed
hasnav@amazon.com

Runeet Vashisht
runeetv@amazon.com

Brian Lewis
lwbr@amazon.com

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.