Application Modernization using Cloud Native Approach

ESTD

BDSAF

2015

Bangladesh System Admin Day 2023

aws certified Machine Learning Specialty

aws certified Database Specialty

aws certified Solutions Architect Professional

aws certified Security Specialty

aws certified Cloud Practitioner

aws certified Developer Associate

aws certified SysOps Administrator Associate

aws certified Solutions Architect Associate

Mohammad Mizanur Rahman

CTO, Brain Station 23

Co-founder of AWS User Group Bangladesh
AWS Partner Network (APN) Ambassador

aws partner network

Advanced
Consulting
Partner

Key Primitives of a Modern Cloud Native Application?

“…application is container-based”

“…dynamically managed…”

…microservice oriented…”

Why Building Cloud Native Matters

Speed

Scale

Resiliency

Key Building Blocks for Success

Containers + Functions

Cloud

Culture

Key Building Blocks for Success

Containers + Functions

Cloud

Culture

Time to Value

The fast companies are **440x** faster than the slow

We found that, compared to low performers, high performers have:

46x more frequent code deployments

440x faster lead time from commit to deploy

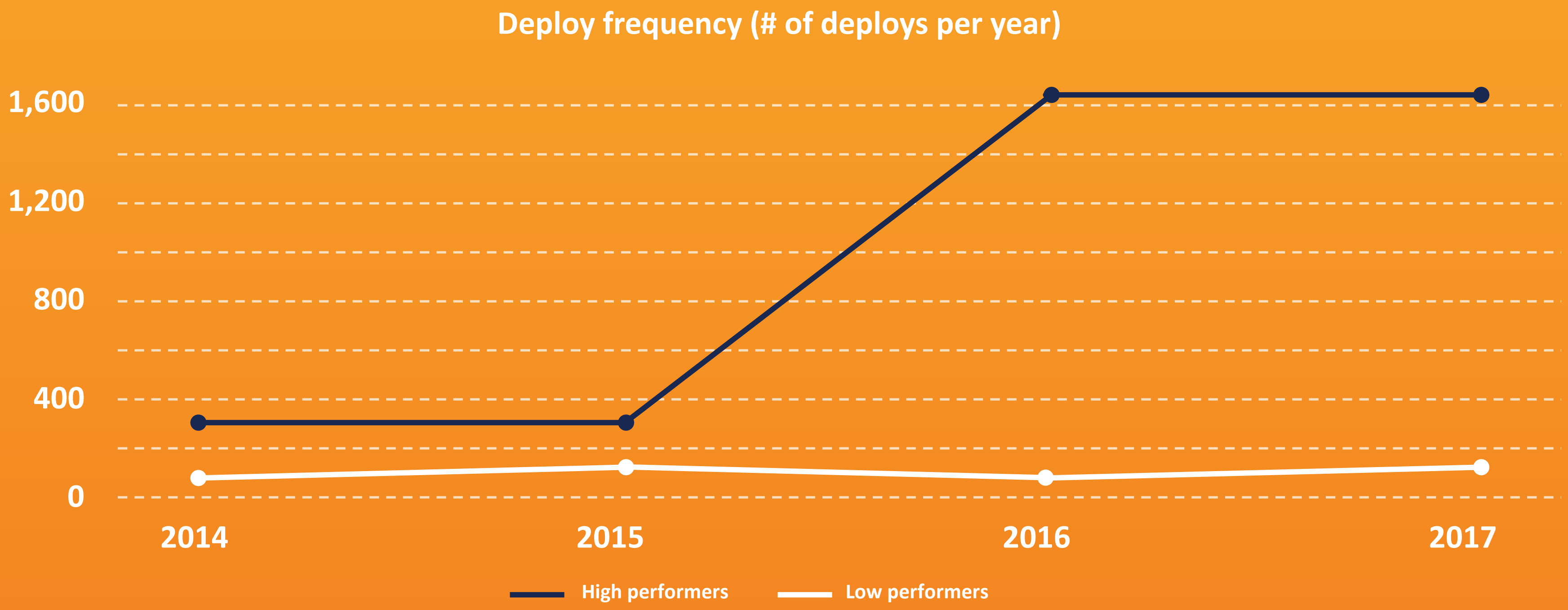
96x faster mean time to recover from downtime

5.0x lower change failure rate (changes are 1/5 as likely to fail)

https://puppet.com/resources/whitepaper/state-of-devops-report

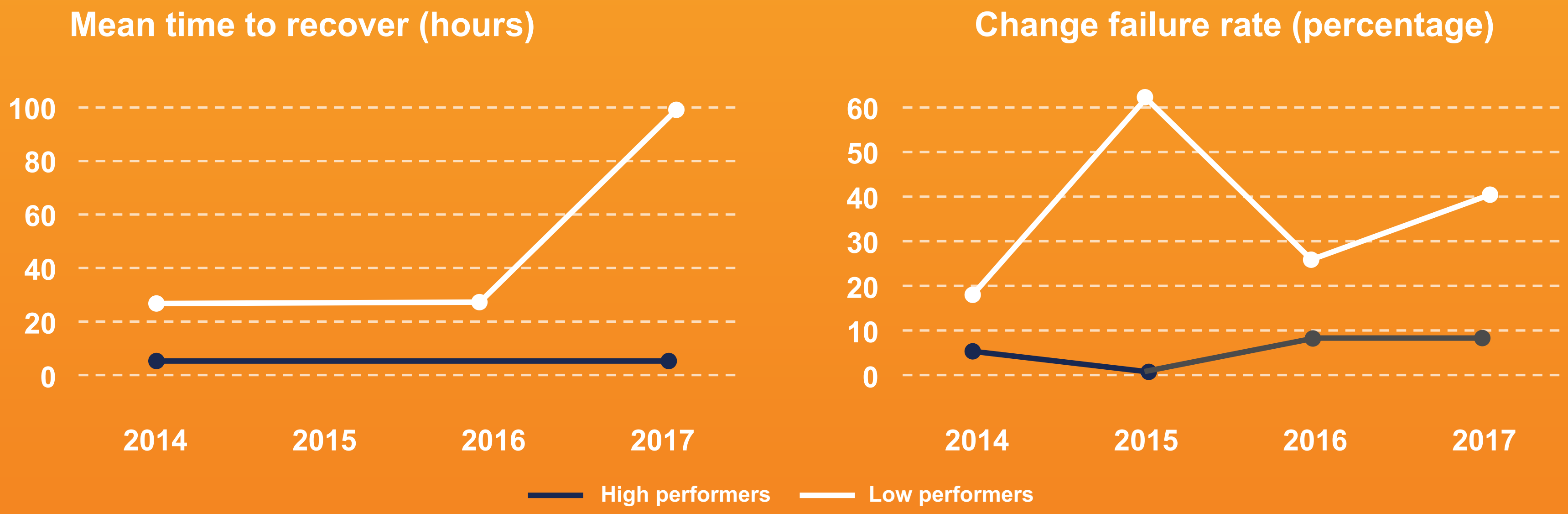
Ship features, not just code

Containers Enable Fast Deployments



https://puppet.com/resources/whitepaper/state-of-devops-report

Containers Enable Immutable Changes/Rollback



https://puppet.com/resources/whitepaper/state-of-devops-report

Cloud Native Principle #1

Cloud Native Applications enable
high functioning organizations to
build and
ship features faster!

Key Building Blocks for Success

Containers + Functions

Cloud

Culture

Cloud Native Architecture

Pay as you go

Self-Service

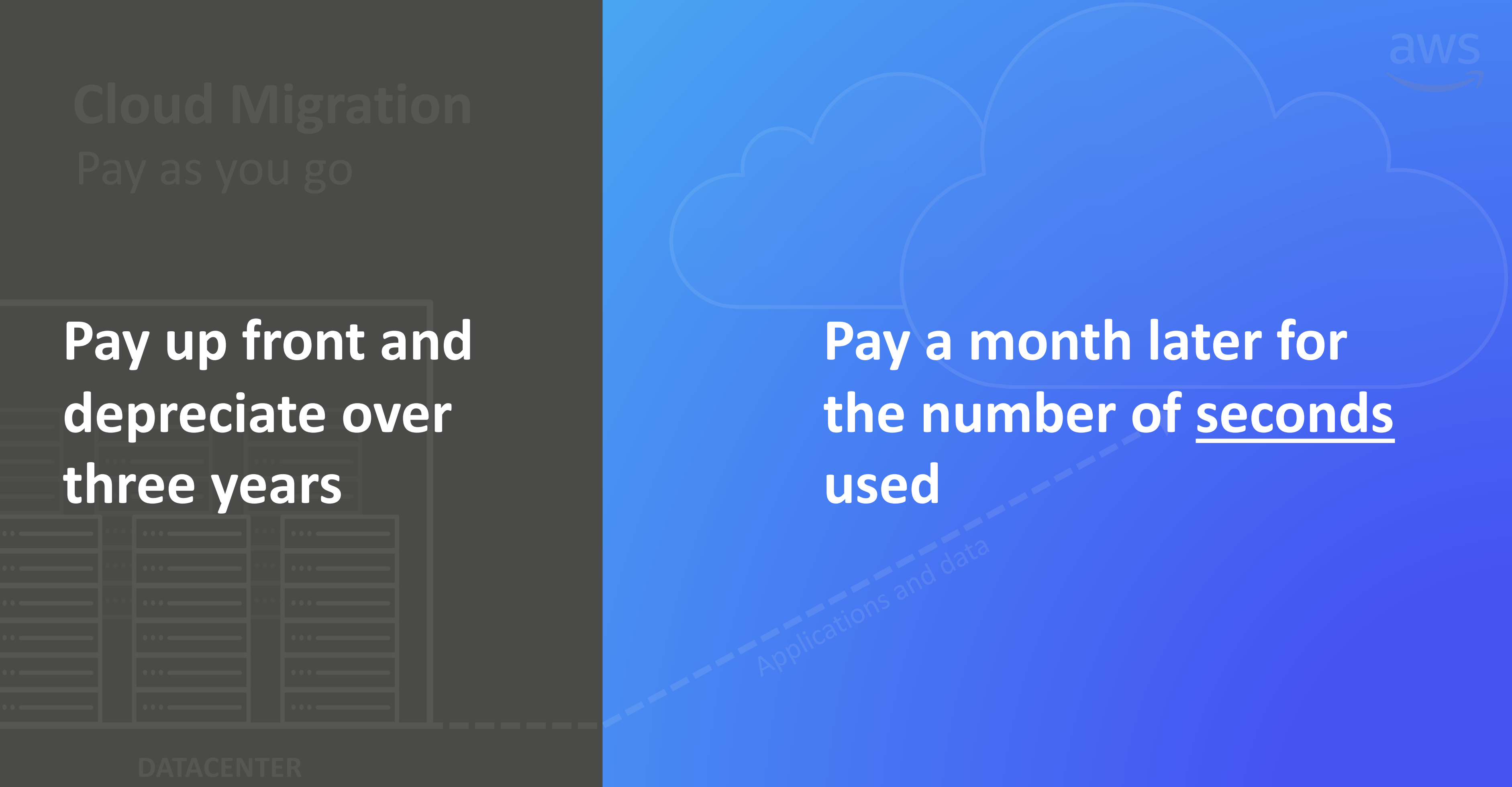
Elastic

Cloud Migration

Pay as you go

Pay up front and depreciate over three years

Pay a month later for the number of seconds used

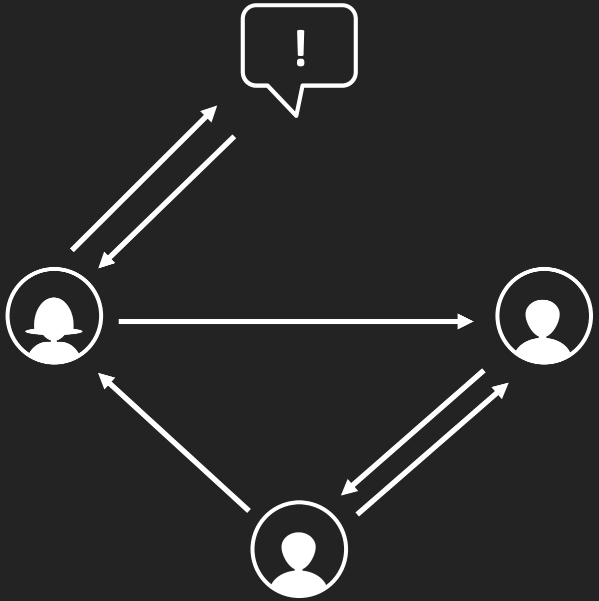


Cloud Native Principle #2

Pay for what you used last month, not what you guess you will need next year.

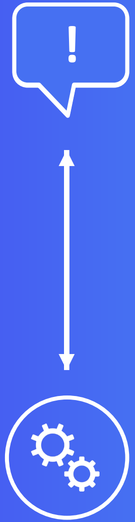
Enable teams to experiment and fail fast, without significant investment.

File tickets and wait for every step

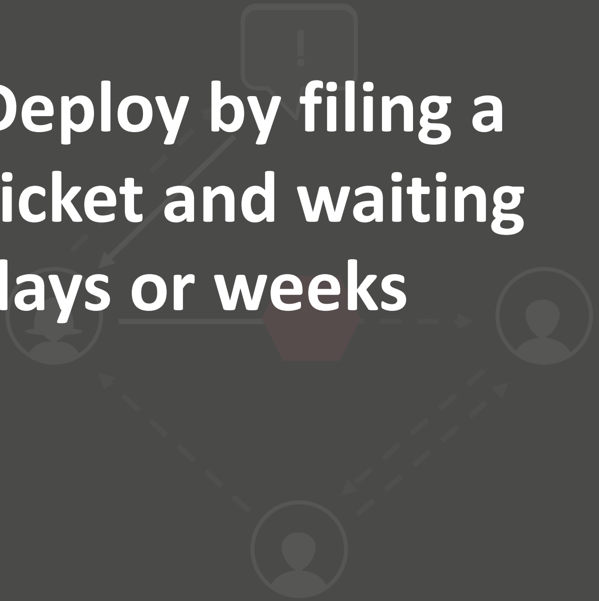


VS

Self service, on-demand, no delays



File tickets and wait for every step



Deploy by filing a ticket and waiting days or weeks

VS

Self service, on-demand, no delays

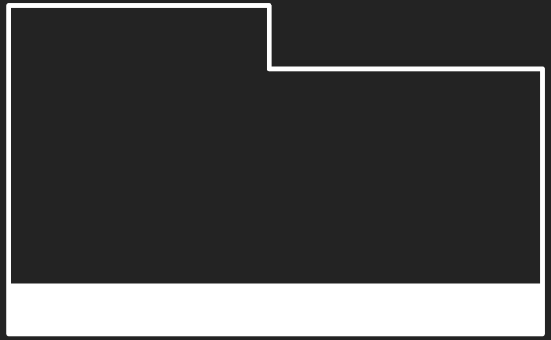
Deploy by making an API call self service within minutes

Cloud Native Principle #3

Self service, API driven, automated.

Move from request tickets at every step to self-service APIs and tools that empower teams.

Elasticity



DATA CENTER

Hard to get over 10% utilization— need extra capacity in case of peak.

CLOUD

Target over 40% utilization— and scale on demand for any size workload.

Cloud Native Principle #4

Turn it off when it’s idle.

Scale for workloads of any size.

Many times higher utilization.

Huge cost savings.

Resiliency

Blast Radius

Loosely Coupled

Geographically Distributed

Microservices limit “blast radius” for software incidents

Build and deploy loosely coupled services.

Enable teams to move fast independently.

Reduce blast radius via service and deployment isolation.

Cloud Native Principle #5

Microservices reduce blast radius, can improve MTTR, and support globally distributed deployment models.

Key Building Blocks for Success

Containers + Functions

Cloud

Culture

“You don’t add innovation to a culture, you get out of its way.”

—Adrian Cockcroft, VP Cloud Architecture Strategy, AWS

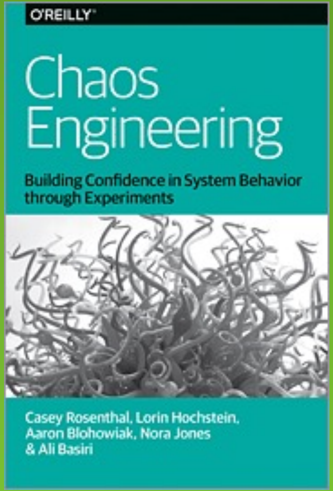
Organization Transformation

“…teams build software that patterns their organizational structure…”

—Conway’s Law

“You build it, you run it.”

—Werner Vogels, VP & CTO Amazon.com


O'REILLY
Chaos Engineering
Building Confidence in System Behavior through Experiments
Casey Rosenthal, Lorin Hochstein, Aaron Blohowiak, Nora Jones & Ali Basiri

"Not what happens ***IF*** it fails,
but what happens ***WHEN*** it fails."

—Nora Jones, Author, and Sr. Chaos Engineer at Netflix

Thank you!

https://www.linkedin.com/in/mizans/