Kubernetes on AWS: Multi-Arch Workloads on AWS Graviton2

nClouds℠

aws

www.nclouds.com

Special OFFERS

nClouds℠

Free Migration Assessment to AWS Graviton2
for all eligible attendees

Multi-Arch Workloads with AWS Graviton2

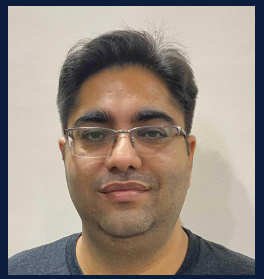
PRESENTERS

Scott Malkie
Specialist Solutions Architect

aws

Marius Ducea
VP of DevOps Practice

nClouds



Navdeep Singh
Senior DevOps Engineer

nClouds

Amazon Web Services (AWS) is the world's most comprehensive and broadly adopted cloud platform, offering over 175 fully featured services from data centers globally. Millions of customers—including the fastest-growing startups, largest enterprises, and leading government agencies—are using AWS to lower costs, become more agile, and innovate faster.

nClouds℠
Deliver Innovation Faster

nClouds is a deeply-credentialed, award-winning provider of AWS and DevOps consulting and implementation services. We are an integrated team of skilled engineers, architects, developers, project managers, and sales & marketing professionals who are passionate about client success, software excellence, and innovation. We enable our clients to deliver innovation faster and create awesome customer experiences.

Trusted by INNOVATIVE BRANDS

nClouds

Multi-Arch Workloads on AWS Graviton2

AGENDA

DETAILS ***(All times PT)***

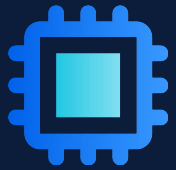
- **11:00 - 11:05 am** - Intro & Session Objectives *by Randy Newell, nClouds*
- **11:05 - 11:25 am** - Intro to AWS Graviton2 on AWS *by Scott Malkie, AWS*
- **11:20 - 11:35 am** - Container-based Workloads on Graviton2 *by Marius Ducea, nClouds*
- **11:35 - 11:50 am** - Demo: Multi-Architecture Containers Using AWS Graviton2 on Amazon EKS with nCodeLibrary *by Navdeep Singh, nClouds*
- **11:50 - 12:00 noon** - Q&A *by AWS and nClouds*

Multi-Arch Workloads on AWS Graviton2

OBJECTIVES

Intro to AWS Graviton2

New Features, Cost & Performance Advantages

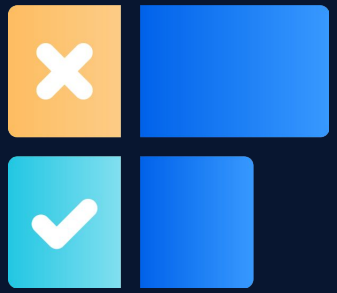


AWS Graviton2 on Amazon EKS

Building Multi-Arch Containers Using AWS Graviton2

nCodeLibrary Demo

Building Amazon EKS/AWS Graviton2-Based Clusters



Poll

Intro to AWS Graviton2

Scott Malkie
Specialist Solutions Architect

Broadest and deepest compute platform choice

CATEGORIES

General purpose

Burstable

Compute intensive

Memory intensive

Storage (High I/O)

Dense storage

GPU compute

Graphics intensive

+

CAPABILITIES

Choice of processor
(AWS, Intel, AMD)

Fast processors
(up to 4.0 GHz)

High memory footprint
(up to 12 TiB)

Instance storage
(HDD, SSD, NVMe)

Accelerated computing
(GPUs and FPGA)

Networking
(up to 100 Gbps)

Bare Metal

Size
(Nano to 32xlarge)

+

OPTIONS

Amazon EBS

Amazon Elastic Inference

=

MORE THAN

300

INSTANCE TYPES

for virtually every workload and business need

The path to Graviton

Custom AWS silicon

Targeted optimizations for cloud-native workloads

Rapidly innovate, build, and iterate on behalf of customers

re:Invent 2018: first instances powered by AWS Graviton

Amazon EC2 A1

Optimized cost and performance for scale-out applications

AWS Graviton Processor with 64-bit Arm Neoverse cores and custom AWS silicon

Applications

Scale-out workloads

Web tier

Containerized microservices

Arm-based software development

Configurations

6 instance sizes

Up to 16 vCPUs, 32GiB memory

Up to 10 Gbps NW, 3.5 Gbps EBS

Bare metal option

Availability

9 Regions

US *(N. Virginia, Oregon, Ohio)*

EU *(Ireland, Frankfurt)*

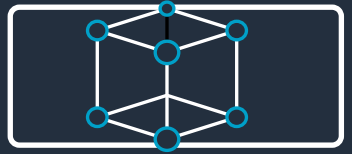
APAC *(Mumbai, Sydney, Tokyo, Singapore)*

Broadening workloads and target applications

Web and gaming servers

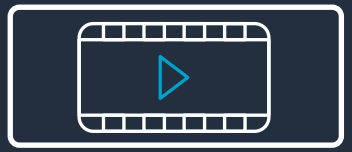
Open-source databases

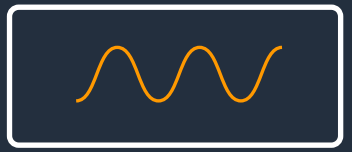
High performance computing



In-memory caches

Media encoding





Electronic design automation

Analytics

Microservices

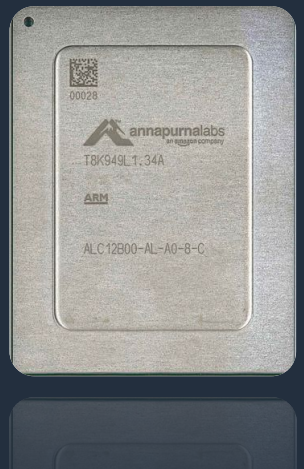
Leap to AWS Graviton2

AWS Graviton processor

- First Arm processor in AWS
- First-class citizen in EC2
- 16nm
- ~5 Billion transistors

AWS Graviton2 processor

- **4x** vCPUs with **7x** CPU performance
- **~2x** performance per vCPU
- **7nm** *(First 7nm chip in EC2)*
- **~30** Billion transistors



AWS Graviton2 vs. AWS Graviton (first generation)

AWS Graviton2 processor

64-bit Arm Neoverse cores

~30B Transistors

7nm technology

7x performance

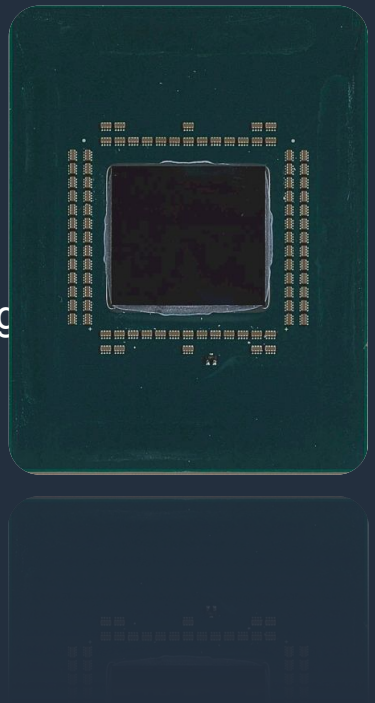
4x compute cores

5x faster memory

2x cache

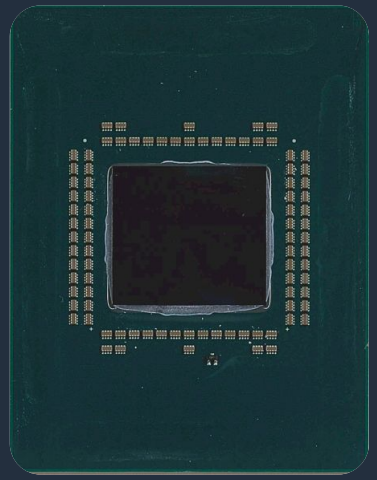
AWS Graviton2: Cores

- Arm® Neoverse™ N1 cores
- Arm v8.2 compliant
- Worked closely with Arm on creation of N1
  - Large 64KB L1 caches and 1MB L2 cache per vCPU
  - Coherent Instruction cache
  - Lower overheads of interrupts, virtualization, and context switching
  - 4-wide front-end with 8-wide dispatch per issue
  - Dual-SIMD units
  - Data types to accelerate ML inference: int8 and fp16
- Every vCPU is a physical core
  - No simultaneous multithreading (SMT)



AWS Graviton2: Interconnect

- 64 cores connected together with a mesh
- ~2TB/s bisection bandwidth
- 32MB last level cache (LLC)
  - With private caches → over 100MB of user-accessible caches
- No NUMA concerns
  - Every core sees the same path to memory and to other cores
- PCIe gen4
  - Provides flexibility for different instance configurations



AWS Graviton2-based instances

Up to **40% better price-performance** over comparable current generation x86-based instances.

M6g	T4g	R6g	C6g
General purpose workloads	Burstable general purpose workloads	Memory-intensive workloads	Compute-intensive workloads

Free Trial

Local NVMe-based SSD storage options are also available:
general purpose (M6gd), compute-optimized (C6gd), and memory-optimized (R6gd)

M, C & R instance types also have bare-metal options:
(M6g.metal, M6gd.metal, C6g.metal, C6gd.metal, R6g.metal, R6gd.metal)

M6g, C6g, R6g available in N. Virginia (us-east-1), Ohio (us-east-2), N. California (us-west-1), Oregon (us-west-2), Frankfurt (eu-central-1), Dublin (eu-west-2), Tokyo (ap-northeast-1), Mumbai (ap-south-1), Singapore (ap-southeast-1, Sydney (ap-southeast-2)

T4g available in N. Virginia (us-east-1), Ohio (us-east-2), Oregon (us-west-2), Frankfurt (eu-central-1), Dublin (eu-west-2), Tokyo (ap-northeast-1), Mumbai (ap-south-1)

Lower TCO with Graviton2-powered instances

Highest performance
in their instance families

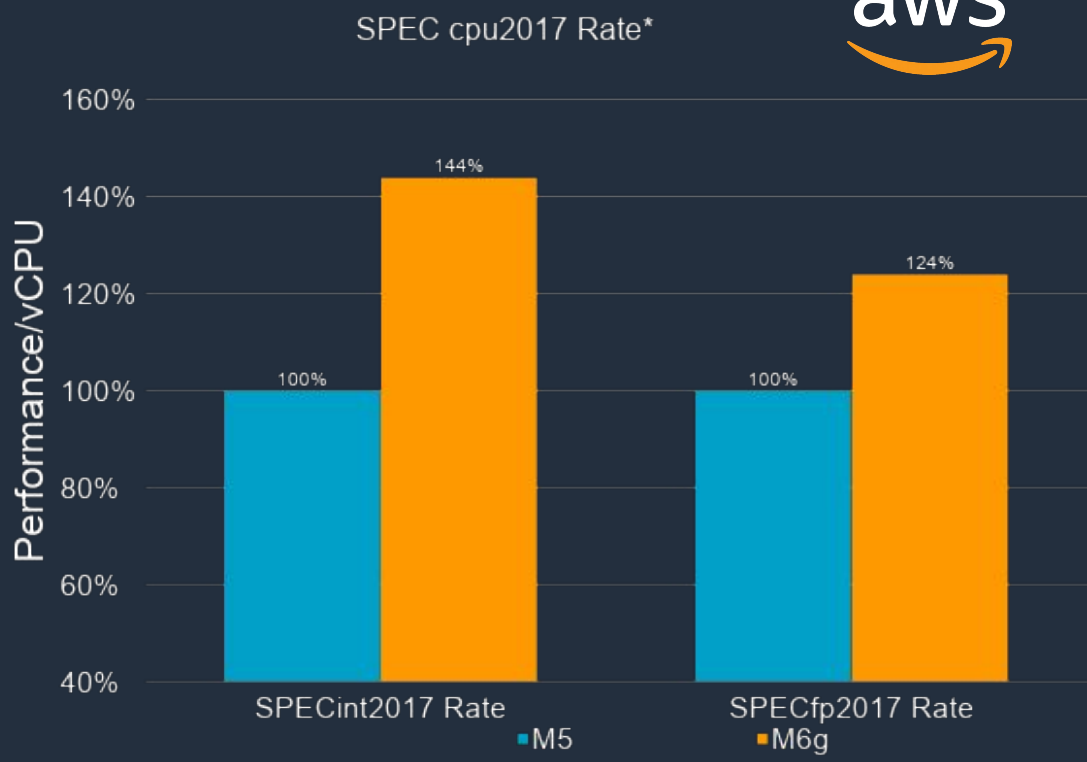
20% lower cost
vs same-sized comparable instances

Up to 40% better price-performance
vs comparable instances

Best price-performance within their instance families

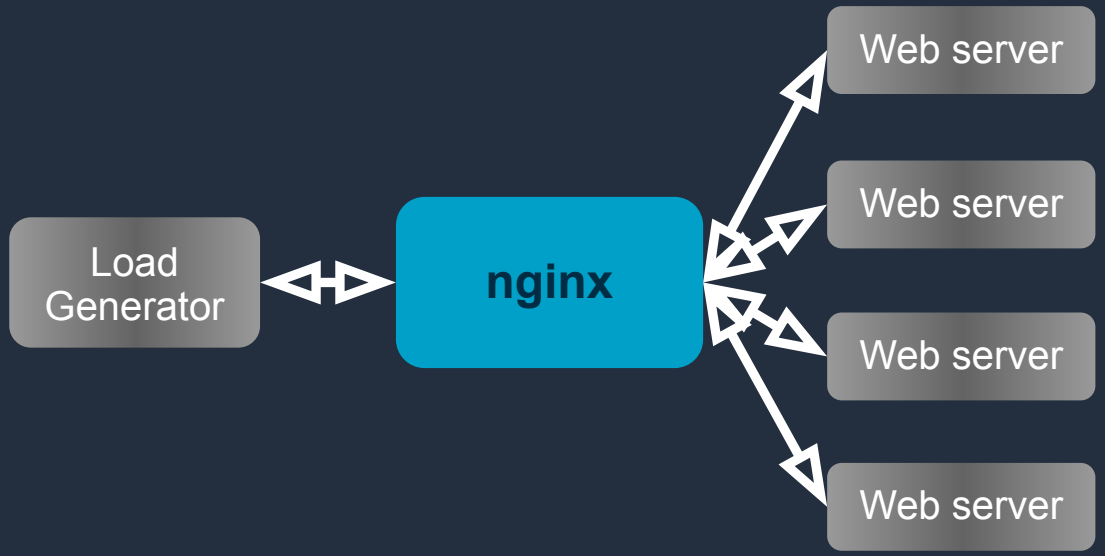
SPEC cpu2017

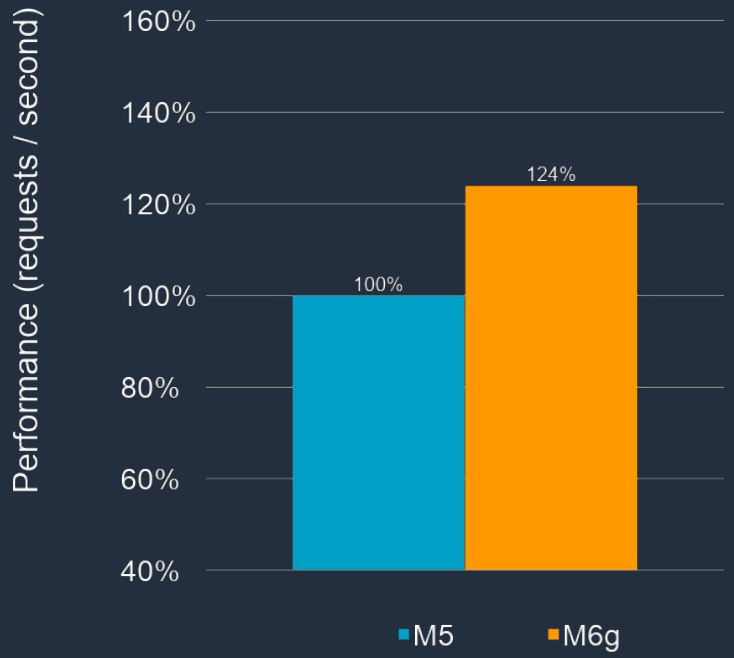
- Industry standard CPU intensive benchmark
- Runs on all vCPUs concurrently
- Compares performance by vCPU



** All SPEC scores estimates, compiled with gcc v9 -O3 -march=native, run on largest single-socket size for each instance type tested.*

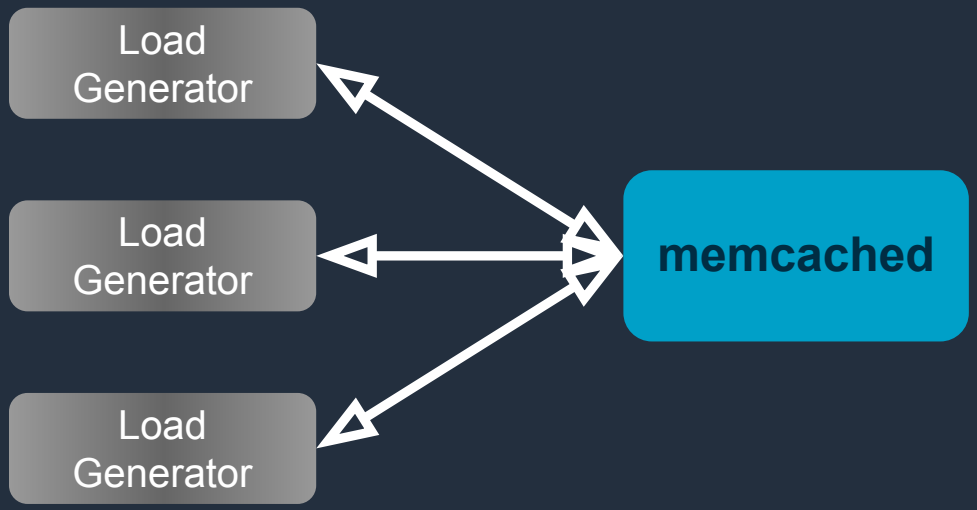
Load Balancing with NGiNX

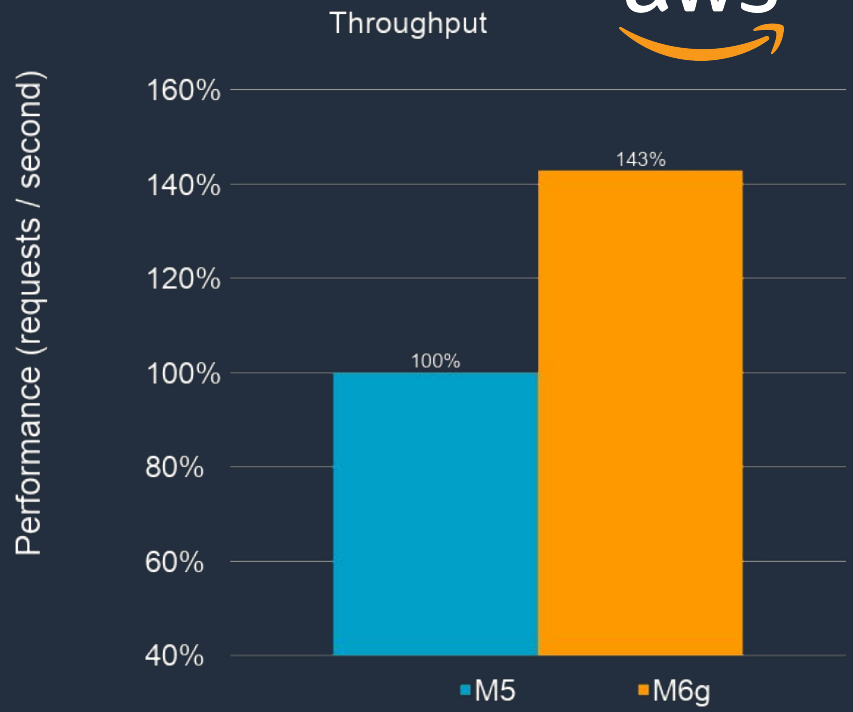




NGiNX v1.15.9, 512 clients, 128 GET/POST payloads, all HTTPS connections, AES128-GCM-SHA256, OpenSSL 1.1.1, 4 target machines, all tests run on 4xl size; load generator c5.9xl; web servers c5.4xls; All servers run in a cluster placement group.

Caching with Memcached





Memcached v1.5.16, 16B keys, 128B values, 7.8M KV-pairs, 576 connections for load generation from 2x c5.9xlarge instances; 16 additional connections measuring latency from 1 additional c5.9xlarge, each connection maintains 4096 outstanding requests; All servers in a cluster placement group.

AWS Graviton software ecosystem momentum

Operating Systems

- Amazon Linux 2
- ubuntu 18.04LTS, 20.04LTS
- SLES 15
- Red Hat Enterprise Linux 8.2+
- CentOS
- Debian
- fedora
- FreeBSD
- NetBSD

Containers

- Docker Desktop Community
  Docker Enterprise Engine
- Amazon Elastic Container Service (Amazon ECS)
- Amazon Elastic Container Service for Kubernetes (Amazon EKS)
- Amazon Elastic Container Registry (Amazon ECR)
- Firecracker (MicroVMs)
  (github.com/firecracker-microvm)
- Bottlerocket (Container OS)
  (github.com/bottlerocket-os)

Tools and software

- AWS Marketplace
- AWS Systems Manager
- Amazon CloudWatch
- Amazon Inspector
- AWS Batch
- AWS Cloud9
- AWS CodeBuild
- AWS CodeCommit
- AWS CodePipeline
- Amazon Corretto OpenJDK

Graviton2: Growing support in managed services

ElastiCache: Up to a 45% price/performance improvement over previous generation instances. Graviton2 instances are now the default choice.

Arm64: Languages, Toolkits, and Runtimes

- Interpreted and compiled-bytecode languages can run without modification
  - Python, Java, Ruby, PHP, Node.js, many others
  - .NET Core supports Linux and arm64
  - Typically “just works”
- Compiled applications will need to be recompiled for arm64
  - Every major compiler supports arm64
  - C, C++, Go , Rust all support arm64
- Most AWS Tools and SDKs support Graviton2 transparently:
  - AWS CLI v1, AWS CLI v2, CloudWatch agent, SSM agent
  - SDKs for C/C++, node.js, Python, Go, Java, .NET
- Currently no Windows or GPU support on Graviton or Graviton2-based instances

AWS Graviton2 getting started guide on GitHub

https://github.com/aws/aws-graviton-getting-started

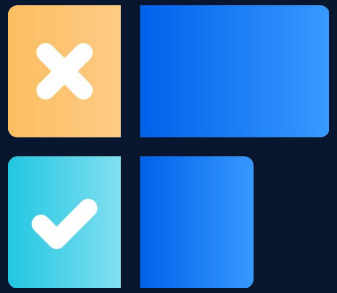
This guide has been assembled by our Graviton team and is designed to help customers transition and optimize their applications.

It covers various languages and libraries, and includes tips and tricks for each.

In general, using latest versions of operating systems, compilers, and language runtimes will provide access to latest Arm64 improvements and optimizations.

Summary

- Amazon EC2 M6g, C6g, R6g, and T4g instances–powered by AWS Graviton2 processors–provide **up to 40% higher price-performance** over comparable x86-based instances.
- Graviton2-powered instances power **a broad spectrum of workloads**: application servers, open source databases, in-memory caches, microservices, gaming servers, electronic design automation, high-performance computing, and video encoding.
- Most Linux and open source-based applications can **easily run on multiple processor architectures** and are well suited for the new instance types.
- These instances are **supported by popular Linux distributions** and an extensive ecosystem of Independent Software Vendors (ISVs).
- Finally, **M6g, C6g, R6g, and T4g instances are available now**, along with Local NVMe (**M6gd, C6gd, R6gd**), and bare-metal variants.



Poll

Running Multi-Arch Containers Using AWS Graviton2 on Amazon EKS

Marius Ducea
VP of DevOps Practice
nClouds

Container-Based Workloads on AWS GRAVITON2

- The AWS Graviton2 processors have been optimized and can be considered ideal for container-based workloads.

Preparing for
AWS GRAVITON2

- The first step for leveraging the benefits of Graviton-based instances as container hosts is to ensure all software dependencies support the arm64 architecture.

STAGE ONE:

Identify Dependencies & Status

- Identify all libraries, dependencies, and agents in your production environment
  - Language-standard and open-source libraries.
  - Paid-for proprietary or commercial libraries.
  - Monitoring, logging, or security daemons or agents.

STAGE ONE:

Identify Dependencies & Status

- Check arm64 support status
  - Look for existing arm64 binaries, or architecture-agnostic source code (OSS).
  - Ask vendor or provider for roadmap towards arm64 support.
  - Reach out to your AWS Account Team or nClouds team who can help investigate or escalate.

STAGE TWO:

Test, Infrastructure & Deployment

- Test! All tests should support multiple architectures, and that may mean writing new tests.
- This is especially pertinent if you had to recompile your application.
    - Perform the usual range of unit testing, acceptance testing, and pre-prod testing.

STAGE TWO:

Test, Infrastructure & Deployment

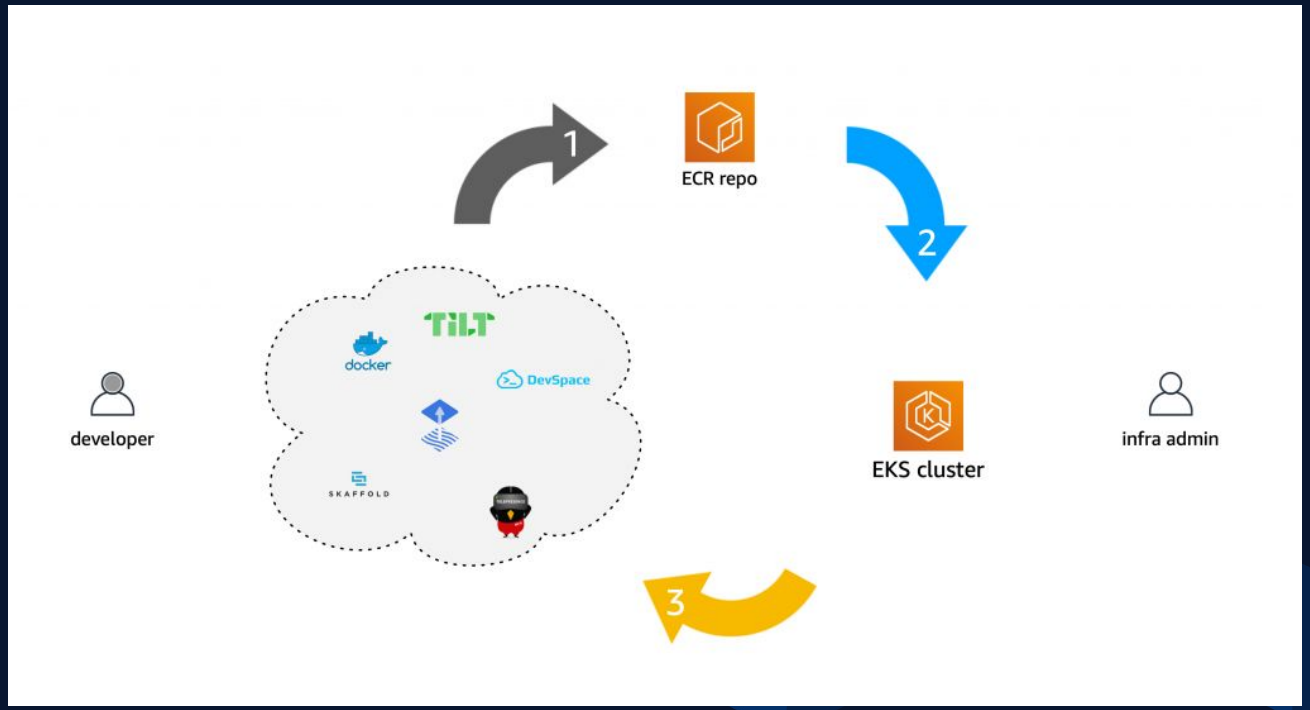
- Update your infrastructure-as-code resources, leveraging tools like Terraform or AWS CloudFormation/CDK, to provision your application to arm64 instances.
    - This will likely be a simple change, modifying the instance type, AMI, and user data to reflect the Graviton2 instances.

STAGE TWO:

Test, Infrastructure & Deployment

- Deploy via a Green/Blue method. Either manually (or via existing CI/CD tooling) create your new arm64-based stack alongside your existing stack, then leverage weighted routing to send a small percentage of requests to the new stack.
- Monitor error rates, user behavior, load, and other critical factors in order to determine the health of the ported application in production.

Multi-Arch across development & deployment



Creating Multi-Arch Container Images

- Docker Buildx
- Using a CI/CD build pipeline like Amazon CodePipeline or CircleCI.

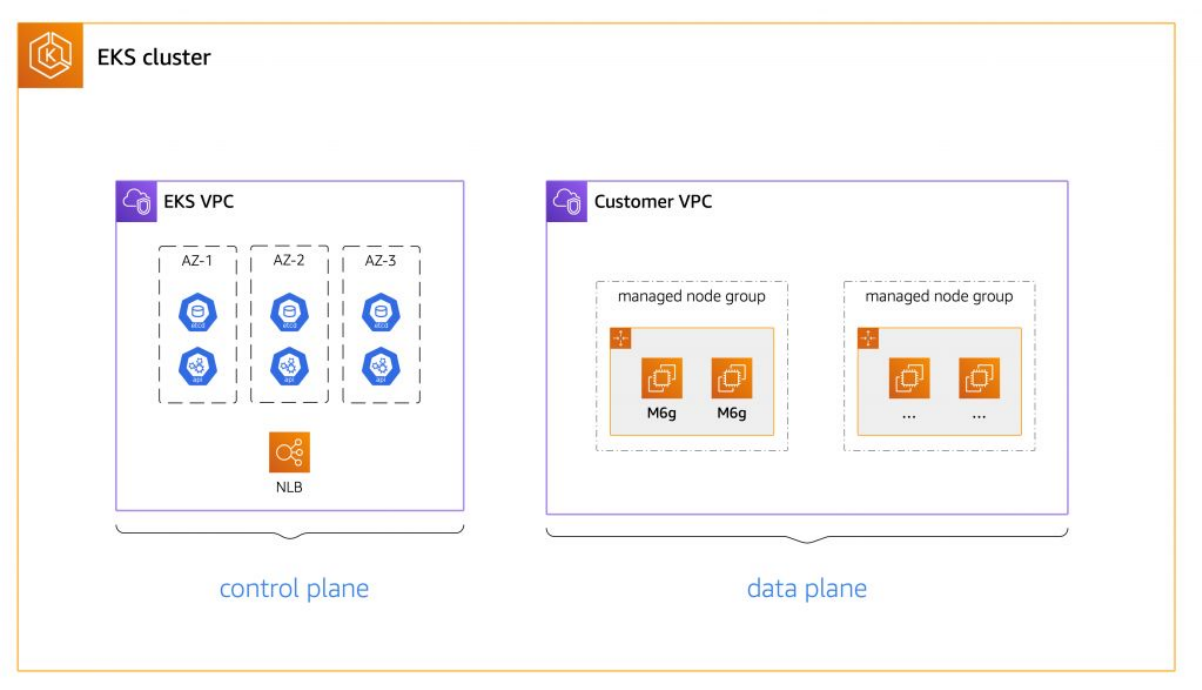
Multi-Arch Container Image Repositories

- The major container image repositories, including Docker Hub, Quay, and Amazon Elastic Container Registry (ECR) all support multi-arch images.

Graviton2 and AWS
Container Services

- Amazon ECS and Amazon EKS both support AWS Graviton and AWS Graviton2 instances, including mixed x86 and Arm clusters.
- Graviton2 on EKS - generally available since August 2020.
- EC2 Arm-based instances are supported on new and existing clusters running Kubernetes version 1.15 and above.

Hybrid
EKS Cluster



Deploying to AWS Graviton2
Container Services

- Once we have the multi-arch docker images available in our registry, we can deploy these in the same way as any docker images.

nCodeLibrary

nClouds

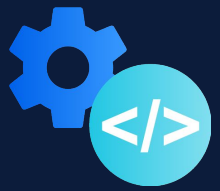
Demo: Multi-Architecture Containers Using AWS Graviton2 on Amazon EKS with nCodeLibrary

Navdeep Singh
Senior DevOps Engineer

nClouds

How do we build Amazon EKS/AWS Graviton2-Based Clusters

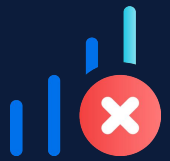
Manually create resources?



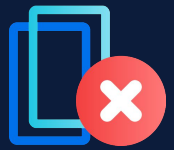
Infrastructure as Code (IaC)

...

Manual
PROVISIONING



Inconsistent



Hard to replicate environments

Prone to errors

Who made this?

Infrastructure as Code

IaC

Consistent

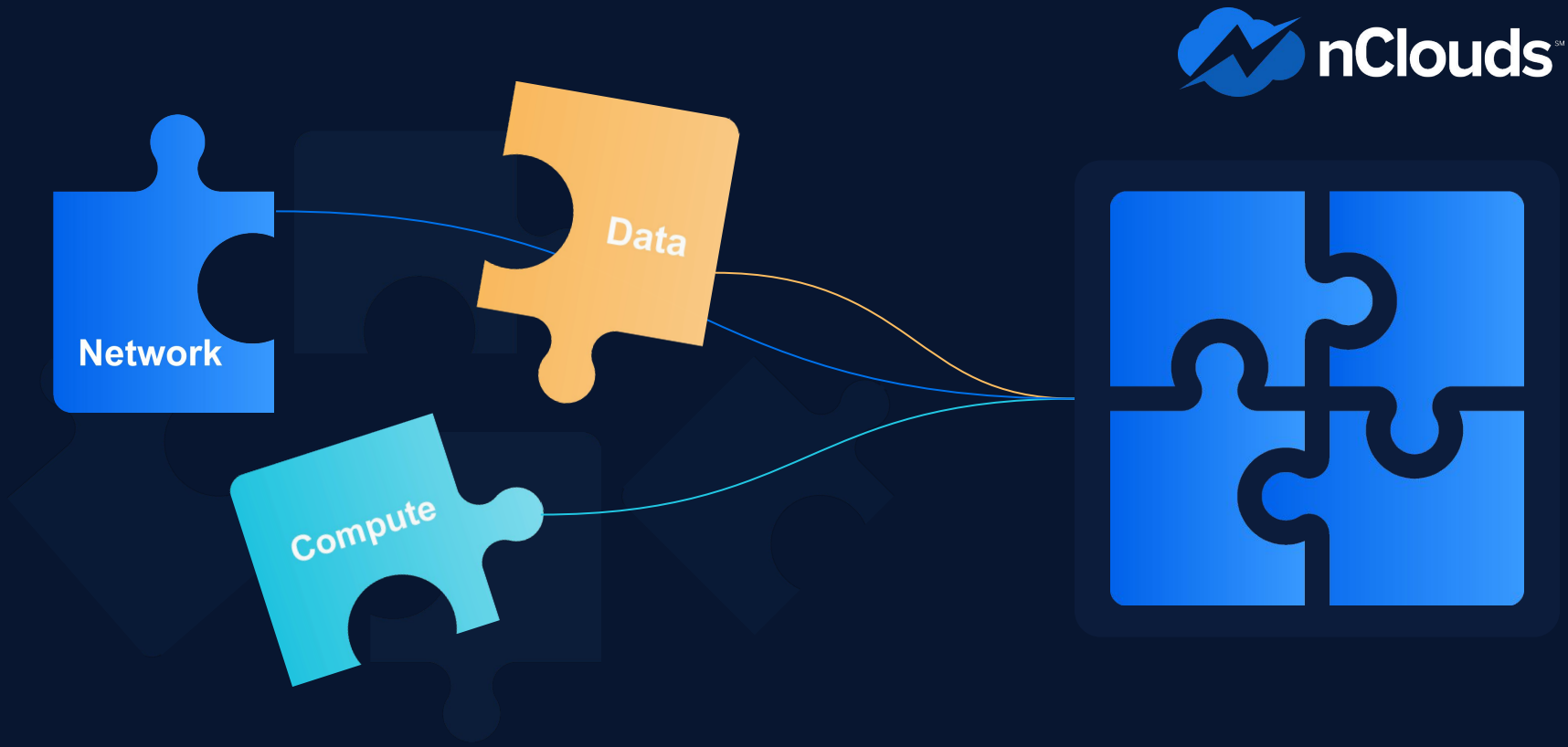
Easy to replicate

Parameter-izable

….

nCodeLibrary

nClouds Infrastructure as Code

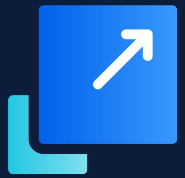


Building Blocks

Bundles

nClouds℠

Test Cases



Scale

Code Standards

Demo

Q&A

Multi-Arch Workloads with AWS Graviton2

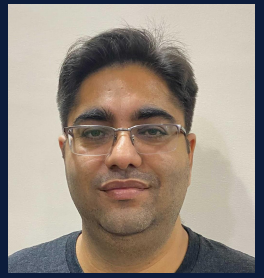
PRESENTERS

Scott Malkie
Specialist Solutions Architect

aws

Marius Ducea
VP of DevOps Practice

nClouds



Navdeep Singh
Senior DevOps Engineer

nClouds

Kubernetes on AWS:
Advanced Networking

Wednesday, November 18, 11 am PT

REGISTER HERE

nClouds℠ aws

www.nclouds.com