AWS re:Invent

NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

XNT401

Boosting .NET Application Performance with Arm64 and Graviton3

Kirk Davis (he/him)

Principal Solutions Architect
AWS

Sreelaxmi Pai (she/her)

Principal Application Architect
AWS

Related sessions

XNT305 (Workshop)

Modernize .NET Framework applications with AWS tools

Wednesday, November 30, 2:30 pm – 4:30 pm

Mandalay Bay , Level 1 North, Islander C

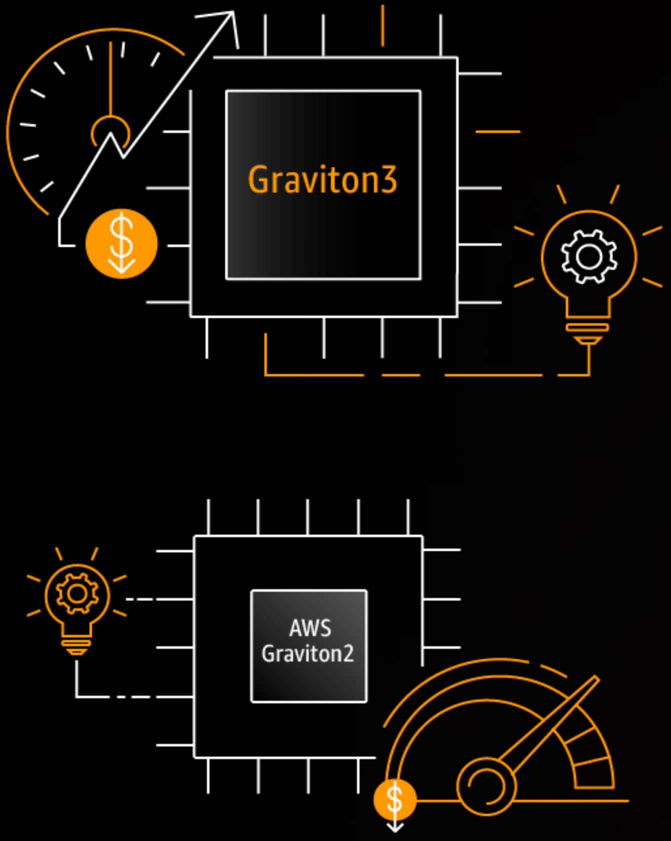
CMP411 (Chalk Talk)

Learnings from developers who adopted AWS Graviton

Wednesday, November 30, 7 pm – 8 pm

Wynn, Level 1, La Tache 1

AWS Graviton Processors



Custom AWS silicon with 64-bit Arm processor cores

Targeted optimizations for cloud-native workloads

Rapidly innovate, build, and iterate on behalf of customers

Components of AWS Graviton-based instances

Graviton Processors

- Exceptional performance
- Reduced costs

Nitro Security Chip

- Integrated into motherboard
- Protects hardware resources

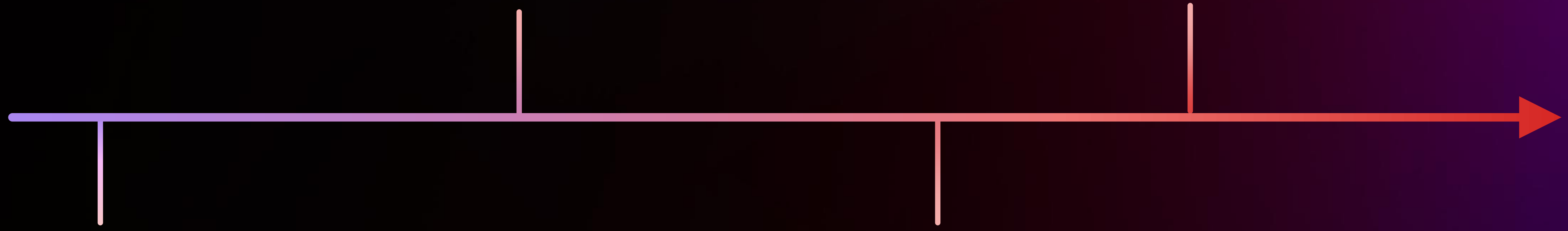
Nitro Card

- Amazon Elastic Block Store
- Elastic Network Adapter
- Monitoring and security

Nitro Hypervisor

- Lightweight hypervisor
- Memory and CPU allocation
- Bare-metal-like performance

History of AWS Graviton processors



Original Graviton instances launched in 2018 (A1 instance family)

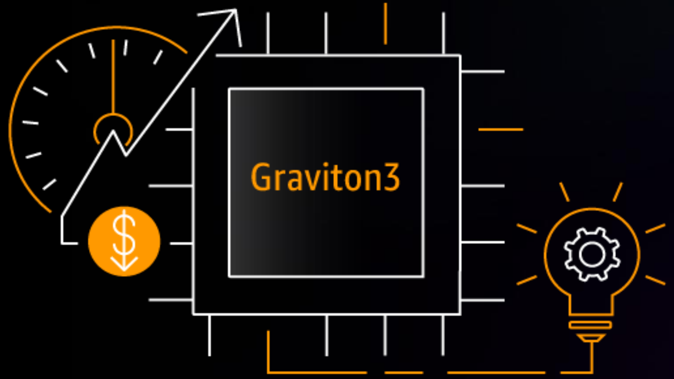
Graviton2 was announced at re:Invent 2019

Graviton3 was announced at re:Invent 2021

Graviton3 instances go GA H1 2022

Graviton instances use “g” suffix: t4g.large, c7g.2xlarge

AWS Graviton3 – best price/performance



Graviton2 had up to 40% better price-performance vs comparable x86-64 instances

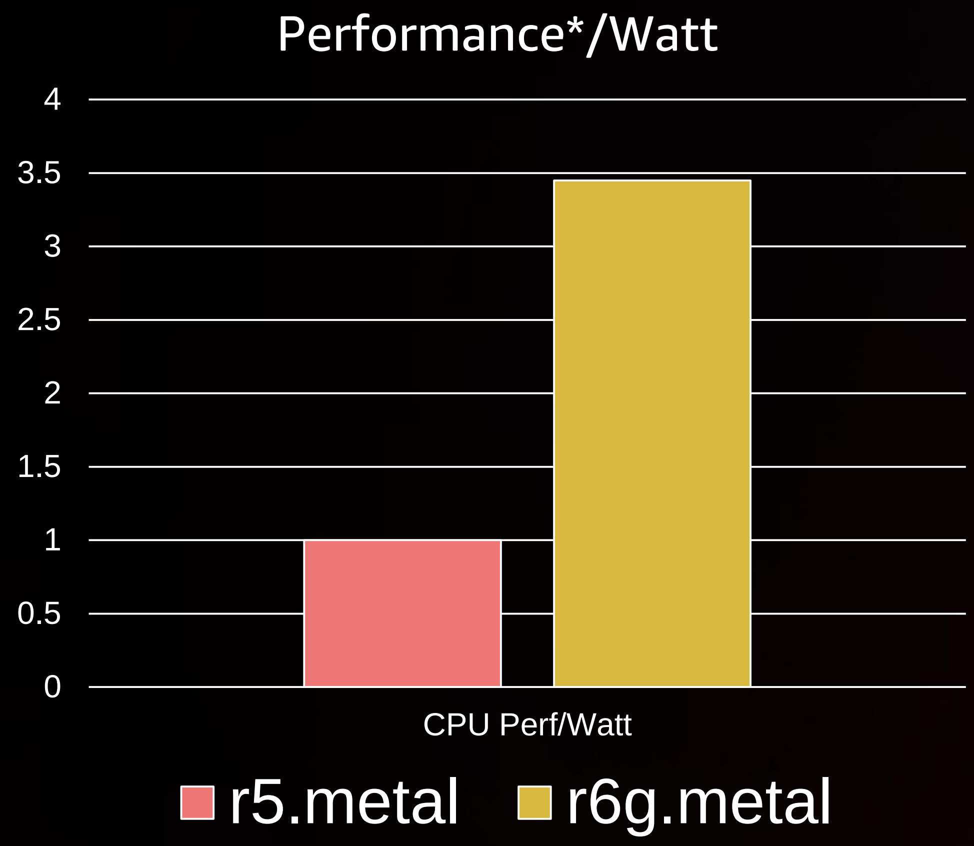
Graviton3 has up to 25% better performance than Graviton2

- Up to 2x higher floating-point performance
- Up to 2x faster cryptographic workload performance
- Up to 3x better machine learning performance

First in the cloud to feature DDR5 memory

C7g instances provide the best price-performance for compute-intensive workloads in Amazon EC2

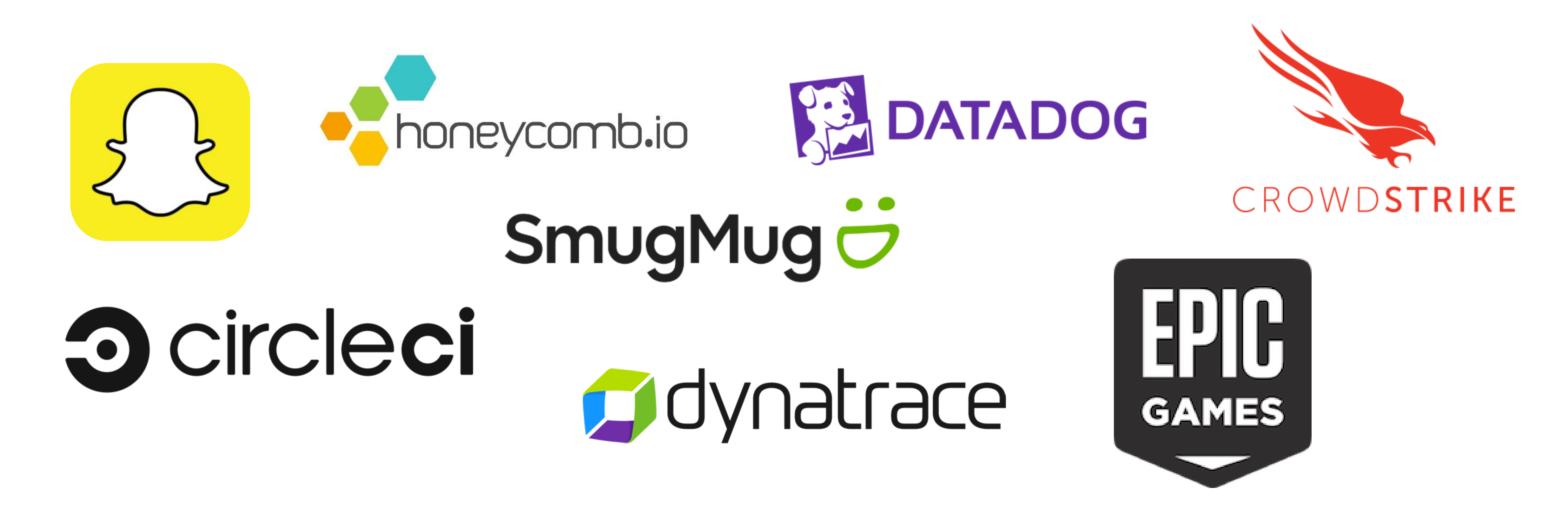
Sustainability: AWS Graviton Processors



*Estimated SPECint2017

- AWS Graviton2: processor power efficiency up to 3.5x better performance/watt*
  - Lower power
  - Higher density
  - Lower costs
  - Lower carbon footprint
- AWS Graviton3: 60% more energy efficient over comparable EC2 instances
- New – Sustainability Pillar for AWS Well-Architected Framework and Customer Carbon Footprint Tool (CFFT)

Graviton AWS Partners and customers



https://aws.amazon.com/ec2/graviton
https://aws.amazon.com/ec2/graviton/partners/

Serverless .NET on Graviton

AWS Lambda on Graviton

- Managed runtime for .NET 6
- Supports other versions via custom runtime or container image
- 19% better performance at 20% lower cost
- Deploy to Arm64 with AWS Toolkit for Visual Studio, or the Lambda tools for .NET CLI

AWS Fargate on Graviton

- Any .NET version that runs on Arm64
- Orchestrate with Amazon ECS or EKS
- Amazon ECR support for multi-architecture images

Graviton resources compared to x64 resources

	AWS Graviton3	Comparable x86-64
vCPU	Physical core	Logical core (hyperthread)
Sample EC2 costs (on demand)	c7g.xlarge: 4 vCPU, 8 GiB $0.1445 / hour	c6i.xlarge: 4 vCPU, 8 GiB $0.1700 / hour
Sample Amazon Aurora DB costs	db.r6g.4xlarge * $2.076 / hour	db.r6i.4xlarge $ 2.320/ hour
Sample Fargate costs	$0.03238 per vCPU / hour $0.00356 per GB / hour	$0.040480 per vCPU / hour $0.004445 per GB / hour

Costs for Ohio Region as of October 2022

.NET on AWS Graviton processors

.NET versions that can run on Graviton

.NET Core 2.1	runtimes only, no SDK (obsolete)
.NET Core 3.1	runtimes, SDK (EOS December 2022)
.NET 5.0	runtimes, SDK (Arm64 optimizations, out of support)
➡ .NET 6.0	runtimes, SDK (macOS Arm*, Windows on Arm*)
➡ .NET 7.0	runtimes, SDK (OSR, h/w vectorization, more)

Remember

- ✓ .NET Core 3.1 and .NET 6.0 are LTS (3 years support by MSFT)
- ✓ .NET 7 is a “current” releases (18 months support by MSFT)

Arm64 performance improvements in .NET 7

On-stack replacement (OSR)

OSR allows the runtime to optimize methods that are long-running, even currently executing (for both x64/ARM64)

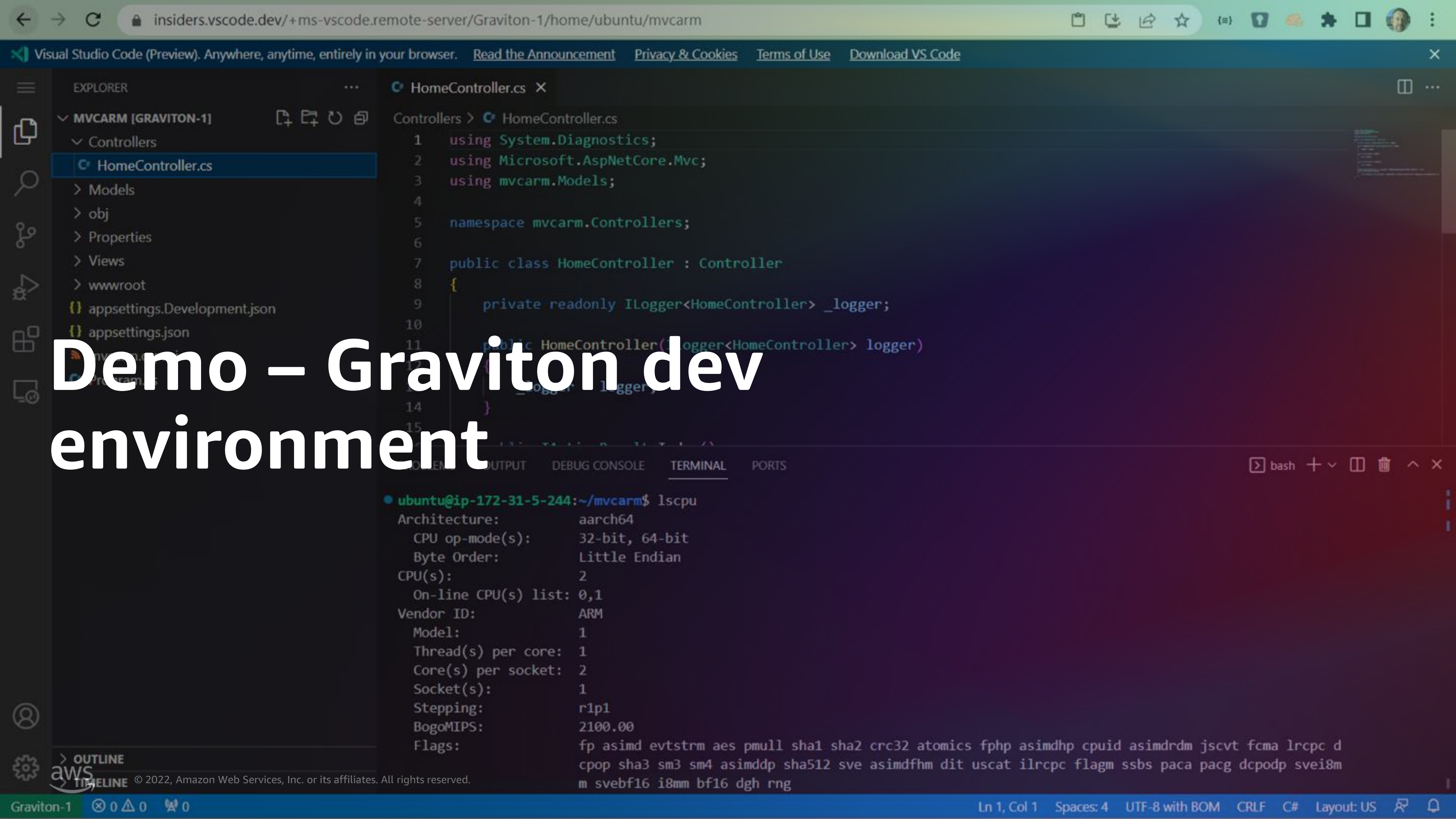
Hardware vectorization improvements

Improved use of SIMD in CPUs and GPUs for faster vector operations

Thread pool scaling for high core count

Improved performance on high-core-count CPUs (>32 cores) by moving from single global queue that worker threads poll to additional queues; for >32 cores, one new concurrent queue per 16 cores

Demo – Graviton dev environment



```
using System.Diagnostics;
using Microsoft.AspNetCore.Mvc;
using mvcarm.Models;

namespace mvcarm.Controllers;

public class HomeController : Controller
{
    private readonly ILogger<HomeController> _logger;

    public HomeController(ILogger<HomeController> logger)
    {
        _logger = logger;
    }
```

```
ubuntu@ip-172-31-5-244:~/mvcarm$ lscpu
Architecture:            aarch64
  CPU op-mode(s):        32-bit, 64-bit
  Byte Order:            Little Endian
CPU(s):                  2
  On-line CPU(s) list:   0,1
Vendor ID:               ARM
  Model:                 1
  Thread(s) per core:    1
  Core(s) per socket:    2
  Socket(s):             1
  Stepping:              r1p1
  BogoMIPS:              2100.00
  Flags:                 fp asimd evtstrm aes pmull sha1 sha2 crc32 atomics fphp asimdhp cpuid asimdrdm jscvt fcma lrcpc d
                         cpop sha3 sm3 sm4 asimddp sha512 sve asimdfhm dit uscat ilrcpc flagm ssbs paca pacg dcpodp svei8m
                         m svebf16 i8mm bf16 dgh rng
```

Broad Graviton support in DevOps Ecosystem

Fully managed

AWS CodeBuild

circleci

Cirrus CI

Travis CI

Hybrid (Hosted/bring-your-own-runner)

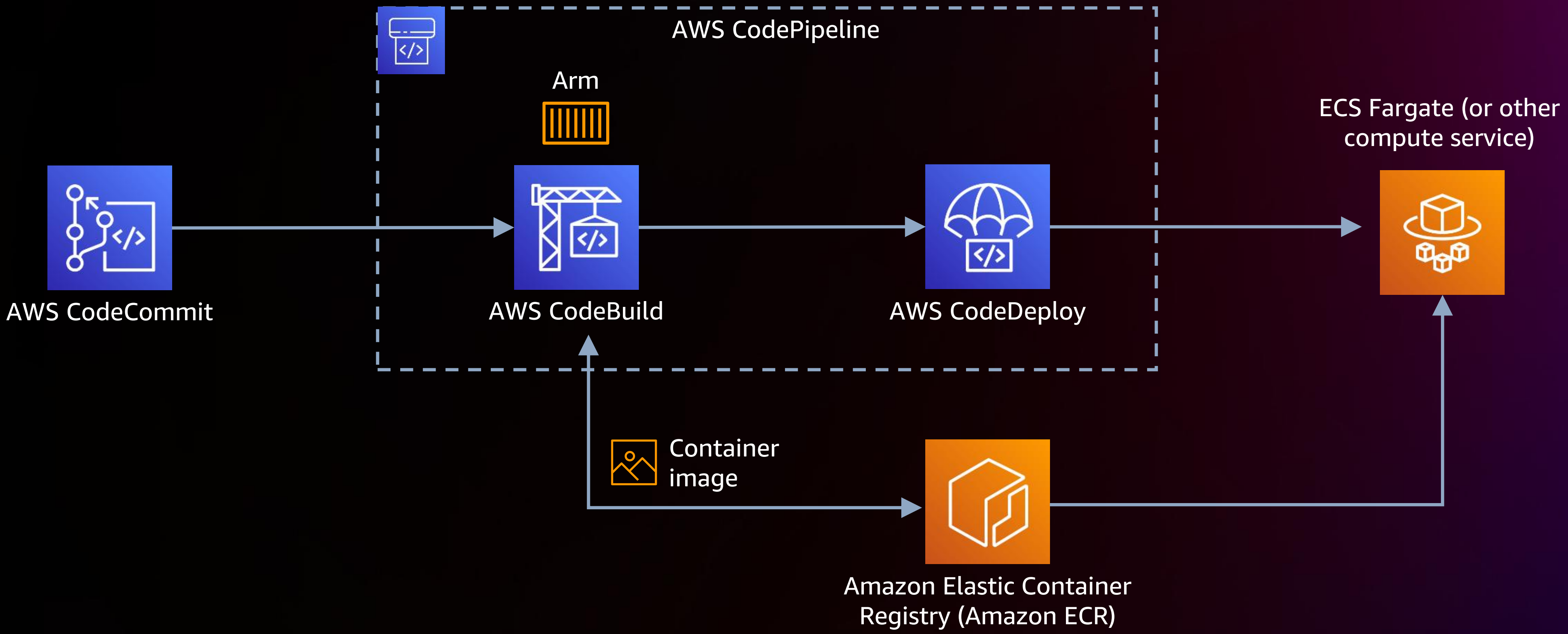
GitHub

GitLab

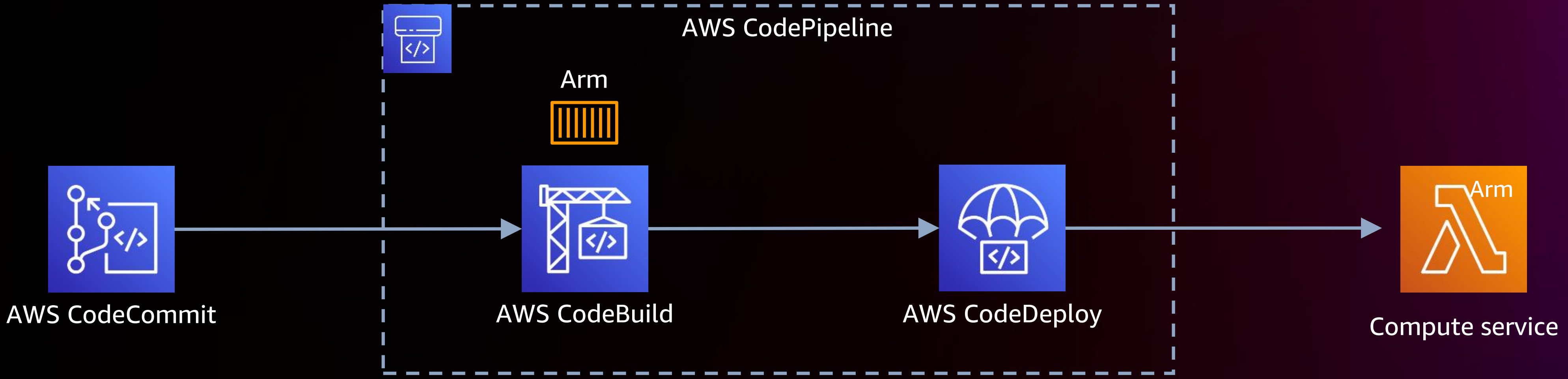
Self-managed

Jenkins

CI/CD pipeline on AWS – Using Arm64


AWS CodePipeline
Arm
AWS CodeCommit
AWS CodeBuild
AWS CodeDeploy
ECS Fargate (or other compute service)
Container image
Amazon Elastic Container Registry (Amazon ECR)

CI/CD pipeline on AWS – Using Arm64



Demo – CI/CD on Arm64

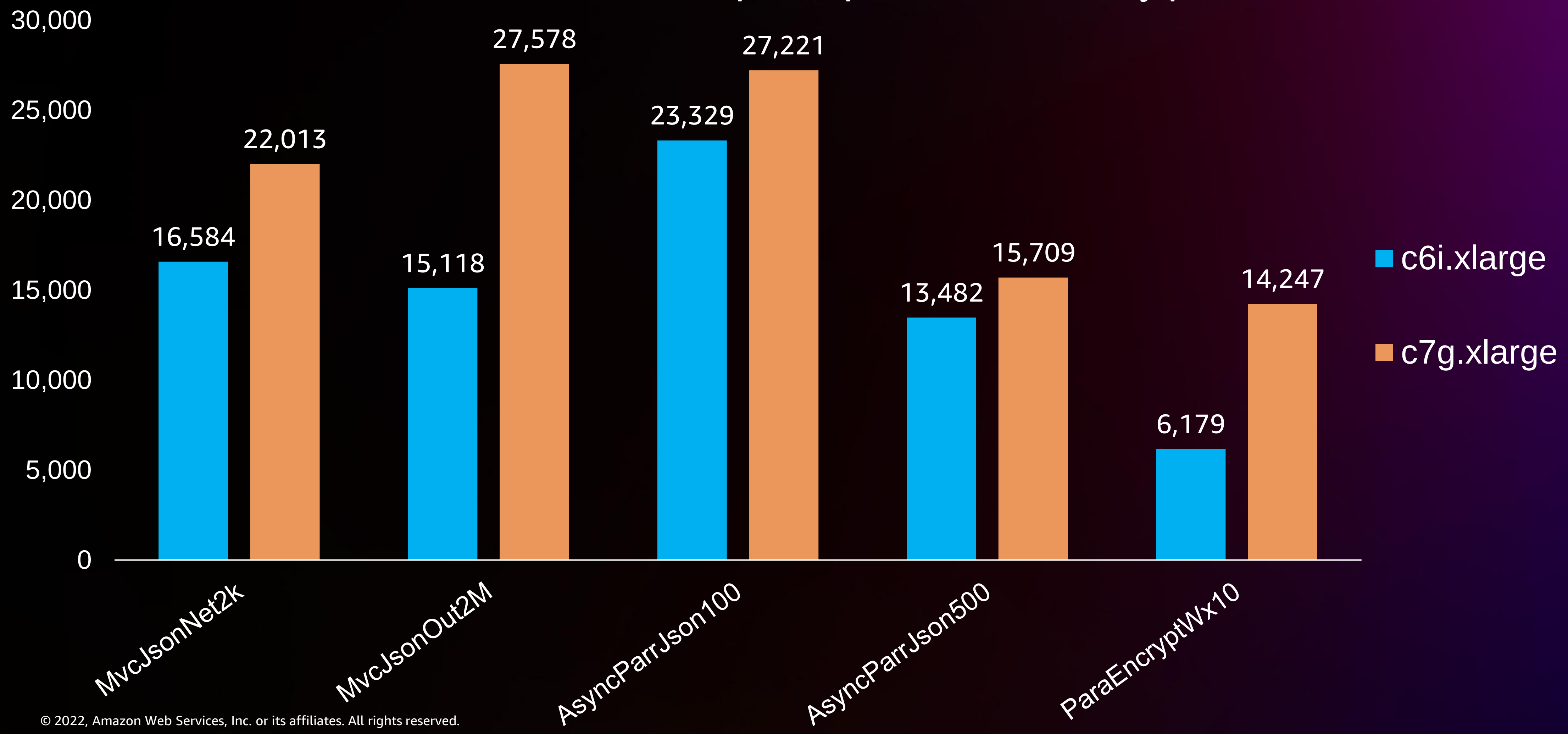
Performance & cost advantages

SIGNIFICANTLY BETTER PRICE/PERFORMANCE

Price-performance comparison: ASP.NET Core

ASP.NET Core MVC and Web API targeting .NET 7

Performance/dollar (requests per second/hourly price)



Demo – Benchmarking vs. x86-64 instance

AWS managed services supporting Graviton

EXTENDING THE GRAVITON2 PRICE PERFORMANCE TO MANAGED SERVICES

Databases

- Amazon DocumentDB
- Amazon Aurora
- Amazon RDS
- Amazon Elasticache
- Amazon MemoryDB
- Amazon Neptune

Analytics

- Amazon OpenSearch
- Amazon EMR

Compute

- AWS Lambda
- AWS Fargate
- AWS Elastic Beanstalk

Storage

- Amazon FSx for Lustre, Open ZFS

More Graviton + .NET resources

Workshop:
Running ASP.NET Core on EKS with Graviton2
bit.ly/3bVieOv

GitHub page:
AWS Graviton getting started – .NET
bit.ly/2Ywgd8z

Walkthrough:
Build & deploy Razor app to ECS with Graviton2 using AWS CDK
bit.ly/3bSdV6K

Code used in this session
bit.ly/3yMdxSU

@dotnetonaws

Thank you!

Kirk Davis
Principal Solutions Architect
@KirkAws

Sreelaxmi Pai
Principal Solutions Architect

Please complete the session survey