

AWS Graviton Migration

Service migration methodology and best practices

Efficient Compute Specialist SA
SoonBeom Kwon



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

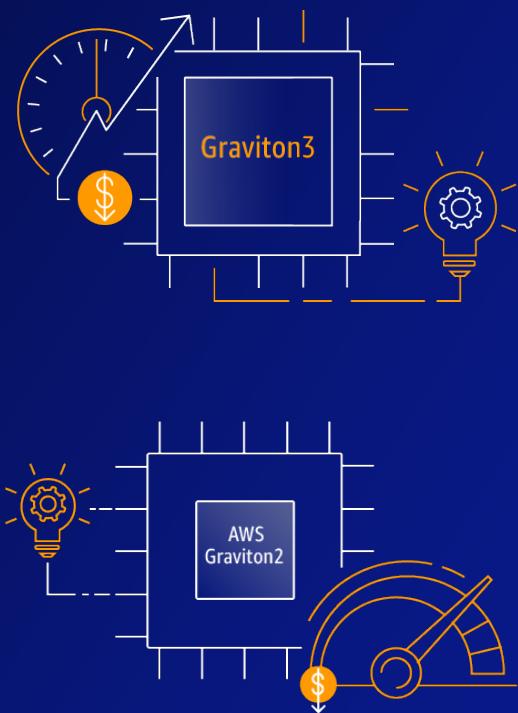
Agenda

- Steps toward Graviton
- Migrating EC2 Services
- Migrating Managed Services
- Building Graviton Container Images
- EKS / ECS for Graviton

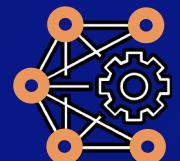


© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Graviton



Custom AWS silicon with 64-bit Arm processor cores



Targeted optimizations for **cloud-native workloads**



Best price/performance in Amazon EC2 for a broad array of workloads

AWS services supporting Graviton

EXTENDING THE GRAVITON PRICE PERFORMANCE TO MANAGED SERVICES

Developer tools



AWS CodeCommit



AWS CodeDeploy



AWS CodePipeline



AWS CodeBuild



Amazon Corretto

Compute & containers



AWS Lambda



AWS Elastic Beanstalk



AWS Fargate



Amazon ECS



Amazon ECR



Amazon EKS
Amazon EC2 Auto Scaling

Databases



Amazon Aurora



Amazon RDS



Amazon Neptune



Amazon MemoryDB
for Redis



Amazon DocumentDB
(with MongoDB compatibility)



Amazon ElastiCache

Analytics & AI/ML



Amazon OpenSearch Service



Amazon EMR



Amazon SageMaker

Other services



Amazon Inspector



AWS Systems Manager



Amazon CloudWatch



Amazon FSx
for Lustre



Amazon Linux



AWS App Mesh
Amazon GameLift

Learn more:

github.com/aws/aws-graviton-getting-started/blob/main/managed_services.md
aws.amazon.com/ec2/graviton



AWS Graviton Generations: A Timeline

2018: AWS Graviton1

- Introduced with EC2 A1 instance types
- 16 Cortex A72 cores with a clock speed of 2.3 GHz

2019: AWS Graviton2

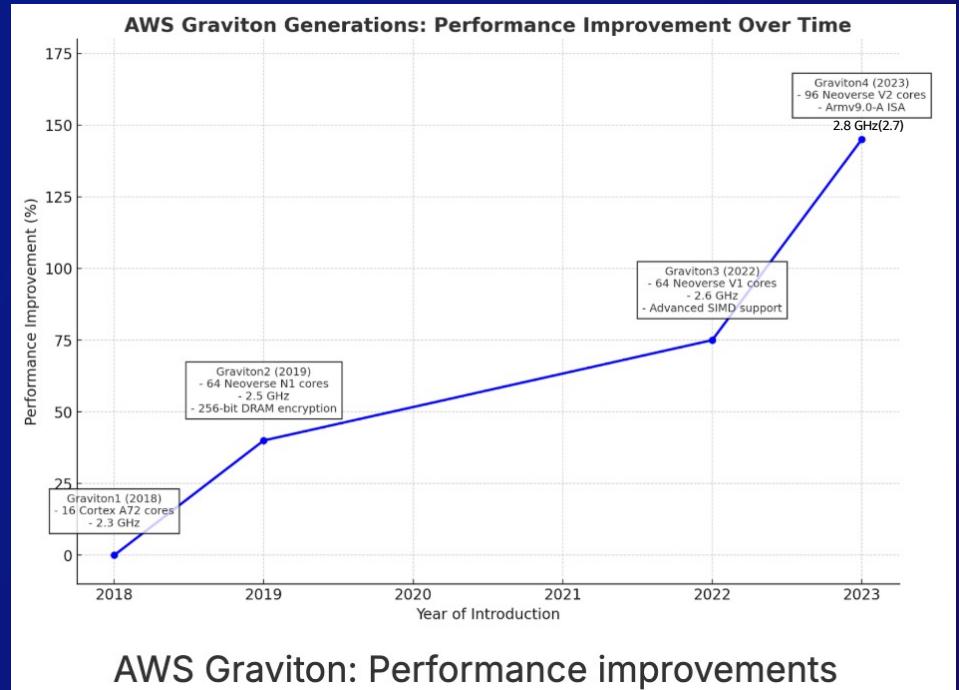
- Featured 64 Neoverse N1 cores clocking at 2.5 GHz
- Introduced always-on 256-bit DRAM encryption for improved security
- Available in a wide range of EC2 instances, M6g, C6g, R6g, and more

2022: AWS Graviton3

- Enhanced core count and performance with 64 Neoverse V1 cores at 2.6 GHz
- Support for advanced extensions like 4x128-bit Neon SIMD for high-performance computing
- Available in EC2 instances like C7g, M7g, and R7g

2023: AWS Graviton4

- Introduced 96 Neoverse V2 cores, based on Armv9.0-A ISA
- Dual Socket configuration with coherent NUMA



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

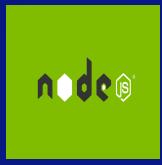
More recent software gives better performance



11+



5+



18+



1.18+



7.4+



11+



3.0+



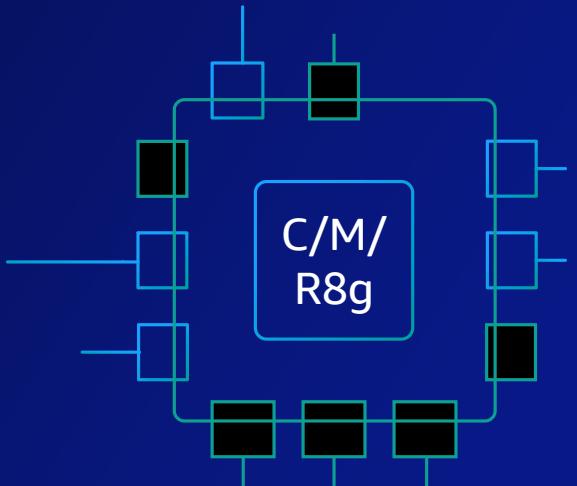
19.3+



1.0+

The minimum version supporting ARM architecture , each latest would be much better.

Graviton4 is Available in Seoul



Up to **30%** better performance compared to Graviton3-based instances

Up to **3x** more vCPUs and **3x** more memory than Graviton3-based instances

Up to 30% faster for web applications, 40% faster for databases, and 45% faster for large-scale Java applications compared to AWS Graviton3 processors



Steps toward Graviton



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Migration journey to Graviton

Plan and Validation



Prioritize target service
Compatibility check
including 3rd software

Target System Build and Test



EC2 arm AMI and Multi-
architecture image build
Test and optimize

Deployment and Monitoring



CI/CD pipeline update
Traffic routing
Monitoring and trouble
shooting

S1 - Languages and Runtimes Check

- 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

But external library dependency and runtime version check are required.



Porting Advisor

An open-source tool that **provides guidance** on getting applications Graviton ready

How? Analyzes application source code and generates recommendations of minimum required or recommended versions for a subset of language runtime and library dependencies
<https://github.com/aws/porting-advisor-for-graviton>

Currently supports

- Python 3+
- Java 8+
- Go 1.11+
- C, C++, Fortran

Project Information			
Project:	java-samples		
Source root:	/Users/waynetoh/Documents/Code/porting-advisor-for-graviton-1.0.0/sample-projects/java-samples		
Report Date:	2024-04-11 14:02:08		
Results			
File	Line #	Comments	
/Users/waynetoh/Documents/Code/porting-advisor-for-graviton-1.0.0/sample-projects/java-samples/pom.xml		3 files scanned. detected java code. we recommend using Corretto. see https://aws.amazon.com/corretto/ for more details.	
/Users/waynetoh/Documents/Code/porting-advisor-for-graviton-1.0.0/sample-projects/java-samples/pom.xml		dependency library: leveldbjni-all is not supported on Graviton	
/Users/waynetoh/Documents/Code/porting-advisor-for-graviton-1.0.0/sample-projects/java-samples/pom.xml		using dependency library snappy-java version 1.1.3. upgrade to at least version 1.1.4	
/Users/waynetoh/Documents/Code/porting-advisor-for-graviton-1.0.0/sample-projects/java-samples/pom.xml		using dependency library zstd-jni version 1.1.0. upgrade to at least version 1.2.0	
/Users/waynetoh/Documents/Code/porting-advisor-for-graviton-1.0.0/sample-projects/java-samples/pom.xml		using dependency library hadoop-lzo. this library requires a manual build more info at: https://github.com/aws/aws-graviton-getting-started/blob/main/java.md#building-multi-arch-jars	
		detected java code. min version 8 is required. version 11 or above is recommended. see https://github.com/aws/aws-graviton-getting-started/blob/main/java.md for more details.	



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon Q Developer Code Transformation

Complete Java language upgrades in a fraction of the time

Preview for .NET Framework to .NET Core

Internal Amazon results

1,000

production applications upgraded from Java 8 to Java 17 in just two days

10

minutes on average to upgrade each application

Less than 1

hour to complete the longest upgrade

2+

days previously taken to upgrade each application



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon Q Developer Code Transformation

Supported Java upgrades and IDEs

Amazon Q currently supports the following Java source code versions and target versions for transformations. Transforming code to the same Java version includes upgrading libraries and other dependencies in the source code version.

Supported Java upgrades	
Source code version	Supported target versions
Java 8	Java 17 and Java 21
Java 11	Java 17 and Java 21
Java 17	Java 17 and Java 21
Java 21	Java 21

Amazon Q supports Java upgrades in the following IDEs:

- Modules in JetBrains IDEs
- Projects and workspaces in Visual Studio Code

Step 1: Prerequisites

Before you continue, make sure you've completed the steps in [Set up Amazon Q in your IDE](#).

Make sure that the following prerequisites are met before you begin a Code Transformation job:

- Your project is written in a [supported Java version](#) and is built on Maven.
- Your project successfully builds with Maven in your IDE. Maven 3.8 or later is currently supported.

1. Converting Gradle Build File to Maven POM

<https://www.baeldung.com/gradle-build-to-maven-pom>

2. Upgrade Java Code with Q Code Transformation



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

S1 - Agents and Toolset Check

- 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
- Update your Infrastructure-As-Code resources such as Terraform 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 Graviton instance



K8S Cluster Components



Software Ecosystem Dashboard for Arm

For AI, Cloud, Data Center, 5G, Networking and Edge Software Packages

Legal disclaimers here

arm | Developer Develop Blogs Community CPU & Hardware Support Documentation

Filter by:

Search by package name 🔍

Active Filters: License: All Category: Cloud-native Reset Filters

Displaying 232 of 891 packages.

Package name	Supported since	Download
Agent	2016 March	
Amazon Elastic Kubernetes Service (EKS)	2020 August	
Anbox Cloud	2021 November	
Ansible	2017 August	
Apache Airflow	2022 May	
Apache Ant	2014 April	

<https://www.arm.com/developer-hub/ecosystem-dashboard/>



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

S2 - Test and Optimize

- Build multi-architecture image or EC2 arm AMI supporting specific requirements
 - such as security agent etc
- Run your test suite
 - run test suite to ensure all regular unit and function tests pass
 - establish a performance baseline on Graviton
 - compare performance with X86
- Optimize
 - upgrade runtime version and adjust parameters if applicable
 - upgrade external library version or patch
 - In case of single logic thread application, execute more applications per node



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

S3 – Deploy and Monitoring

- Perform Canary or Blue-Green deployment
 - start a Canary or Blue-Green deployment to re-direct a portion of application traffic to the Graviton-based instances
 - Canary is more appropriate than Blue-Green
- Monitoring and trouble shooting



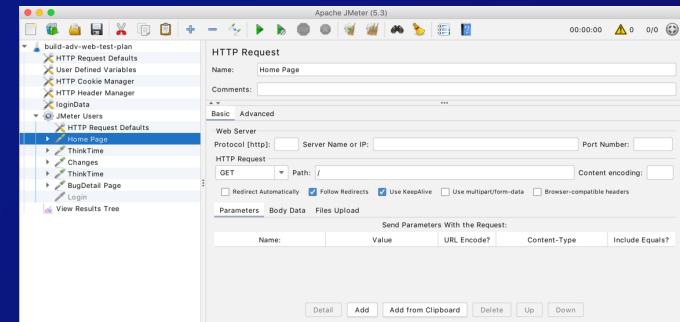
© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Performance Test Tools

```
Concurrency Level:      1
Time taken for tests:  10.696 seconds
Complete requests:     400
Failed requests:       0
Total transferred:    11120400 bytes
HTML transferred:     10968400 bytes
Requests per second:   37.40 [#/sec] (mean)
Time per request:     26.740 [ms] (mean)
Time per request:     26.740 [ms] (mean, across all concurrent requests)
Transfer rate:        1015.32 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median  max
Connect:        0    0.0  0.0    0    1
Processing:    24   26.0  0.9   26   37
Waiting:       24   26.0  0.9   26   35
Total:         24   27.1  1.0   27   38

Percentage of the requests served within a certain time (ms)
  50%   27
  66%   27
  75%   27
  80%   27
  90%   27
  95%   28
  98%   29
  99%   30
 100%  38 (longest request)
```



APACHE
JMeter™



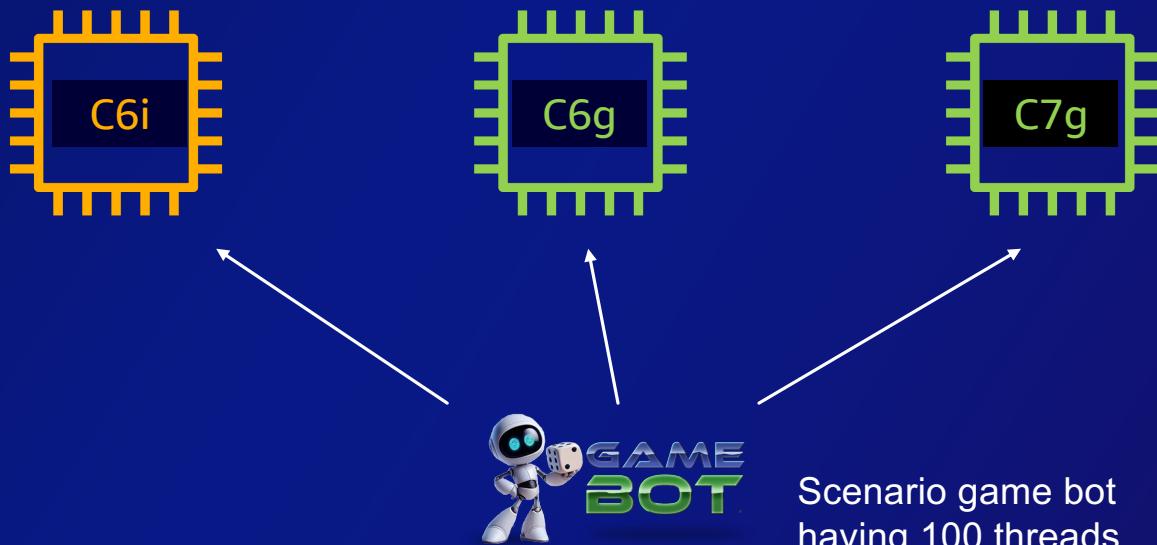
© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

And so on ...

Build performance test scenario

Design your own scenario considering user behaviors and service characteristics and implement with test tool

1 Gacha and 5 times battle with enemy and a boss might appear randomly. Go into a lobby in the middle. Do this scenario a total of 5 times and takes merely 40 minutes.



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Verify Performance

Compare performance metrics between graviton and X86

- Throughput Per Second (TPS) or Concurrent Users
- Latency such as P30, P50, P90 etc
- Error rate
- CPU utilization per cores and Load average
 - User call, system call, IO wait and idle
- Memory Usage
- Some internal metrics of runtime if it is needed (optional)



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Internet Shopping test scenario

- 첫 페이지 로딩 속도 측정 (랜딩페이지)
- 신규 회원 가입 - 사용자 입력 폼 작성, 이메일 인증 요청, 가입 완료 처리
- 로그인 처리 - 사용자 로그인 시도 (ID/PW 입력), 세션 생성 및 쿠키 설정, 마이페이지 이동
- 상품 검색 속도
- 상품 카테고리 이동 간의 반응 속도
- 장바구니 상품 담기 - 상품 상세 페이지 진입, 장바구니 담기, 장바구니 조회
- 주문 - 상품 선택 후 결제 페이지 진입, 주소/결제 수단 입력, 결제 완료 및 주문 확인
- 이벤트 트래픽 시뮬레이션 - 상품 조회/검색/결제 등 혼합 행동



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Profiling – linux perf

```
Samples: 1K of event 'cycles', Event count (approx.): 52279641135
Children      Self  Command      Shared Object      Symbol
- 97.81%    0.00%  java          libpthread-2.27.so  [.] start_thread
  start_thread
  ThreadJavaMain
  JavaMain
  jni_CallStaticVoidMethod
  jni_invoke_static
  JavaCalls::call_helper
  StubRoutines (1)
- Interpreter
  - 97.79% void CryptoBench.execute()
    - 78.28% java.lang.String CryptoBench.crypt(java.lang.String)
      + 21.87% int sun.security.provider.DigestBase.engineDigest(byte[], int, int)
      5.64% java.lang.Object java.security.Provider$Service.newInstance(java.lang.Object)
      4.39% StubRoutines (2)
+ 97.81%    0.00%  java          libjli.so        [.] ThreadJavaMain
+ 97.81%    0.00%  java          libjli.so        [.] JavaMain
+ 97.81%    0.00%  java          libjvm.so       [.] jni_CallStaticVoidMethod
+ 97.81%    0.00%  java          libjvm.so       [.] jni_invoke_static
+ 97.81%    0.00%  java          libjvm.so       [.] JavaCalls::call_helper
+ 97.81%    0.00%  java          [JIT] tid 17442  [.] StubRoutines (1)
+ 97.81%    0.00%  java          [JIT] tid 17442  [.] Interpreter
+ 97.79%  18.80%  java          [JIT] tid 17442  [.] void CryptoBench.execute()
+ 78.43%  40.08%  java          [JIT] tid 17442  [.] java.lang.String CryptoBench.crypt(java.lang.String)
+ 22.36%  22.02%  java          [JIT] tid 17442  [.] StubRoutines (2)
+ 21.91%  4.12%   java          [JIT] tid 17442  [.] int sun.security.provider.DigestBase.engineDigest(byte[], int, int)
+ 7.50%   7.50%   java          [kernel.kallsyms] [k] rmi
+ 6.68%  4.81%   java          [JIT] tid 17442  [.] java.lang.Object java.security.Provider$Service.newInstance(java.lang.Object)
+ 1.59%  0.05%   java          libc-2.27.so   [.] __clock_gettime
+ 1.48%  1.41%   java          [vdso]          [.] _vdso_clock_gettime
+ 1.22%  0.00%   java          [unknown]       [.] 0x000000002b37e39c
  0.29%  0.24%   java          [JIT] tid 17442  [.] java.security.Provider$Service sun.security.jca.ProviderList.getService(ja
```

<https://bell-sw.com/announcements/2022/04/07/how-to-use-perf-to-monitor-java-performance/>



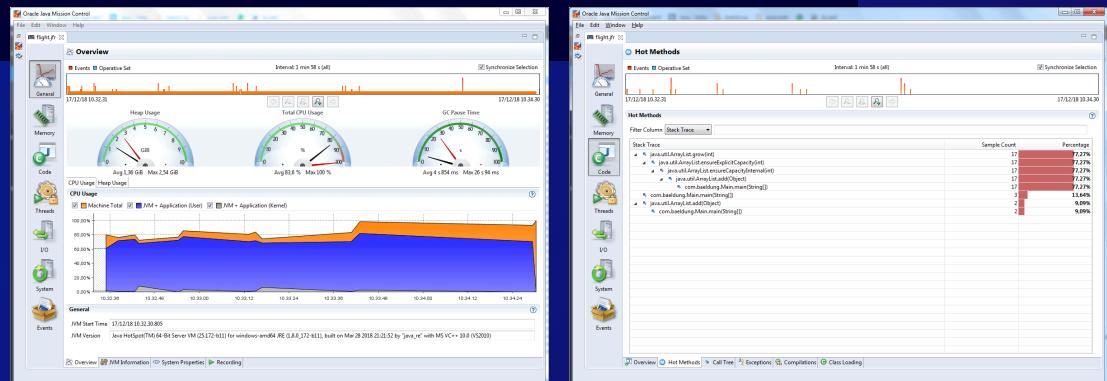
© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Profiling - JFR

Collects information about the events in a Java Virtual Machine (JVM) during the execution of a Java application. JFR is part of the JDK distribution, and it's integrated into the JVM. designed to affect the performance of a running application as little as possible. Java Mission Control (JMC), which is part of the JDK distribution, used to visualize the data collected by JFR.

```
kubectl exec -n graviton pod-name -- jcmt 1 JFR.start duration=60s  
filename=/tmp/recording.jfr settings=/usr/lib/jvm/java/lib/jfr/profile.jfc
```

```
kubectl cp or aws s3 cp to retrieve dump file
```



<https://docs.oracle.com/javacomponents/jmc-5-5/jfr-runtime-guide/index.html>
<https://www.baeldung.com/java-flight-recorder-monitoring>



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

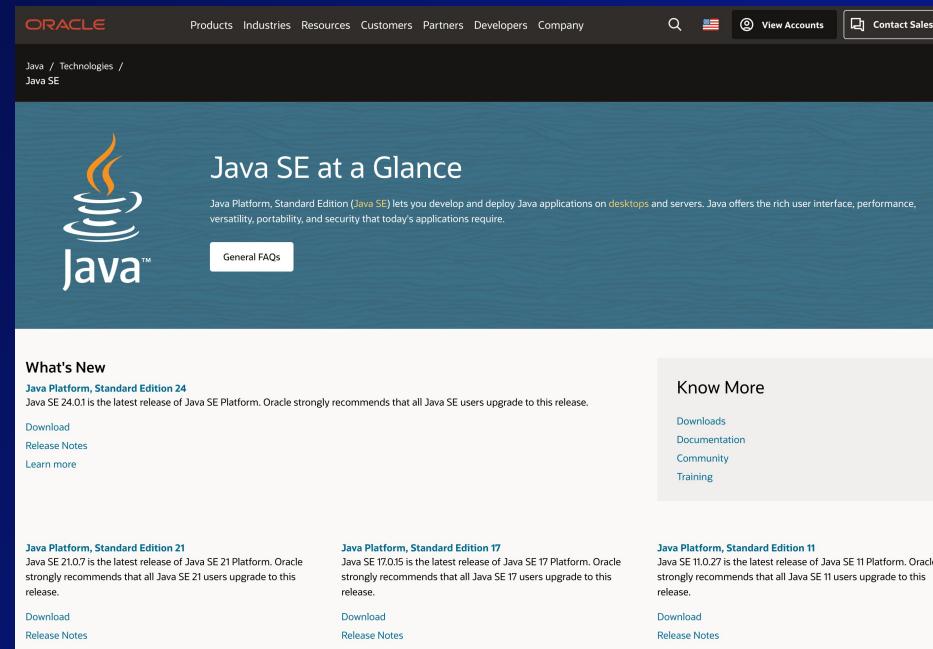
Monitoring

- Pod CPU, Memory, Logs, Crash or not
- Http Response Code
- Http Request latency (ingress prefix path metrics, URL)
- Graviton Node CPU, Memory Usage

And as much as you wish ...



Java Bug Fix database



The screenshot shows the Oracle Java SE at a Glance page. It features a large Java logo on the left and a "Java SE at a Glance" section with a brief overview of the platform's capabilities. Below this are sections for "What's New" (Java Platform, Standard Edition 24), "Know More" (Downloads, Documentation, Community, Training), and links for Java SE 21, 17, and 11.

Java SE at a Glance

Java Platform, Standard Edition (Java SE) lets you develop and deploy Java applications on desktops and servers. Java offers the rich user interface, performance, versatility, portability, and security that today's applications require.

General FAQs

What's New

Java Platform, Standard Edition 24
Java SE 24.0.1 is the latest release of Java SE Platform. Oracle strongly recommends that all Java SE users upgrade to this release.

[Download](#)
[Release Notes](#)
[Learn more](#)

Java Platform, Standard Edition 21
Java SE 21.0.7 is the latest release of Java SE 21 Platform. Oracle strongly recommends that all Java SE 21 users upgrade to this release.

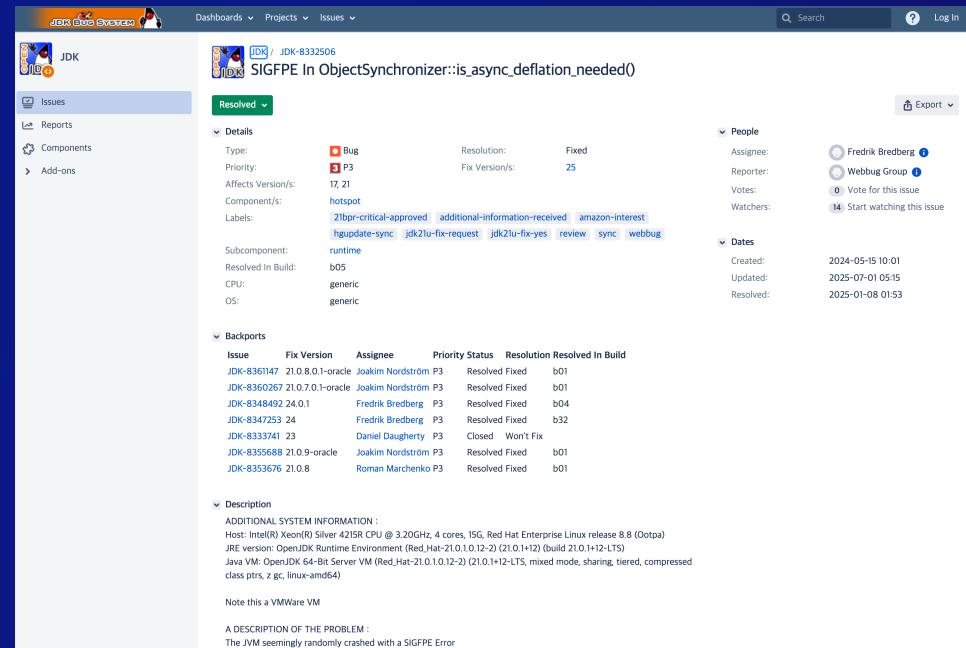
[Download](#)
[Release Notes](#)

Java Platform, Standard Edition 17
Java SE 17.0.15 is the latest release of Java SE 17 Platform. Oracle strongly recommends that all Java SE 17 users upgrade to this release.

[Download](#)
[Release Notes](#)

Java Platform, Standard Edition 11
Java SE 11.0.27 is the latest release of Java SE 11 Platform. Oracle strongly recommends that all Java SE 11 users upgrade to this release.

[Download](#)
[Release Notes](#)



The screenshot shows a Java Bug Fix database issue details page for bug ID JDK-8332506. The issue is titled "SIGFPE In ObjectSynchronizer::is_async_deflation_needed()". The "Resolved" tab is selected, showing the following details:

Details

Type:	Bug	Resolution:	Fixed
Priority:	P3	Fix Version/s:	25
Affects Version/s:	17, 21		
Component/s:	hotspot		
Labels:	2bpr-critical-approved additional-information-received amazon-interest hgupdat-sync jdk21u-fix-request jdk21u-fix-yes review sync webbug		
Subcomponent:	runtime		
Resolved In Build:	b05		
CPU:	generic		
OS:	generic		

People

Assignee:	Fredrik Bredberg
Reporter:	Webbug Group
Votes:	0
Watchers:	14

Dates

Created:	2024-05-15 10:01
Updated:	2025-07-01 05:15
Resolved:	2025-01-08 01:53

Backports

Issue	Fix Version	Assignee	Priority	Status	Resolution	Resolved In Build
JDK-836147	21.0.8.0.1-oracle	Joakim Nordström	P3	Resolved	Fixed	b01
JDK-8360267	21.0.7.0.1-oracle	Joakim Nordström	P3	Resolved	Fixed	b01
JDK-8348492	24.0.1	Fredrik Bredberg	P3	Resolved	Fixed	b04
JDK-8347253	24	Fredrik Bredberg	P3	Resolved	Fixed	b32
JDK-8333741	23	Daniel Daugherty	P3	Closed	Won't Fix	
JDK-8355688	21.0.9-oracle	Joakim Nordström	P3	Resolved	Fixed	b01
JDK-8353676	21.0.8	Roman Marchenko	P3	Resolved	Fixed	b01

Description

ADDITIONAL SYSTEM INFORMATION :
Host: Intel(R) Xeon(R) Silver 4215R CPU @ 3.20GHz, 4 cores, 15G, Red Hat Enterprise Linux release 8.8 (Ootpa)
JRE version: OpenJDK Runtime Environment (Red Hat 21.0.10.12-2) (21.0.1+12-LTS)
Java VM: OpenJDK 64-Bit Server VM (Red Hat 21.0.10.12-2) (21.0.1+12-LTS, mixed mode, sharing, tiered, compressed class ptrs, z gc, linux-amd64)

Note this a VMWare VM

A DESCRIPTION OF THE PROBLEM :
The JVM seemingly randomly crashed with a SIGFPE Error

<https://www.oracle.com/java/technologies/java-se-glance.html>



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Node.js Bug Fix database

The screenshot shows the GitHub Issues page for the nodejs/node repository. The URL is <https://github.com/nodejs/node/issues?q=state%3Aopen%20label%3Aarm>. The page is filtered by state:open and label:arm. There are 20 open issues listed:

- #54801: parallel.test-debugger-heap-profiler (arm, flaky-test) - opened on Sep 6, 2024
- #54347: meta: update devcontainer to the latest schema (arm, commit-queue-squash, meta) - opened on Aug 13, 2024, approved, 17/28
- #52963: test-pipe-file-to-http is flaky (timeout) (arm, flaky-test) - opened on May 13, 2024
- #51233: armv7l libc don't have version 'GLIBCXX_3.4.26' (arm, build) - opened on Dec 20, 2023
- #49933: sequential.test-watch-mode-inspect seems flaky (arm, flaky-test) - opened on Sep 29, 2023
- #47297: Problems with "wasm-webapi" tests and V8 >= 11.2 (arm, test, v8 engine, wasm) - opened on Mar 30, 2023
- #42643: arm64 cross-compile fails on amd64 (arm, build) - opened on Apr 8, 2022
- #41209: Move debug testing over to ARM from X86 (arm, test) - opened on Dec 17, 2021
- #41163: stack size exceeded on M1/arm64 but not on x86 (arm, v8 engine) - opened on Dec 14, 2021

<https://github.com/nodejs/node/issues?q=state%3Aopen%20label%3Aarm>



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

.Net Bug Fix database

The screenshot shows the GitHub Issues page for the `dotnet / sdk` repository. The page has a dark theme. At the top, there are navigation links for Code, Issues (2.7k), Pull requests (166), Discussions, Actions, Projects (1), Security (3), and Insights. A search bar is at the top right. A modal window titled "Want to contribute to dotnet/sdk?" is open, containing instructions to read the contributing guidelines before opening an issue and a link to first issues. Below the modal, there is a search bar with the query `is:issue state:open`, and buttons for Labels, Milestones, and New issue. A filter bar shows Open (2,723) and Closed (11,456) issues, along with dropdowns for Author, Labels, Projects, Milestones, Assignees, Types, and sorting by Newest. The main area lists ten open issues, each with a title, a link to the issue page, and the number of comments. The issues are:

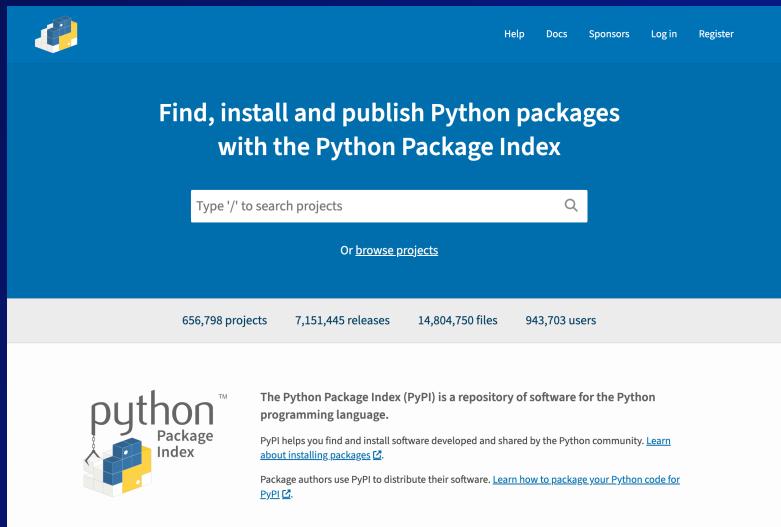
- Print clear informational message when `launchSettings.json` is used by `dotnet test` (`Area-dotnet test`)
- Improve output of `dotnet test` with MTP (`Area-dotnet test`)
- Better error message for missing `--solution`, `--project`, `--directory`, or `--test-modules` in `dotnet test` (`Area-dotnet test`)
- `dotnet test` for MTP handling of "global" command-line options (`Area-dotnet test`)
- Static web asset building is not incremental (`Area-AspNetCore`)
- `dotnet test` on 10.0.100-preview.5 not working with `--blame` (`Area-dotnet test`)
- Running `dotnet publish` fails on Blazor WebAssembly hosted project when using `PublishSingleFile=true` (`Area-WebSDK`)
- After installing 10.0 Preview 7 SDK on non-ENU OS, the installation completion page is not localized (`Area-Install`)
- Need to include common analyzers in our R2R generation (`Area-NetSDK`)

<https://github.com/dotnet/sdk/issues>



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Python on ARM (<https://pypi.org/>)



numpy 2.3.1

`pip install numpy`

Released: Jun 21, 2025

Fundamental package for array computing in Python

Navigation

- Project description
- Release history
- Download files**

Source Distribution

- [numpy-2.3.1.tar.gz](#) (20.4 MB [view details](#))
Uploaded Jun 21, 2025 [Source](#)

Built Distributions

- [numpy-2.3.1-py311-pypy311_pp73-win_amd64.whl](#) (12.9 MB [view details](#))
Uploaded Jun 21, 2025 PyPy Windows x86-64
- [numpy-2.3.1-py311-pypy311_pp73-manylinux_2_28_x86_64.whl](#) (16.8 MB [view details](#))
Uploaded Jun 21, 2025 PyPy/manylinux: glibc 2.28+ x86-64
- [numpy-2.3.1-py311-pypy311_pp73-manylinux_2_28_aarch64.whl](#) (14.4 MB [view details](#))**
Uploaded Jun 21, 2025 PyPy/manylinux: glibc 2.28+ aarch64

파이썬 패키지의 종류

1. Pure-Python Package

순수 Python 코드로만 이루어진 패키지 - Django, Flask 등이 해당

2. Python Package with Extension Module

Python 확장 모듈을 포함하는 패키지 - Numpy 등이 해당

배포 형태는

1. Source Distribution - 소스 코드 (일반적으로 .tar.gz 형태)

2. Built Distribution - 빌드 과정 없이 바로 설치 (.whl)

소스 코드만 존재하는 경우 ARM 용으로 빌드 필요 / 하지만 의존성 문제로 ARM 을 지원하지 않는 경우가 있음.



Recent Graviton Migration in APJ

Lottifiles - Complete Migration

- Application Migrated: Raspor to Lottie (core image-to-animation converter)
- Stack: TypeScript/Python, GitHub Actions, EKS with Karpenter
- Achievement: Successfully migrated Python application to Graviton with updated CI/CD pipeline using buildx
- Performance Results: Graviton demonstrated superior performance with lower cost during throttling tests
- Current Graviton Adoption: RDS PostgreSQL (80%), ElastiCache (100%)
- Challenge: GitHub Actions lacks ARM-based runners; buildx on x86 takes 4x longer than normal builds

Maybank - Architectural Progress

- Application Migrated: Strapi (NodeJS frontend/backend)
- Stack: GitHub Actions, ArgoCD, EKS Auto Mode
- Achievement: Identified clear path forward for Graviton adoption
- Next Steps: DevOps team will create Graviton cloud runner for better control over build process
- Challenge: Enterprise GitHub constraints limiting buildx actions approval

Revenue Group - Modernization Success

- Application Upgraded: Payment authentication system
- Achievement: Used Amazon Q Developer to upgrade from Java 8 to 17, then built ARM image using buildx
- Stack: Java/Spring Boot, some PHP, EC2/EKS migration in progress
- Technical Outcome: Successfully deployed upgraded Java 17 application on ARM-based EKS cluster
- Next Steps: Integrate manual deployment process into CodePipeline automation

Amway - Multi-Workload Migration

- Applications Migrated: Python Lambda function, Apache web server
- Achievement: Successfully implemented Lambda canary releases between x86 and ARM64 using aliases
- Stack: Apache web servers (multiple countries), PHP 8.0, Python 3.7 Lambdas
- Technical Outcome: Graviton web server successfully deployed; AMI build planned for ASG refresh
- Next Steps: Build Amazon Linux 2-based AMI for production ASG deployment

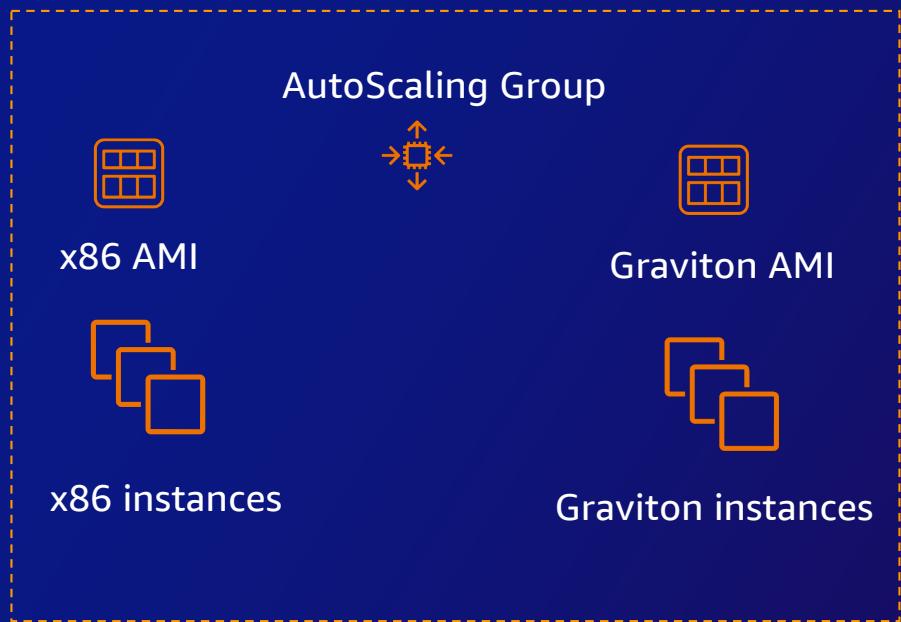
Migrating EC2 Services



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

EC2 AutoScaling Group Migration

1. Create a Graviton-based AMI
2. Create a new ASG launch template version
3. Update with new AMI id and Graviton instance type
4. Use an *instance refresh* to update the instances in your Auto Scaling group



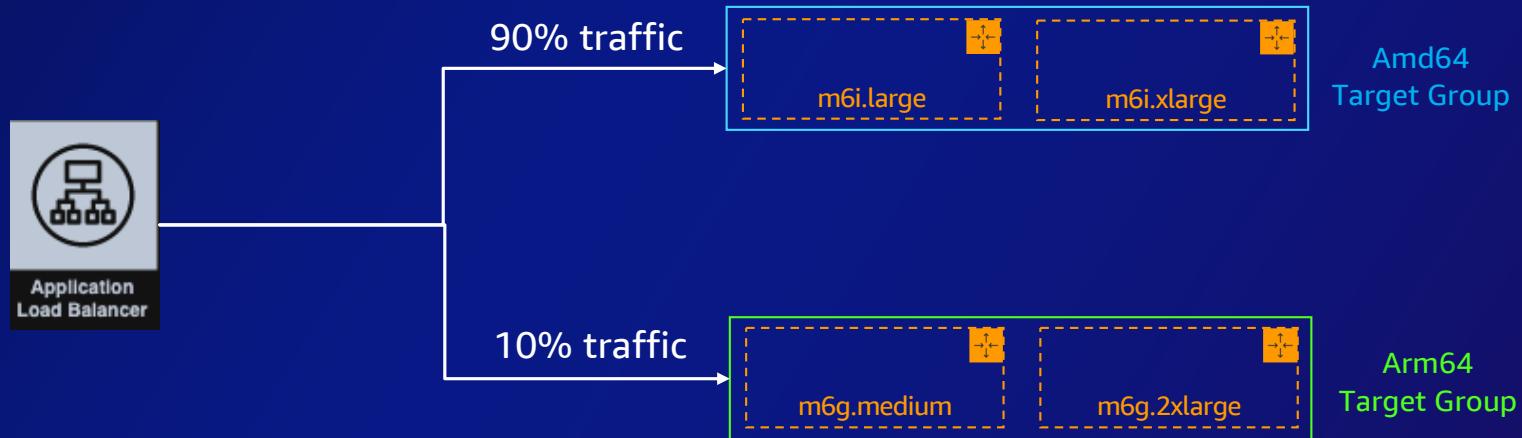
<https://aws.amazon.com/blogs/compute/supporting-aws-graviton2-and-x86-instance-types-in-the-same-auto-scaling-group/>
<https://docs.aws.amazon.com/autoscaling/ec2/userguide/ec2-auto-scaling-mixed-instances-groups-launch-template-overrides.html>



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Canary Releases

Introduce Graviton servers on a few instances and increase the footprint over time



<https://aws.amazon.com/blogs/aws/new-application-load-balancer-simplifies-deployment-with-weighted-target-groups/>



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Migrating Managed Services



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Considerations for Graviton RDS

1. Determine the minimum database version

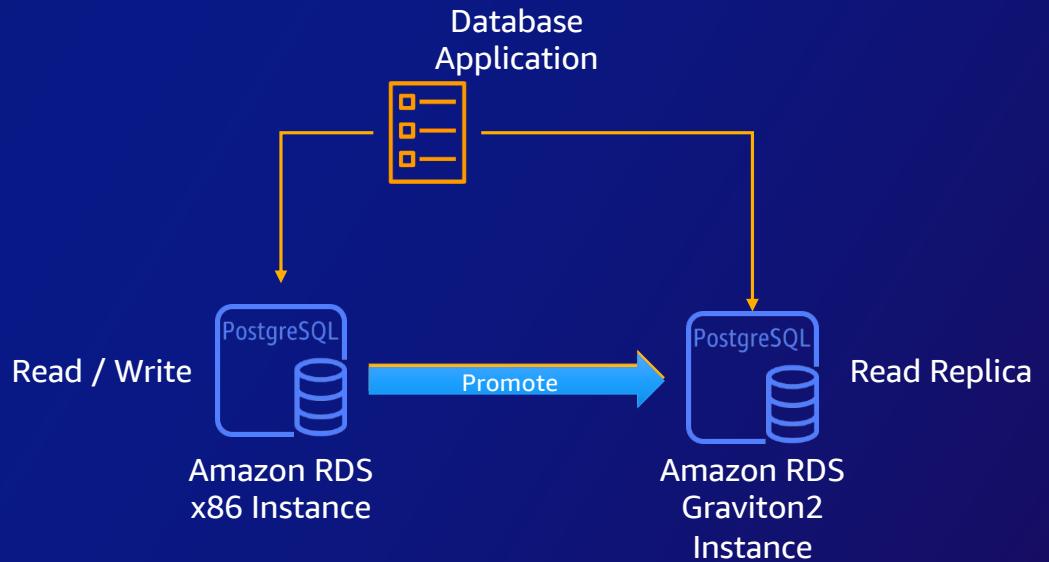
	MySQL	PostgreSQL	MariaDB
Amazon RDS	8.0.23 & higher	14, 13, 12.7 & higher	10.6, 10.5 and all 10.4
Amazon Aurora	2.09.2 and higher, 2.10.0 and higher, 3.01.0 and higher	14.3 and higher, 13.3 and higher, 12.8 and higher, 11.9, 11.12 and higher	—

1. In-place upgrade to minimum version
2. Modify instance class to Graviton with RDS with multi-AZ or Aurora replicas



Read Replicas for Graviton Migration

1. Create a Graviton2-based Read Replica
2. Quiesce the Application
3. Promote the Read Replica
4. Connect the Application to the Graviton2-based RDS Database
5. Remove the x86-based RDS instance



EMR on Graviton

- 20% lower cost and up to 19% improved performance for Spark/Hive workloads on Graviton2-based instances versus previous generation instances.
- Available for EMR Serverless
- C7g improves price-performance of running Spark workloads on Amazon EMR by 7.93–13.35% over C6g, depending on the instance size.

Instance Type	Minimum supported Amazon EMR version
m6g	emr-5.30.0, emr-6.1.0
m7g	emr-5.36.1, emr-6.7.0



EMR with mixed instance types

Considerations for using single versus multiple custom AMIs in an Amazon EMR cluster

Consideration	Single custom AMI	Multiple custom AMIs
Use both x86 and Graviton2 processors with custom AMIs in the same cluster	Not supported	Supported
AMI customization varies across instance types	Not supported	Supported
Change custom AMIs when adding new task instance groups/fleets to a running cluster. Note: you cannot change the custom AMI of existing instance groups/fleets.	Not supported	Supported
Use AWS Console to start a cluster	Supported	Not supported
Use AWS CloudFormation to start a cluster	Supported	Supported

When you use Amazon EMR 5.7.0 or higher, you can choose to specify a custom Amazon Linux AMI instead of the default Amazon Linux AMI for Amazon EMR.

Combining both Graviton-based and non-Graviton instances within the same cluster.

For example, a configuration like the following:

Primary Node: m6g.4xlarge
(Graviton)

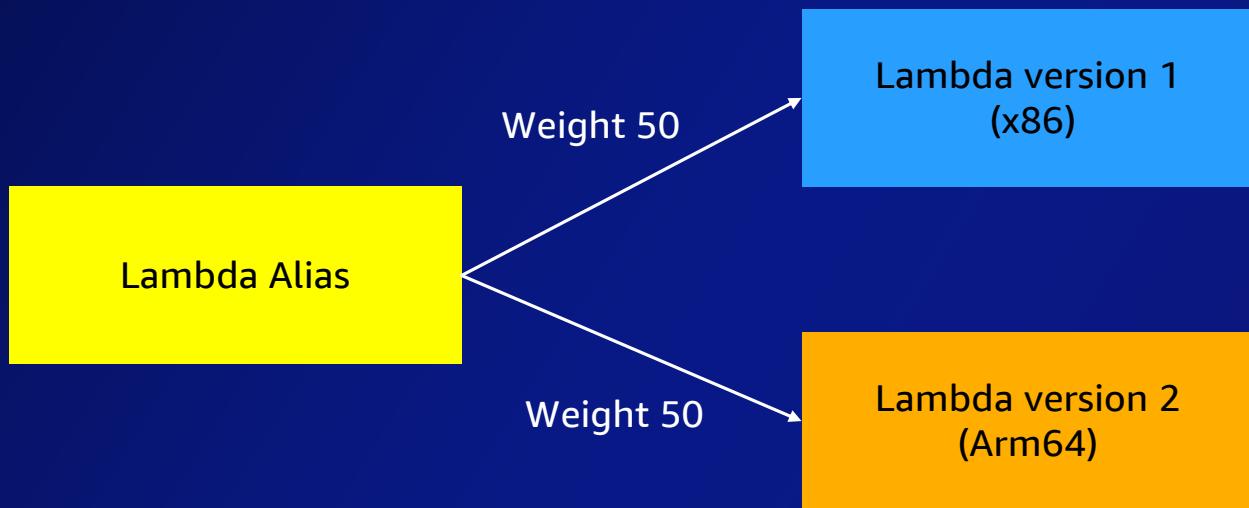
Core Nodes: r6g.2xlarge (Graviton)
Task Nodes (Spot Instances): r6a.xlarge (non-Graviton)

<https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-custom-ami.html>



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lambda



Edit alias

Alias configuration

Description - *optional*
production environment

Version	Weight (%)
2	50

Weighted alias

You can shift traffic between two versions, based on weights (%) that you assign. Click [here](#) to learn more.

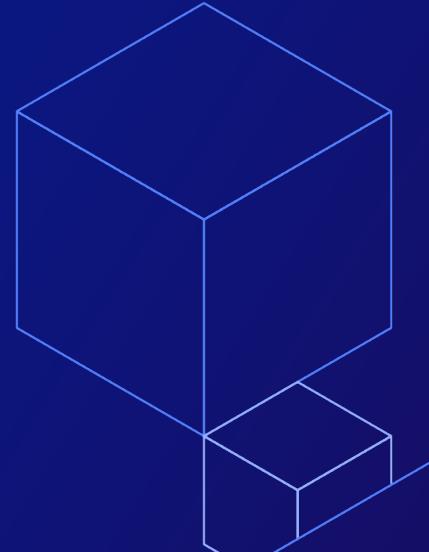
Additional version - <i>optional</i>	Weight (%)
1	50

Cancel **Save**



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Building Graviton Container Images



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Arm64 and container services

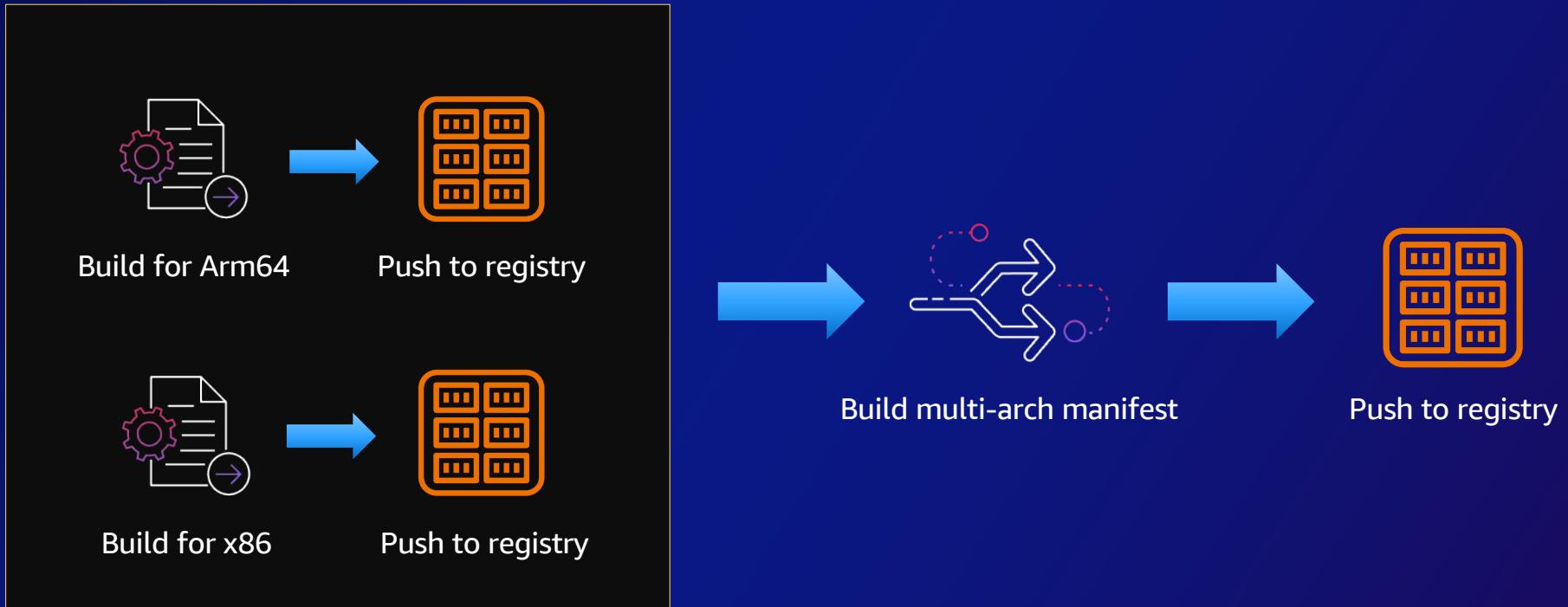
- **Amazon EKS, Amazon ECS and ROSA** support AWS Graviton instances, including mixed x86 and Arm clusters. Same release cadence as x86_64
- **Amazon ECR** supports multi-architecture manifest lists:
 - <https://aws.amazon.com/about-aws/whats-new/2020/05/ecr-now-supports-manifest-lists-for-multi-architecture-images/>



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Building multi architecture image

images that works on x86 and Graviton compute*



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

One step with BuildKit (QEMU)

Simplest way to produce multi-architecture images

```
$ docker buildx create --name mybuilder --use
```

Create a builder

```
$ docker buildx build \  
  --platform linux/arm64/v8,linux/amd64 \  
  -t 123456789012.dkr.ecr.us-west-2.amazonaws.com/myapp:test \  
  --push .
```



Build and push multi-arch image in a single step



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Building multi-arch images natively

Recommended for ci/cd pipelines

On arm64 build instance:

```
$ docker build \
-t 123456789012.dkr.ecr.us-west-2.amazonaws.com/myapp:test-arm64 .
$ docker push 123456789012.dkr.ecr.us-west-2.amazonaws.com/myapp:test-arm64
```



On x86 build instance:

```
$ docker build \
-t 123456789012.dkr.ecr.us-west-2.amazonaws.com/myapp:test-x86 .
$ docker push 123456789012.dkr.ecr.us-west-2.amazonaws.com/myapp:test-x86
```



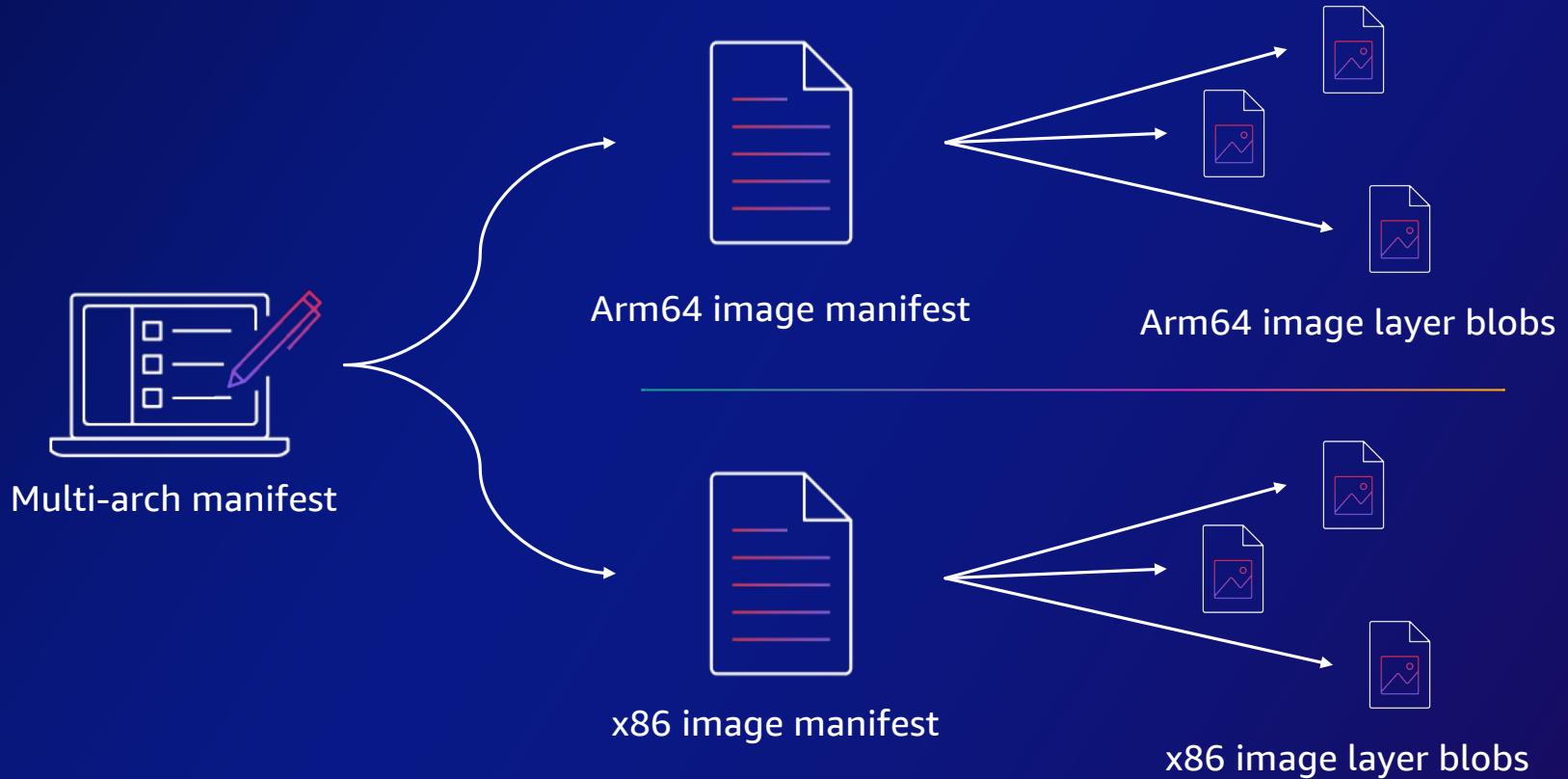
*On a build instance
(x86 or arm64):*

```
$ docker manifest create 123456789012.dkr.ecr.us-west-2.amazonaws.com/myapp:test \
123456789012.dkr.ecr.us-west-2.amazonaws.com/myapp:test-arm64 \
123456789012.dkr.ecr.us-west-2.amazonaws.com/myapp:test-x86
$ docker manifest push 123456789012.dkr.ecr.us-west-2.amazonaws.com/myapp:test
```



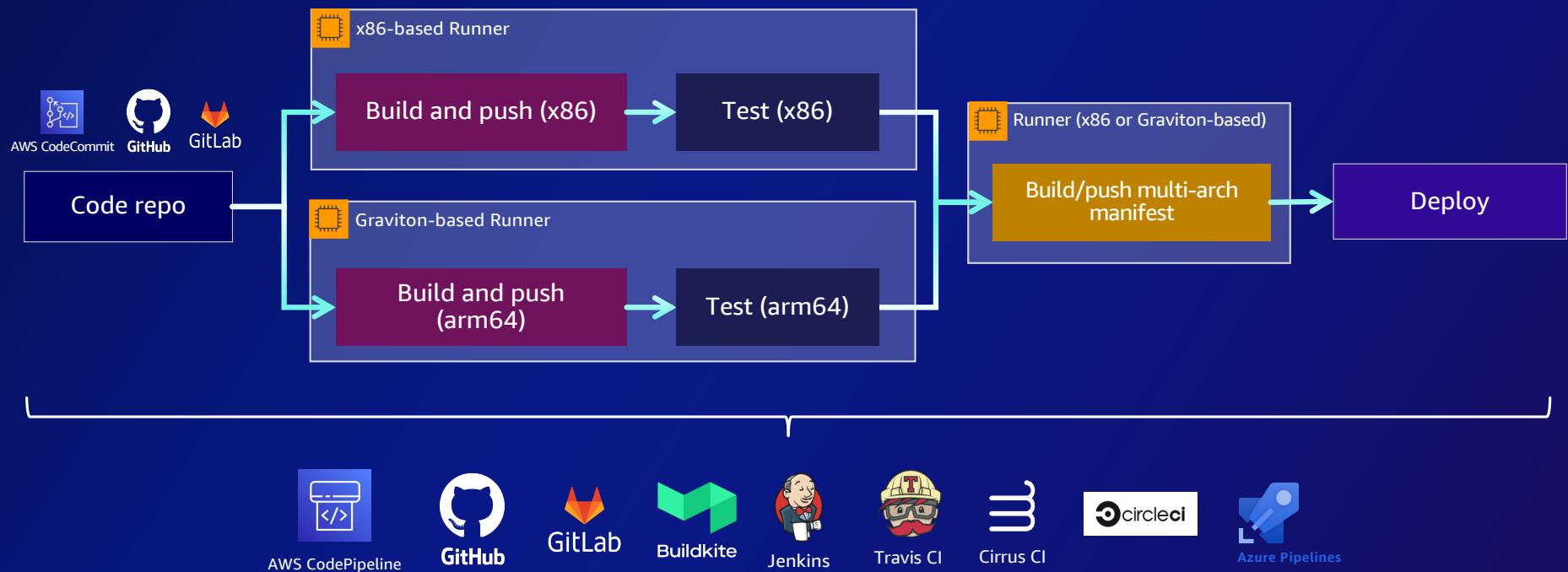
© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Products of the multi-arch build



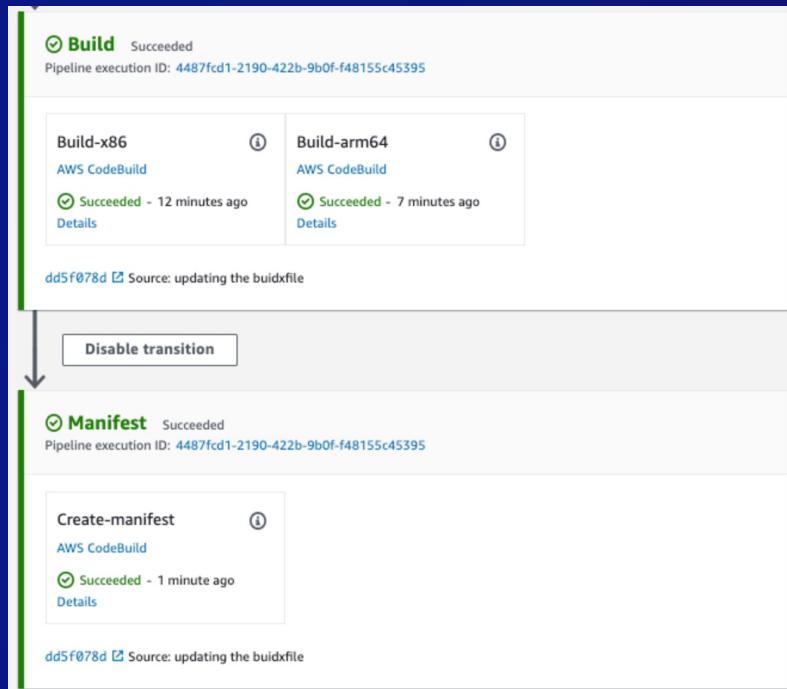
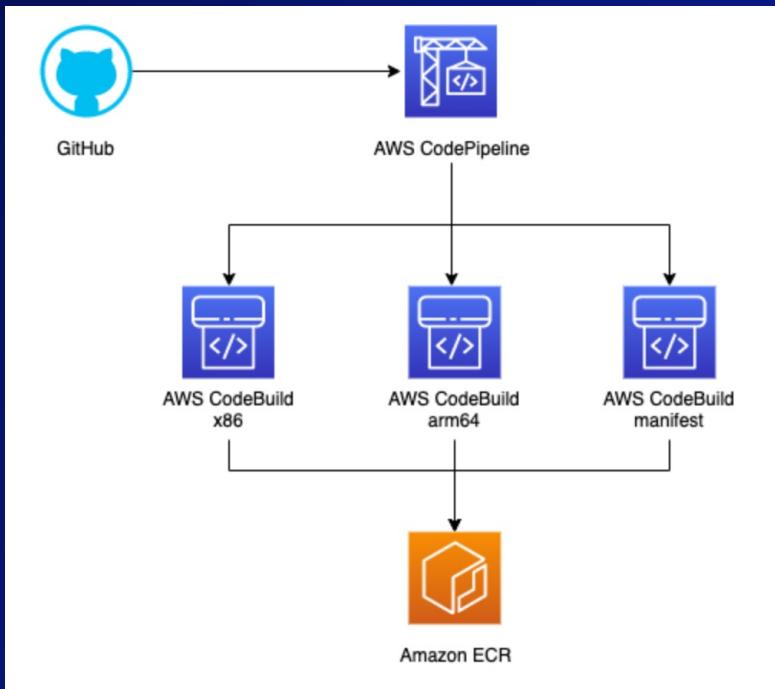
DevOps flow for flexible deployments

Pipeline for building Multi-Architecture Images and deployment



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Creating Multi-Architecture with CodePipeline



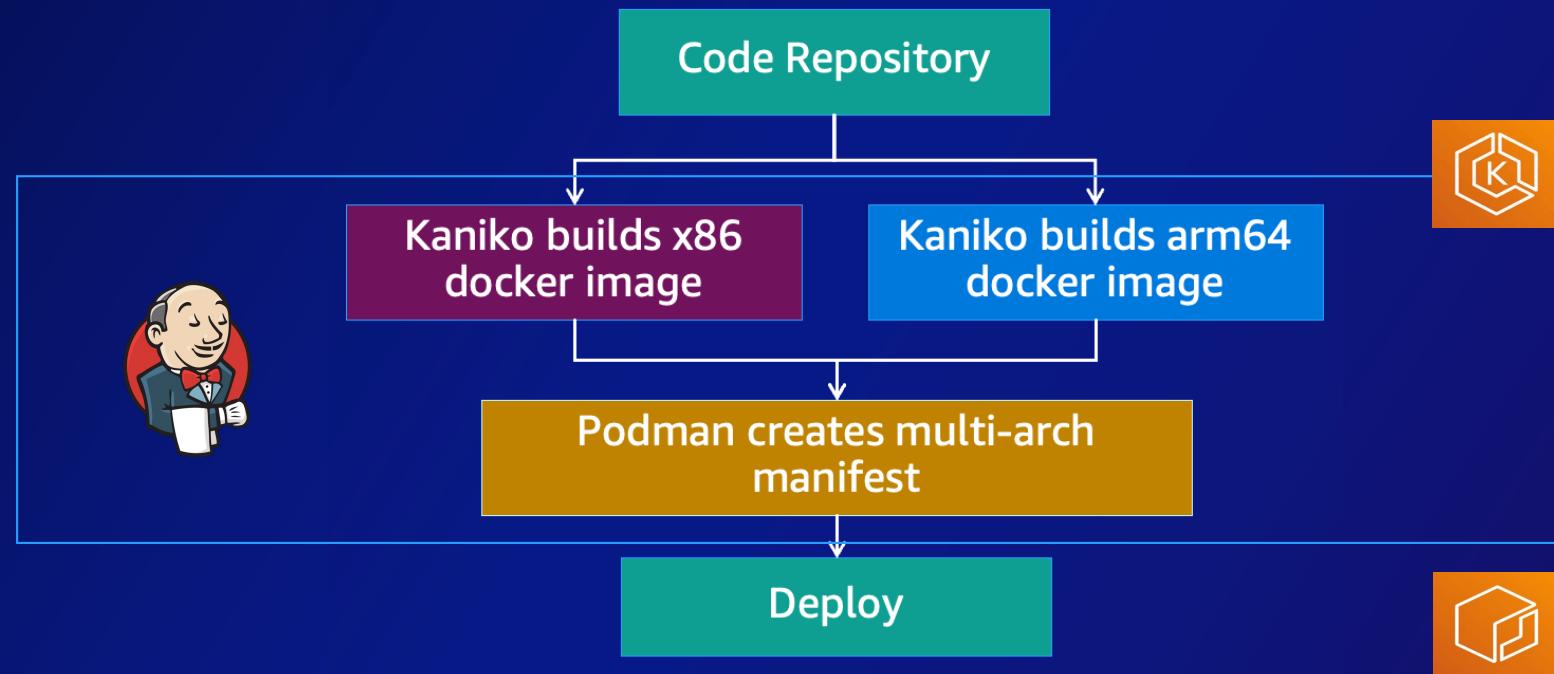
<https://aws.amazon.com/ko/blogs/devops/creating-multi-architecture-docker-images-to-support-graviton2-using-aws-codebuild-and-aws-codepipeline/>



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

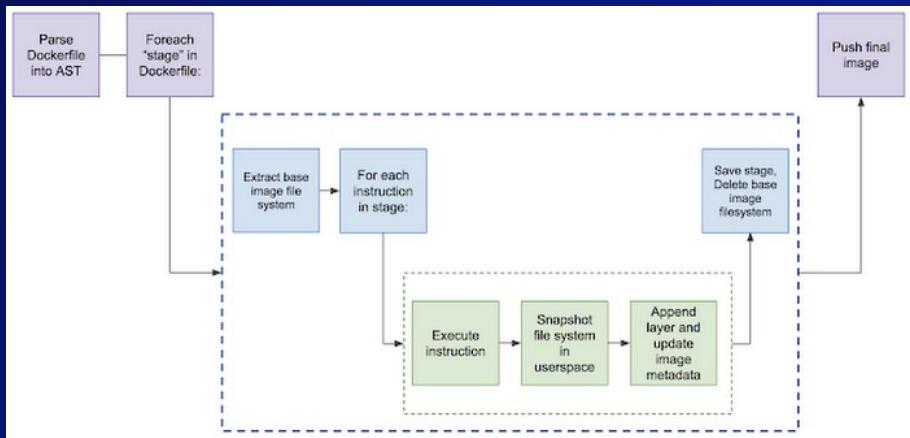
Creating Multi-Architecture with EKS POD

Jenkins parallel pipeline with kaniko and podman



Kaniko

kaniko is a tool to build container images from a Dockerfile, inside a container or Kubernetes cluster. And doesn't depend on a Docker daemon and executes each command within a Dockerfile completely in userspace. This enables building container images in environments that can't easily or securely run a Docker daemon, such as a standard Kubernetes cluster.



```
apiVersion: v1
kind: Pod
metadata:
  name: kaniko-git
spec:
  containers:
    - name: kaniko
      image: gcr.io/kaniko-project/executor:latest
      args: [ "--dockerfile=../Dockerfile",
               "--context=git://git@github.com:scoredev/scoredev.git#main",
               "--destination=docker://scoredev/kaniko:latest" ]
      volumeMounts:
        - name: kaniko-secret
          mountPath: /kaniko/.docker
      restartPolicy: Never
  volumes:
    - name: kaniko-secret
      secret:
        secretName: kaniko-secret
~/workspace/kaniko$ kubectl create -f pod.yaml
pod/kaniko created
~/workspace/kaniko$ kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
kaniko   0/1     ContainerCreating   0          5s
~/workspace/kaniko$ kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
kaniko   1/1     Running   0          13s
~/workspace/kaniko$ kubectl logs -f kaniko
Enabling experimental features...
Counting objects: 100%, done.
Compressing objects: 100% (3/3), done.
Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
INFO[0002] Retrieving image manifest ubuntu
INFO[0002] Retrieving image manifest ubuntu
INFO[0002] Returning cached image manifest
INFO[0005] Executing 0 build triggers
INFO[0005] Skipping unpacking as no commands require it.
INFO[0005] ENTRYPOINT ["bin/bash", "-c", "echo hello"]
INFO[0005] Pushing image to docker://scoredev/kaniko:ubuntu
INFO[0009] Pushed index.docker.io/scoredev/kaniko@sha256:bcd4f13f243f957d7794f3162c59b8e2aedcef103e5087898e8fe36233fa6fc9
```

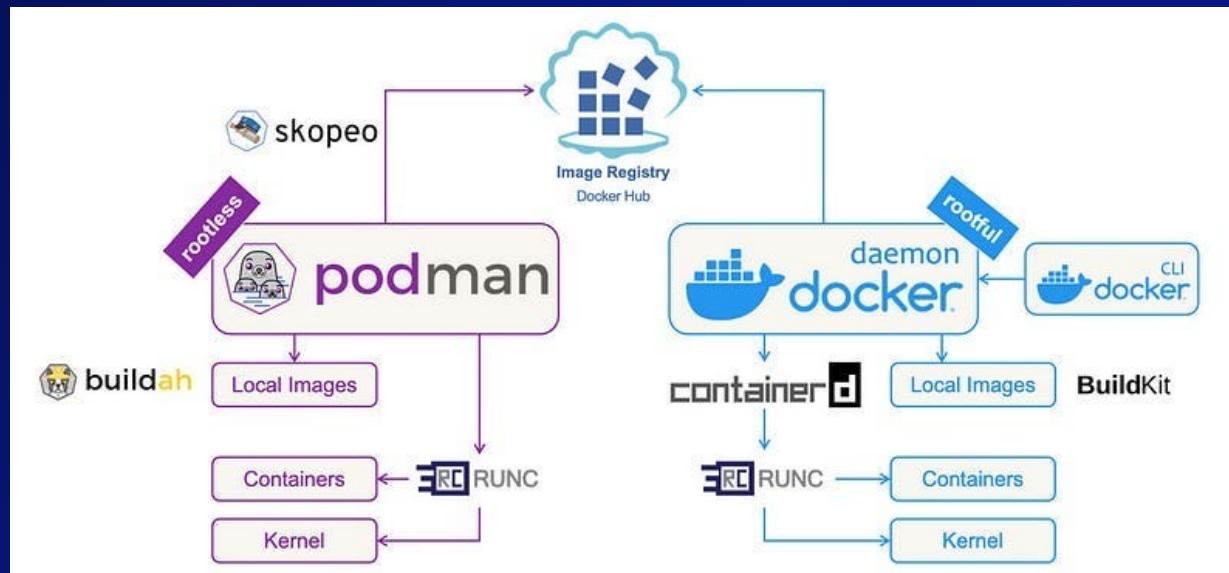
<https://github.com/GoogleContainerTools/kaniko>



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Podman

Podman is an open-source, daemonless container engine that allows users to develop, manage, and run containers and pods using the Open Container Initiative (OCI) standards. It's a popular alternative to Docker, particularly known for its daemonless architecture, which enhances security and efficiency. Podman can manage containers, images, pods, and volumes using a command-line interface (CLI) and the libpod library.



<https://podman.io/>

podman manifest create ~

podman manifest add ~

podman manifest inspect ~

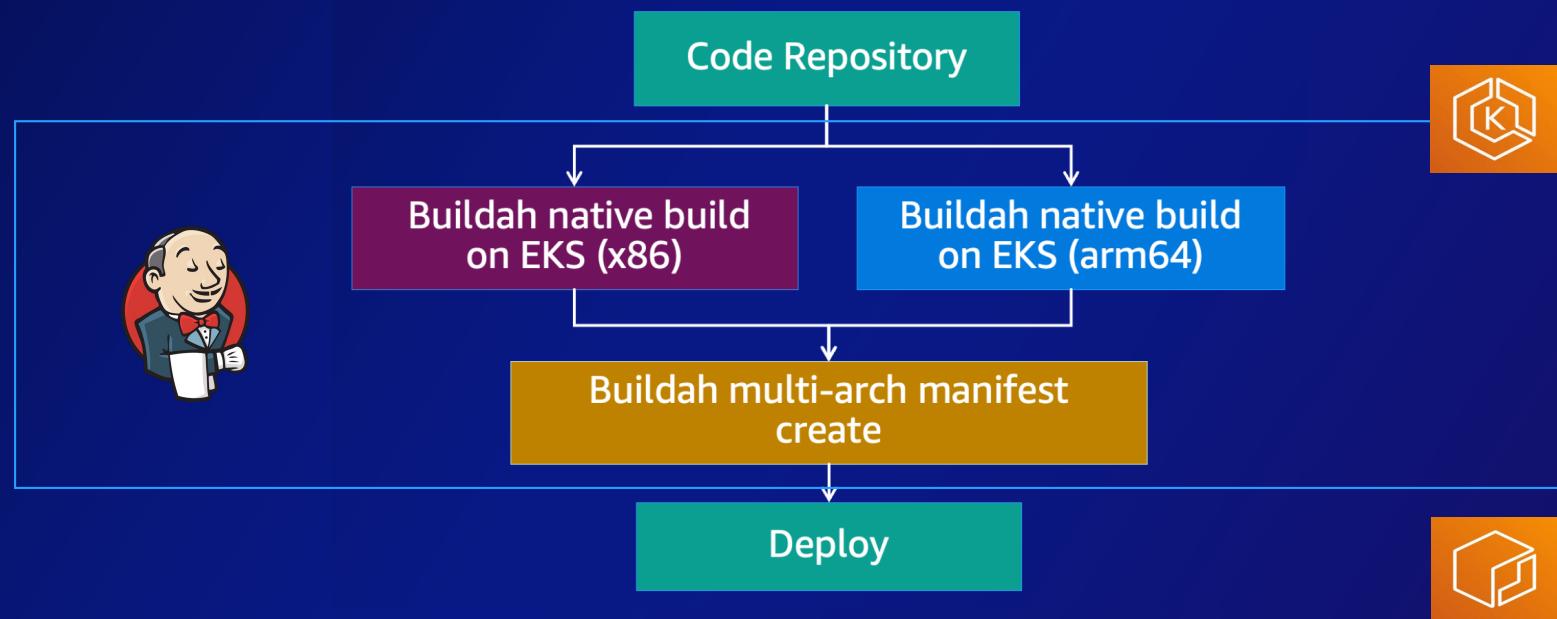
podman manifest push ~



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Creating Multi-Architecture with EKS POD

Jenkins parallel pipeline with buildah



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

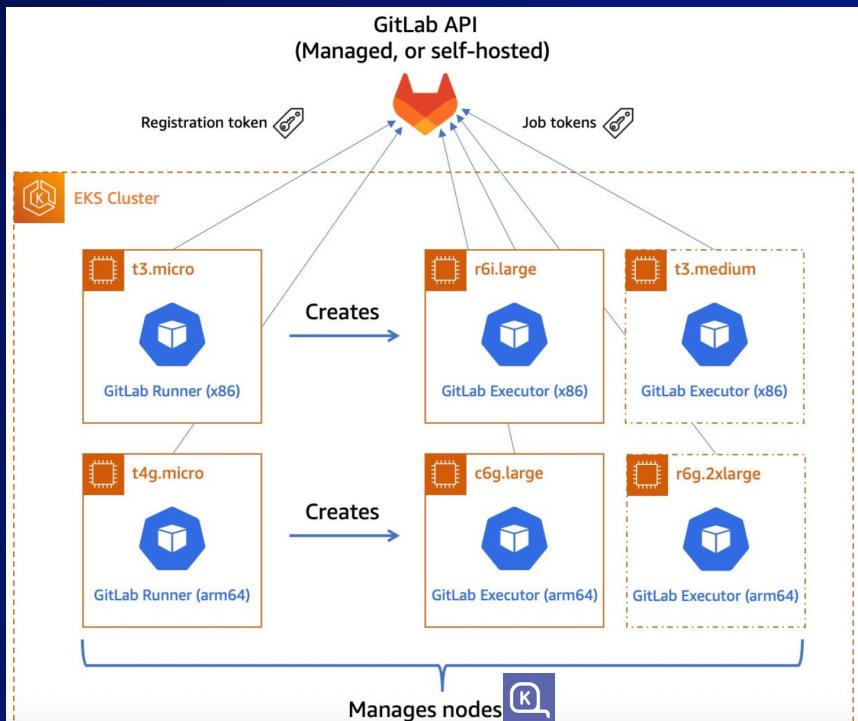
Buildah

Buildah is an open-source, Linux-based tool for building OCI-compliant container images. It's designed to be a lightweight, daemonless alternative to tools like Docker for image creation, offering fine-grained control over the process and integration with build pipelines. Essentially, it allows you to build, create, and manipulate container images without needing a container runtime like Docker or Podman.

- **Compatible with the [Open Container Initiative \(OCI\) Standard](#)**
- Daemonless Operation
- Dockerfile Support
- Image Layering
- Easily integrated into scripts and build pipelines



Creating Multi-Architecture with GitLab



A runner is an application to which GitLab sends jobs that are defined in a CI/CD pipeline. The runner receives jobs from GitLab and executes them—either by itself, or by passing it to an executor

In GitLab CI/CD, the executor's job is to perform the actual build. The runner can create hundreds or thousands of executors as needed to meet current demand, subject to the concurrency limits that you specify. Executors are created only when needed, and they are ephemeral: once a job has finished running on an executor, the runner will terminate it.

GitLab Executor use **Kaniko** and **Podman** to build docker image

<https://aws.amazon.com/blogs/devops/unlock-the-power-of-ec2-graviton-with-gitlab-ci-cd-and-eks-runners/>



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

EKS / ECS for Graviton



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Using Graviton with Amazon EKS

- **Amazon EKS fully supports Graviton EC2 instances**
- **You can add both Graviton and x86 EC2 instances to a cluster**
- **You can schedule pods on Graviton instances via nodeSelector or nodeAffinity**
- **Karpenter is the best and fastest way to provision Graviton instances for Amazon EKS clusters**
- **Cluster Auto Scaler can scale out Graviton Auto Scaling groups**



Breaking down Amazon EKS costs

What it takes to run a workload on Amazon EKS

Control plane

Amazon EKS cost
(fixed, \$0.1 per hour)

Workload (compute cost)

- Pods/containers
 - CPU
 - Memory
- Volumes

Node/
AWS Fargate

Network cost

- Pod data transfer
- Amazon ECR
- Elastic Load Balancing (ELB)
- NAT gateway and more

Observability cost

- Logging
- Monitoring

Average 80% of total cost



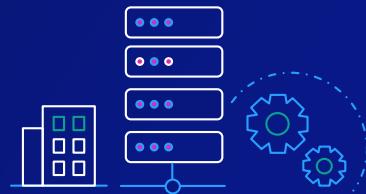
© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Source: CNCF, "FinOps for Kubernetes": bit.ly/3IR5Z6C

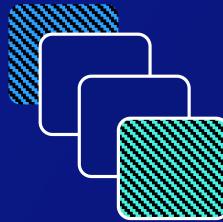
Cost optimization strategies



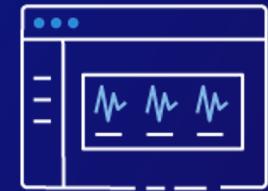
Rightsizing



Hardware



Purchase options

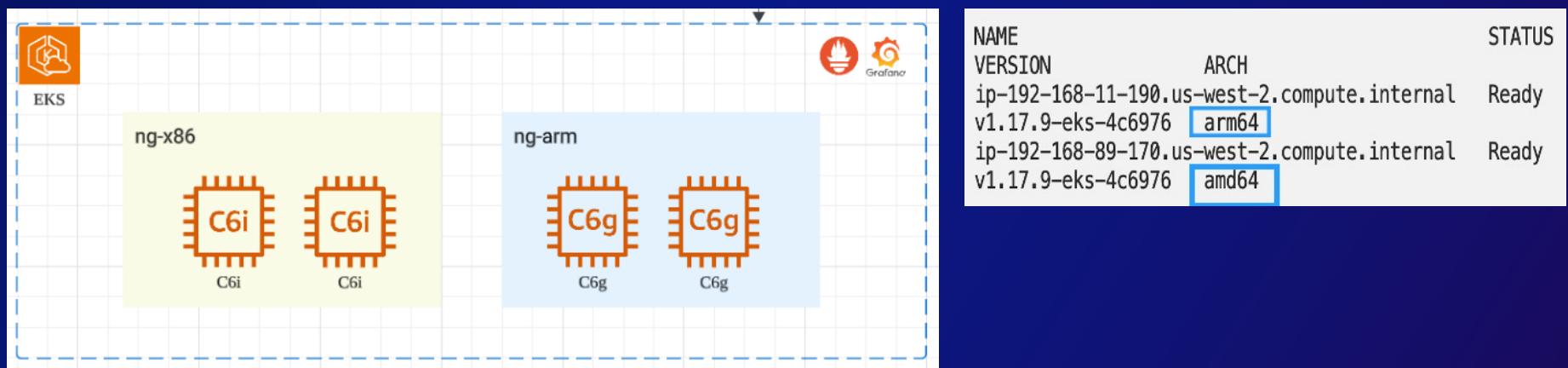


Auto scaling

Mixed nodegroups in single EKS cluster

All instances in a node group must have the following characteristics:

- Be the same instance type
- Be running the same Amazon Machine Image (AMI)
- Use the same Amazon EKS node IAM role



Scheduling pod - nodeSelector

```
apiVersion: v1
kind: Pod
metadata:
  name: my-app
spec:
  nodeSelector:
    kubernetes.io/arch: arm64
  containers:
    - name: my-app
      image: ...
```



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Scheduling pod - nodeAffinity

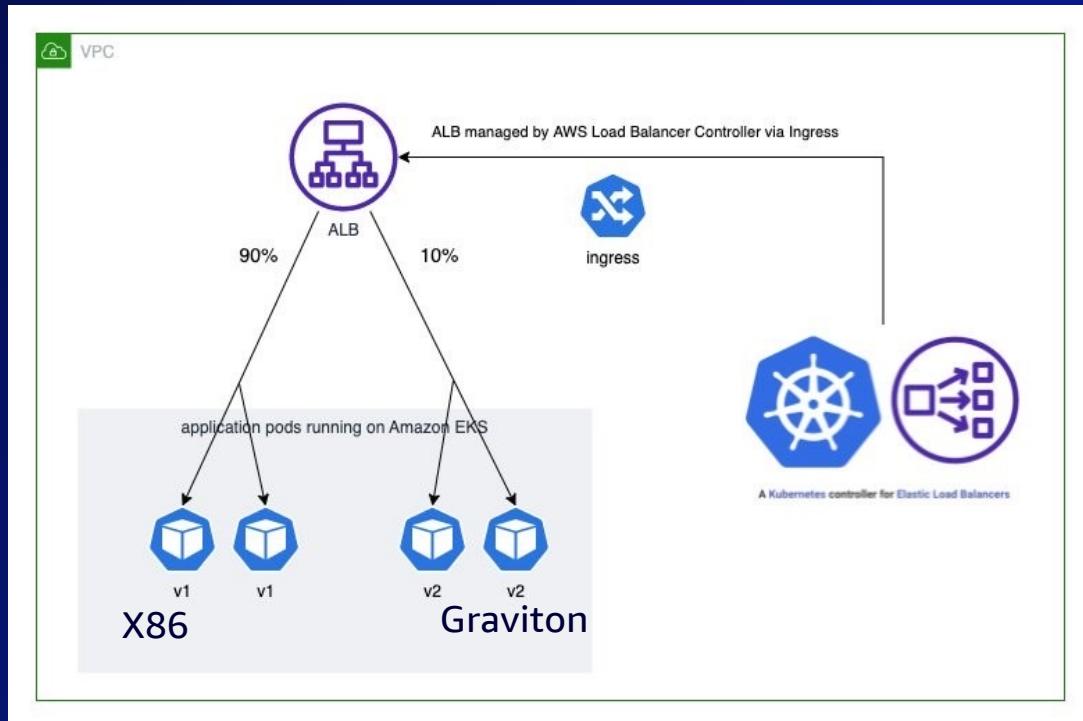
```
apiVersion: v1
kind: Pod
metadata:
  name: my-app
spec:
  affinity:
    nodeAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
        nodeSelectorTerms:
          - matchExpressions:
              - key: kubernetes.io/arch
                operator: In
                values:
                  - arm64
  containers:
    - name: my-app
      image: ...
```



Migrating traffic to Graviton nodes



AWS Load Balancer Controller for Traffic Split



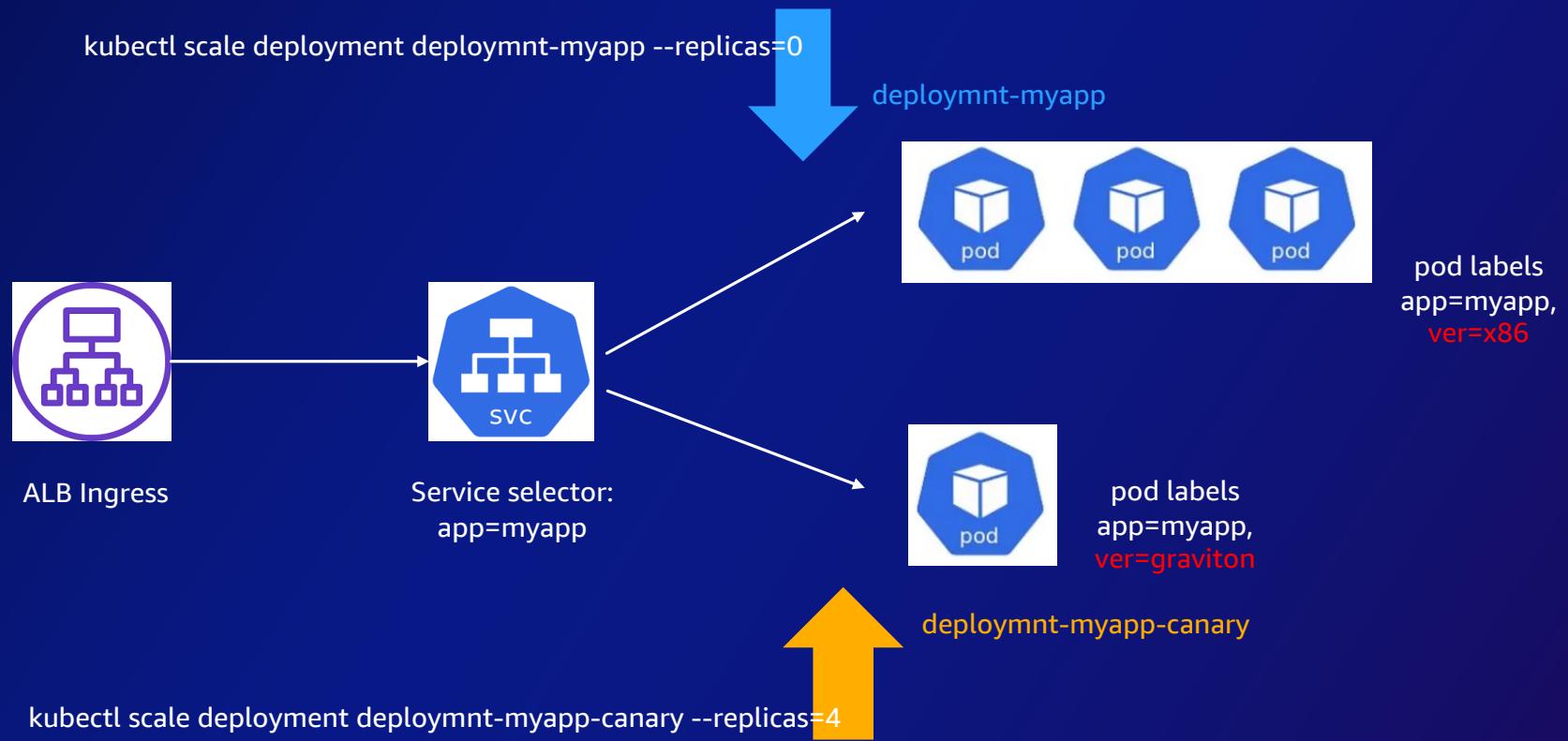
```
alb.ingress.kubernetes.io/actions.blue-green:  
{  
  "type": "forward",  
  "forwardConfig": {  
    "targetGroups": [  
      {  
        "serviceName": "hello-kubernetes-v1",  
        "servicePort": "80",  
        "weight": 90  
      },  
      {  
        "serviceName": "hello-kubernetes-v2",  
        "servicePort": "80",  
        "weight": 10  
      }  
    ]  
  }  
}  
spec:  
rules:  
- http:  
  paths:  
  - path: /  
    pathType: Prefix  
    backend:  
      service:  
        name: blue-green  
        port:  
          name: use-annotation
```

<https://aws.amazon.com/blogs/containers/using-aws-load-balancer-controller-for-blue-green-deployment-canary-deployment-and-a-b-testing/>



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Canary Deployment with Pod label variation



Service mesh - Istio

A service mesh is a dedicated infrastructure layer that manages service-to-service communication in a microservices architecture. It acts as a transparent layer that handles tasks like traffic management, security, and observability.

Istio Traffic Management is perfect solution for A/B testing, where you might want to configure traffic routes **based on percentages** across different service versions, or to direct traffic from your internal users to a particular set of instances. For example, “20% of calls go to the new version” or “calls from these users go to version 2”.

Traffic mirroring (also referred to as *shadowing*) in Istio lets you send a copy of real user traffic to a different version of a service without affecting the original request path.

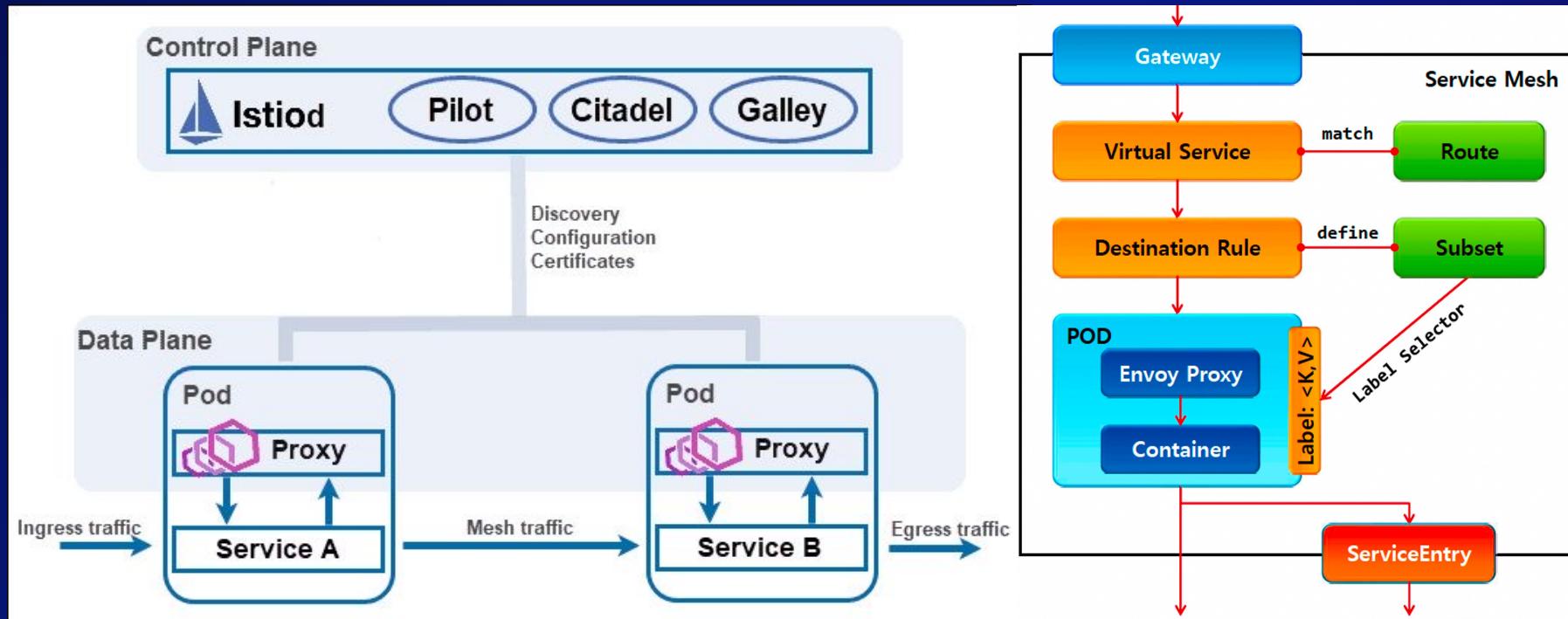


<https://istio.io/latest/docs/concepts/>



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Istio Architecture

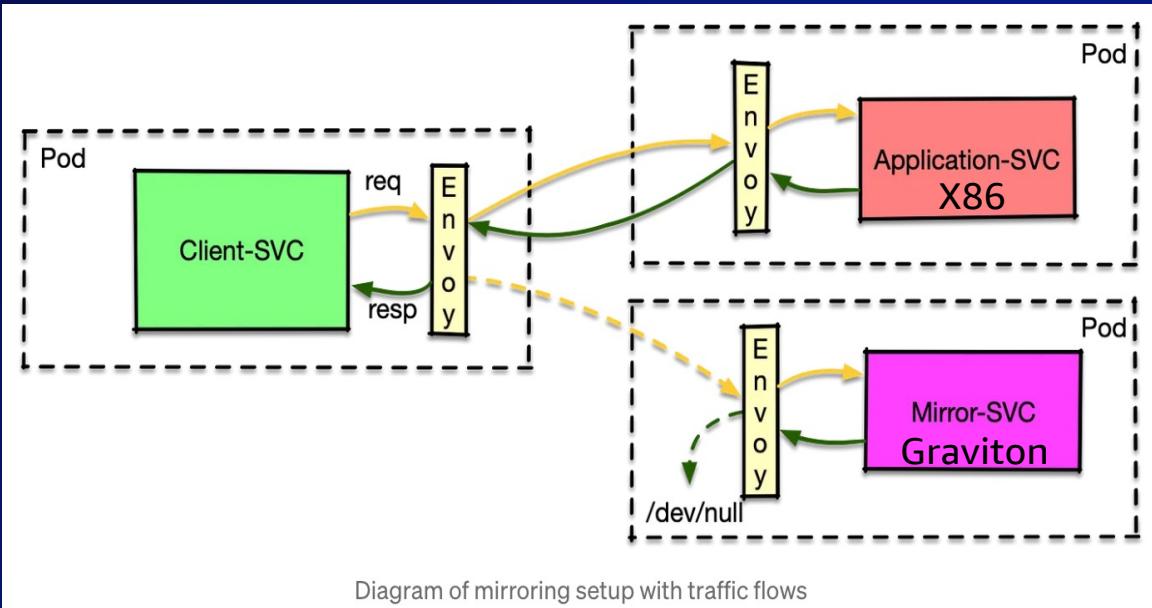


<https://dewble.tistory.com/entry/istio-traffic-management-overview>



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Istio traffic mirroring



When traffic gets mirrored, the requests are sent to the mirrored service with their Host/Authority headers appended with -shadow. For example, cluster-1 becomes cluster-1-shadow. Need to change backend DB and cache of mirror services to other system or apply to read-only service. And can make additional overhead due to traffic copy.

```
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: a-server-dr
spec:
  host: a-service
  subsets:
    - name: prod
      labels:
        version: "v0.1"
    - name: development
      labels:
        version: "devel"

apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: a-mirror
spec:
  hosts:
    - a-service
  http:
    - route:
        - destination:
            host: a-service
            subset: prod
        mirror:
          host: a-service
          subset: development
```

<https://istio.io/latest/docs/tasks/traffic-management/mirroring/>

<https://itnext.io/use-istio-traffic-mirroring-for-quicker-debugging-a341d95d63f8>

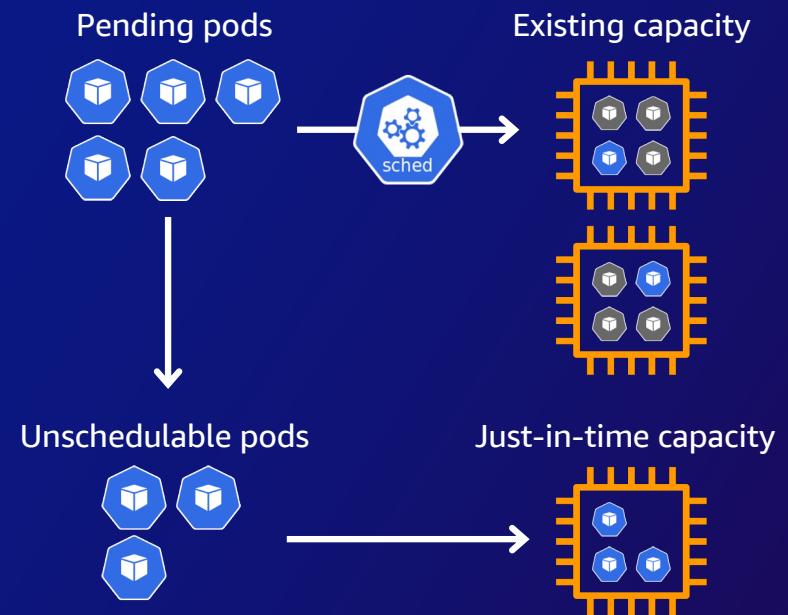


© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Auto scaling: Introducing Karpenter

Open source, flexible, and high-performance Kubernetes cluster autoscaler

- Fully supported by AWS in Amazon EKS clusters
- Work in tandem with the Kubernetes scheduler and EC2
- Launch rightsized nodes in response to unschedulable pods without EC2 Auto Scaling groups
- Natively work with Kubernetes scheduling constraints (e.g., node label selectors, pod affinity, and topology spread constraints)
- Terminates unused nodes and consolidates pods



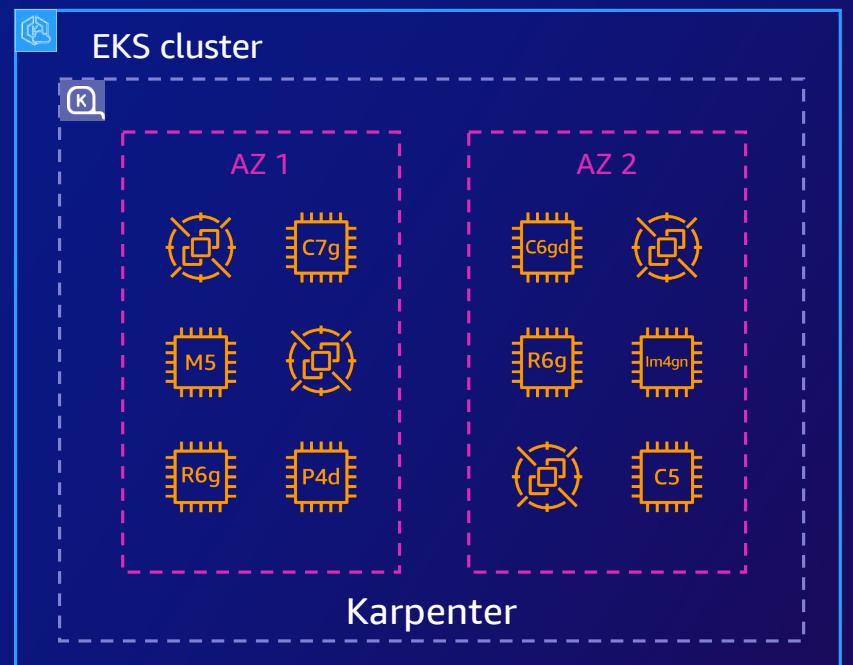
© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

<https://github.com/aws/karpenter>

Karpenter with AWS Graviton and Spot Instances

Karpenter simplifies incorporating **AWS Graviton** and **EC2 Spot Instances** to improve cluster utilization and lower costs

```
kind: NodePool
spec:
  requirements:
    - key: karpenter.sh/capacity-type
      operator: In
      values: ["spot", "on-demand"]
    - key: kubernetes.io/arch
      operator: In
      values: ["amd64", "arm64"]
```



Using Graviton with Amazon ECS

- **Amazon ECS fully supports Graviton**
- **Fargate is the easiest way to use Graviton with Amazon ECS**
 - Simply set `runtimePlatform.cpuArchitecture` to ARM64
- **You can add both Graviton and x86 EC2 instances to a cluster**
- **You can pin containers to Graviton or x86 instances**
 - Placement constraints
 - Capacity provider strategies



ECS Tasks with Capacity Provider Strategies

Capacity Provider Strategy:

Controls how tasks are assigned to CPs

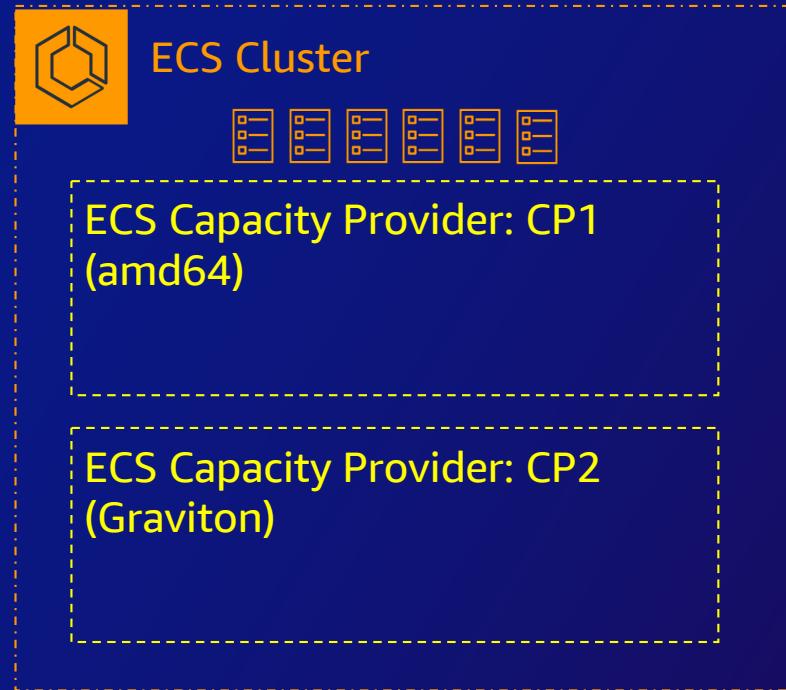
Example strategy:

CP1: weight=2

CP2: weight=1

6 total tasks:

CP1=4 tasks CP2=2 task



ECS Fargate with Graviton

Amazon Elastic Container Service > Task definitions > Create

Step 1
Configure task definition and containers

Step 2
Configure environment, storage, monitoring, and tags

Step 3
Review and create

Configure environment, storage, monitoring, and tags

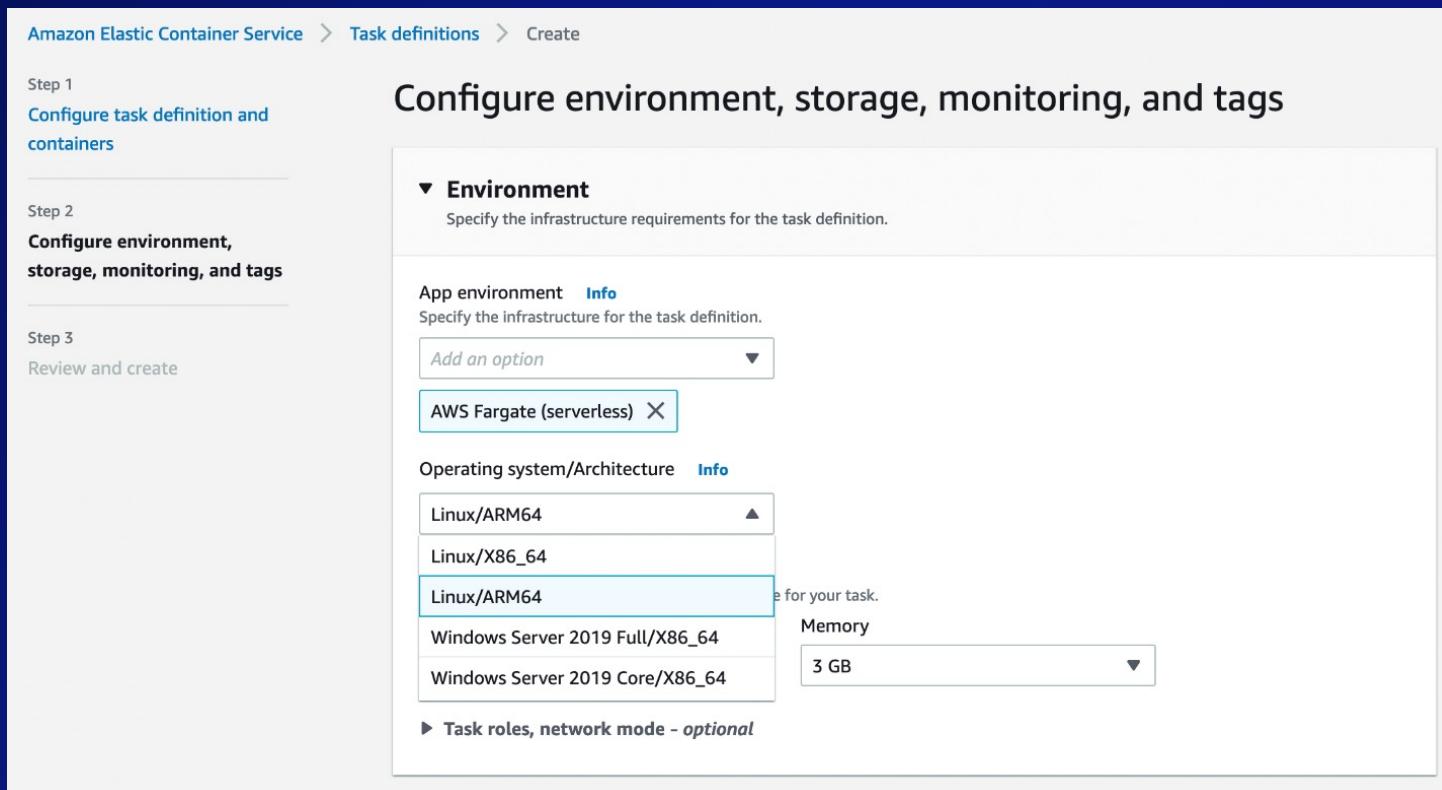
▼ Environment
Specify the infrastructure requirements for the task definition.

App environment [Info](#)
Specify the infrastructure for the task definition.
[Add an option](#) ▾
AWS Fargate (serverless) X

Operating system/Architecture [Info](#)
Select the operating system and architecture for your task.
Linux/ARM64 ▲
Linux/X86_64
Linux/ARM64
Windows Server 2019 Full/X86_64
Windows Server 2019 Core/X86_64

Memory
3 GB ▾

► Task roles, network mode - optional



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Troubleshooting



Problem

Pod crashed
Pod got stuck in 100% CPU busy loop

Solution

```
Nov 08 17:45:18.504 i-0ee97a27583db03b3 mobile-api # Distribution: Custom build (Sat May 4 17:44:26 UTC 2019)
Nov 08 17:45:18.504 i-0ee97a27583db03b3 mobile-api # Derivative: IcedTea 3.12.0
Nov 08 17:45:18.504 i-0ee97a27583db03b3 mobile-api # Java VM: OpenJDK 64-Bit Server VM (25.212-b04 mixed mode linux-aarch64 compressed oops)
Nov 08 17:45:18.504 i-0ee97a27583db03b3 mobile-api # JRE Version: OpenJDK Runtime Environment (8.0.212-b04) (build 1.8.0_212-b04)
Nov 08 17:45:18.504 i-0ee97a27583db03b3 mobile-api #
Nov 08 17:45:18.504 i-0ee97a27583db03b3 mobile-api # SIGSEGV (8xb) at pc=0x000fffff939ae86c, pid=1, tid=0x000fffff8a437aa0
Nov 08 17:45:18.504 i-0ee97a27583db03b3 mobile-api #
Nov 08 17:45:18.504 i-0ee97a27583db03b3 mobile-api # A fatal error has been detected by the Java Runtime Environment:
Nov 08 17:45:18.504 i-0ee97a27583db03b3 mobile-api #
```

Occasional pod restarts (1-2x/week)

Tuned Java runtime options
Updated dependencies
Spring dependency and Datadog agent



Compute for containers

Orchestrator	Deployment Unit	Capacity Provisioner	Scheduling Assignment
 Amazon ECS	 Task	 Serverless: AWS Fargate Self-managed: EC2 Capacity Providers	Fargate: Task definition EC2: Placement constraints
 Amazon EKS	 Pod	 Karpenter Cluster Auto Scaler	Node selector Node affinity



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Resources

- **Graviton2 Getting Started Github Repo - Technical details:**
<https://github.com/aws/aws-graviton-getting-started>

- **Graviton2 workshops:**
<https://graviton2-workshop.workshop.aws/en/gettingstarted.html>

- **Graviton Developer workshop**
<https://catalog.us-east-1.prod.workshops.aws/workshops/dcab7555-32fc-42d2-97e5-2b7a35cd008f/en-US>



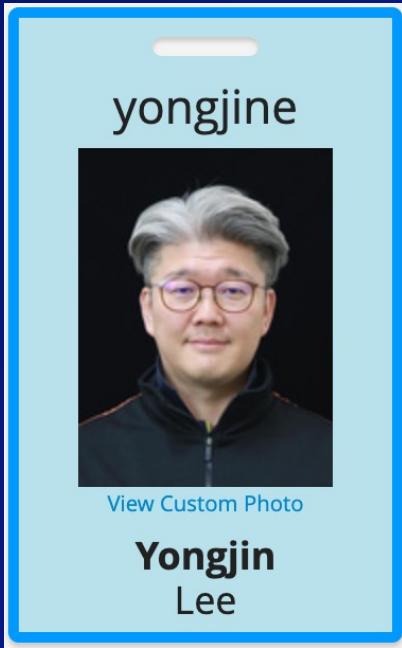
© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Hands-On Labs



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Supporters



Hands-On Labs / URL

<https://shorturl.at/HvOlR>

de17-0302ae-3f



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

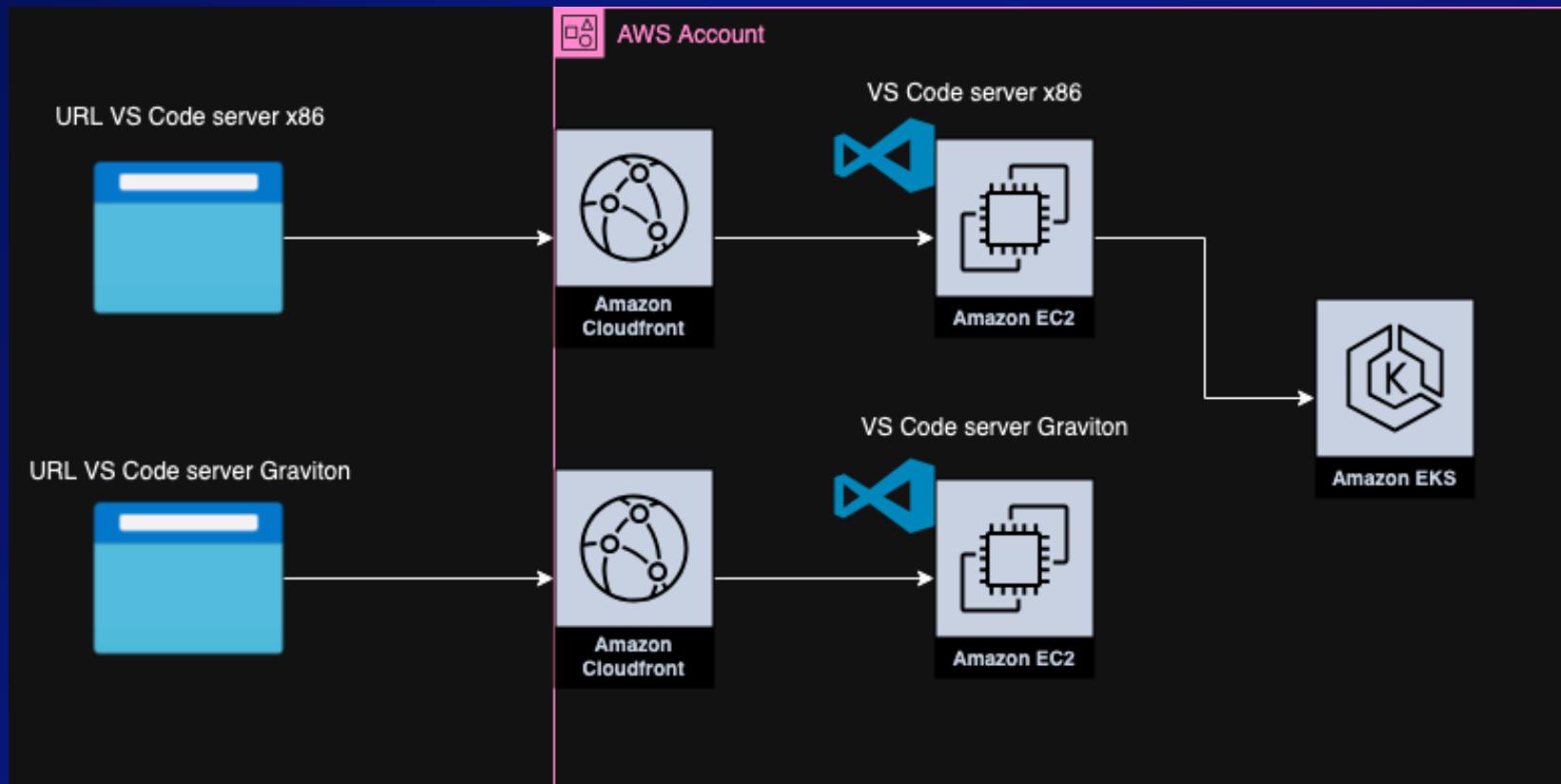
설문 URL

<https://shorturl.at/2CPvw>



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Hands-On Labs / Architecture



Thank you!



© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Appendix



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Fargate Support

- Available: ECS Fargate Spot on Graviton
 - <https://aws.amazon.com/about-aws/whats-new/2024/09/amazon-ecs-graviton-based-spot-compute-fargate/>
- Proposed: EKS Fargate on Graviton
 - <https://github.com/aws/containers-roadmap/issues/1629>
 - Alternative 1) Use EKS EC2 with Karpenter on Graviton
 - Alternative 2) Use ECS Fargate on Graviton



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Ideal Spot Workloads



Fault-tolerant



Flexible



Loosely coupled



Stateless



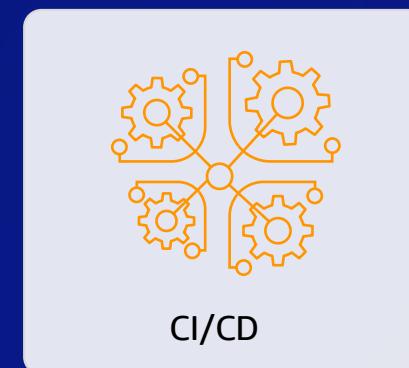
Web services



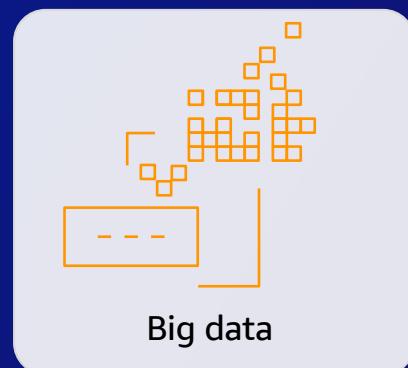
Containers



High Performance
Compute (HPC) + Batch



CI/CD



Big data



AI/ML

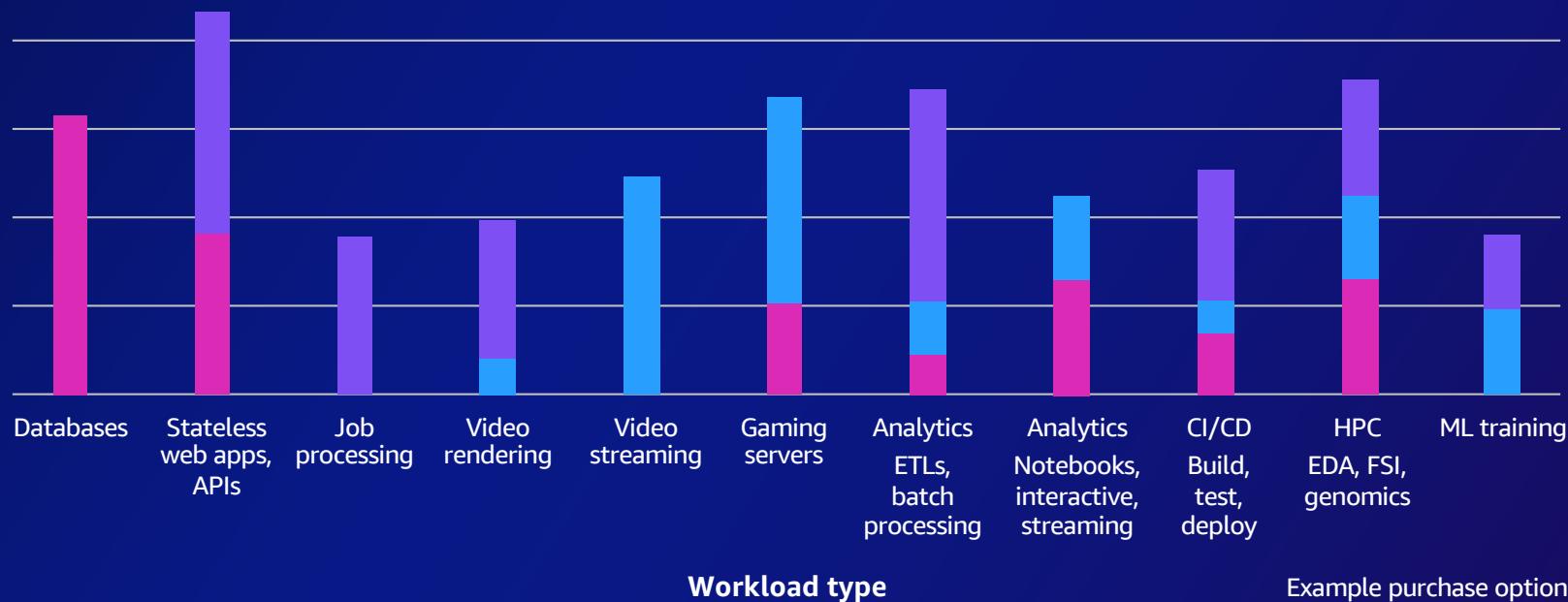
Workload types and purchase options

Use **Savings Plans** for known/steady-state workloads

Scale using **On-Demand** for new or stateful spiky workloads

Scale using **Spot Instances** for flexible, fault-tolerant workloads

Amazon EC2 usage



Example purchase option distributions

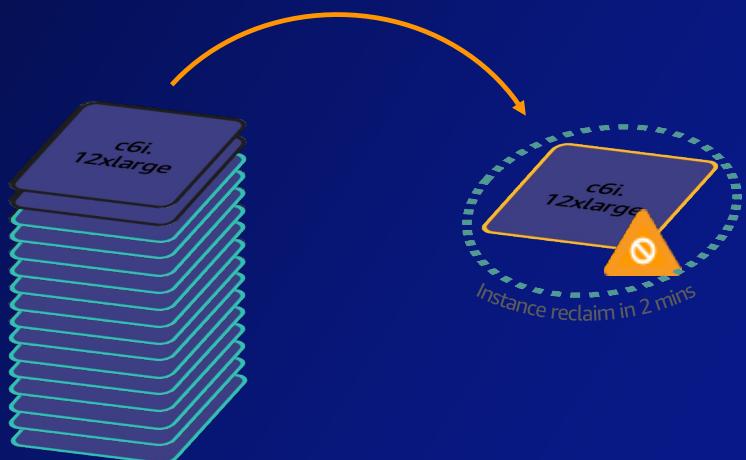


© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Interruptions

By the nature of Spot as spare-capacity, a Spot instance can be interrupted if the instance is needed by On-Demand.

AWS provides two types of notifications to enable you to handle the response in an automated way:



EC2 instance rebalance recommendation *(proactive)*



- Act when your Spot instance is at elevated risk of interruption
- Built in support for integrations such as [EC2 Auto Scaling](#) [EKS](#) [Managed Node Groups](#)



Spot instance termination notice *(reactive)*



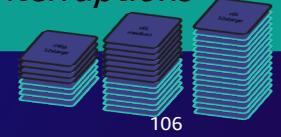
- React 2 minutes before your Spot instance is interrupted
- Built in support for the same AWS services
- Interruption handling ([AWS have recipes](#) for recommended use cases)



Historically

95% of the Spot instances launched in the last 3 months completed without interruption

Diversification across instances reduces interruptions



Amazon EC2 Spot Instance pools explained

C7g	AZ1	AZ2	AZ3	On-Demand
8XL	\$0.28	\$0.27	\$0.29	\$1.76
4XL	\$0.21	\$0.19	\$0.16	\$0.88
2XL	\$0.08	\$0.07	\$0.08	\$0.44
XL	\$0.04	\$0.05	\$0.04	\$0.22
L	\$0.01	\$0.01	\$0.02	\$0.11

Each instance family

Each instance size

Each availability zone

In every Region

Is a separate **Spot pool**



Graviton on Spot Pricing

Price as of Feb 2025 SIN Region

Performance not yet factored in

Name	API Name	Instance Memory	vCPUs	On Demand	Linux Reserved cost	Linux Spot Minimum cost
C6G Double Extra Large	c6g.2xlarge	16.0 GiB	8 vCPUs	\$0.3136 hourly	\$0.1179 hourly	\$0.1378 hourly
C7G Double Extra Large	c7g.2xlarge	16.0 GiB	8 vCPUs	\$0.3332 hourly	\$0.1315 hourly	\$0.1295 hourly
C5 High-CPU Double Extra Large	c5.2xlarge	16.0 GiB	8 vCPUs	\$0.3920 hourly	\$0.1474 hourly	\$0.1782 hourly
C6I Double Extra Large	c6i.2xlarge	16.0 GiB	8 vCPUs	\$0.3920 hourly	\$0.1548 hourly	\$0.1603 hourly



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS C7gn and Hpc7g Instances

POWERED BY AWS GRAVITON3E PROCESSORS

Graviton3E Processor

- Step up from Graviton3 for HPC workloads
- Up to 35% faster vector instruction performance (Neon/SVE SIMD) vs. Graviton3

C7gn instances Built on newest generation of AWS Nitro System

- Up to 200Gbps ENA throughput (2x more than C6gn) and up to 50% higher PPS (vs. C6gn)

Hpc7g instances

- Delivers best price/performance for HPC workloads
- Built on newest generation of AWS Nitro System
- Optimized for EFA networking @ 200Gbps



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

NEW

Amazon EC2 C7gn: Nitro v5 Card

C7GN: FIRST EC2 INSTANCE FEATURING NEXT GEN NITRO V5 CARD



Nitro v5 card:

Latest generation interfaces (PCIe Gen5, DDR5)

30% lower latency (Nitro v5 chip)

40% better performance/Watt (Nitro v5 chip)



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Characteristic of Game Server

게임에서는 대부분의 로직이 다른 로직에 영향을 끼치기 때문입니다. (정확히는 변수값)

예를 들어, A가 B를 공격한다는 패킷이 왔을 때, A가 자신에게 먼저 락을 걸고 B의 변수를 변경하기 위해 B에게도 락을 걸려고 합니다.

그런데, 동시에 B도 A를 공격하려고 했고, B도 자신에게 락을 걸고 A도 락을 걸려고 대기를 합니다.
이러면 서로 락을 걸려고 대기하는 상태가 되어 버려서, 데드락에 걸려 버리죠.

락이 안 걸리게 커맨드 큐를 만들어서 비동기로 처리하게 할 수도 있지만 그렇게 만들면 구조가 너무 복잡해지고, 만약 같은 Tick에서 바로 처리해야 하는 경우엔 답이 안 나옵니다.

애초에 변수 하나 수정할 때마다 락을 걸어야 한다면, 코딩도 복잡해집니다.

게다가 락을 건다는 것 자체도 CPU에 부하를 주기 때문에 그런 코드가 많아지면 전체적인 퍼포먼스도 별로 안 좋아지게 됩니다.

여러 가지 이유로 게임 로직의 경우 그냥 스레드 하나에서 다 처리하는 게 간편하고 안전합니다.
다른 모듈에 전혀 영향을 안 주고 혼자서만 처리가 가능한 로직이라면 딱히 상관없습니다.

싱글 쓰레드를 사용하는 경우 그라비톤이 X86에 비해 일반적으로 느린 응답 속도를 보여준다.
-> 그라비톤 CPU를 제대로 활용하기 위해서는 노드당 게임서버를 여러개 배치 한다.

