

2025 July 16th

Ren Goto

それ、覚えられる？ インスタンスタイプ ラブソディ は 狂詩曲

~ CyberAgent AWS GameDay 2025 小ネタ LT ~



Profile

Ren Goto (@ren510dev)

昨年度 サイバーエージェント SRE 新卒入社。

現在は ABEMA Cloud Platform Team にて

Google Cloud / AWS を主軸に各種 OSS を用いたマルチクラウド・クラスタの構築・運用に従事。

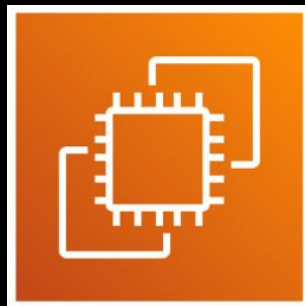


<https://x.com/ren510dev>



<https://www.ren510.dev>

Amazon EC2



【概要】

Amazon EC2 (Elastic Compute Cloud) : AWS の代表的なマネージド VM ソリューション

AWS にあまり触れてなくても

Amazon EC2 なら使ったことがあるのでは？

親しみやすい！

豊富なインスタンスタイプ

- **General Purpose**
 - T / M / A
- **Compute Optimized**
 - C
- **Memory Optimized**
 - R / U / U7i / X
- **Accelerated**
 - DL / G / Inf / P 他
- **Storage Optimized**
 - D / H1 / I
- **High-Performance**
 - Hpc

などなど。

—	General Purpose
—	T (T2, T3, T3a, T4g) – パースト可能
—	M (M5, M5a, M5ad, M5d, M5dn, M5n, M5zn, M6a, M6g, M6gd, M6i, M6id, M6idn, M6in, M7a, M7g, M7gd, M7i, M7i-flex, M8g, M8gd)
—	A (A1)
—	Mac (Mac1, Mac2, Mac2-m1ultra, Mac2-m2, Mac2-m2pro)
—	Compute Optimized
—	C (C4, C5, C5a, C5ad, C5d, C5n, C6a, C6g, C6gd, C6gn, C6i, C6id, C6in, C7a, C7g, C7gd, C7gn, C7i, C7i-flex, C8g, C8gd)
—	Memory Optimized
—	R (R5, R5a, R5ad, R5b, R5d, R5dn, R5n, R6a, R6g, R6gd, R6i, R6id, R6idn, R6in, R7a, R7g, R7gd, R7i, R7iz, R8g, R8gd)
—	U-ファミリー (High Memory: U-3tb1, U-6tb1, U-9tb1, U-12tb1, U-18tb1, U-24tb1)
—	U7i 系 (U7i-6tb, U7i-8tb, U7in-12tb, U7in-16tb, U7in-24tb, U7in-32tb)
—	X 系 (X1, X1e, X2gd, X2idn, X2iedn, X2iezn, X8g)
—	z1d
—	Storage Optimized
—	D (D2, D3, D3en)
—	H1
—	I (I3, I3en, I4g, I4i, I7i, I7ie, I8g, Im4gn, Is4gen)
—	Accelerated
—	DL (DL1, DL2q)
—	F (F1, F2)
—	G (G4ad, G4dn, G5, G5g, G6, G6e, Gr6)
—	Inf (Inf1, Inf2)
—	P (P3, P3dn, P4d, P4de, P5, P5e, P5en, P6-B200)
—	Trn (Trn1, Trn1n, Trn2, Trn2u)
—	VT1
—	HPC
—	Hpc (Hpc6a, Hpc6id, Hpc7a, Hpc7g)
—	旧世代インスタンス
—	汎用: A1?・M1, M2, M3, M4
—	コンピューティング: C1, C3, C4
—	メモリ: R3, R4
—	ストレージ: I2
—	アクセラレーテッド: G3

出典: <https://docs.aws.amazon.com/ec2/latest/instancetypes/instance-types.html>

いっぱいありますね。



多様なユースケースのために最適化された
インスタンスタイプを提供してくれています 🙇

突然ですが

これは何を表しているでしょう 🙋

u-12tb1.metal

じゃあこの辺りは？

t2.micro / m5a.large

(EC2) インスタンスタイプの名前
みんなどうやって覚えてるんですか？

と、いうお話。



よくマシンタイプを見ただけで、
それなら良さそうとか、これだと厳しいよね、的な話をしてる人がいる。



(なるほど。分からん。)

混乱したあげく毎回調べにいらてる  

ホク

Instance families and instance types

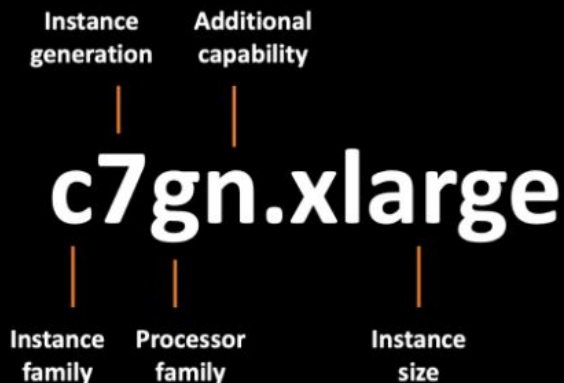
Instance family	Available instance types	Instance size	vCPU	Memory (GiB)	Instance storage (GB)	Network bandwidth (Gbps)	Amazon EBS bandwidth (Gbps)
A1	a1.medium a1.large a1.xlarge a1.2xlarge a1.4xlarge a1.metal	c8g.medium	1	2	EBS-only	Up to 12.5	Up to 10
C1	c1.medium c1.xlarge	c8g.large	2	4	EBS-only	Up to 12.5	Up to 10
C3	c3.large c3.xlarge c3.2xlarge c3.4xlarge c3.8xlarge	c8g.xlarge	4	8	EBS-only	Up to 12.5	Up to 10
C4	c4.large c4.xlarge c4.2xlarge c4.4xlarge c4.8xlarge	c8g.2xlarge	8	16	EBS-only	Up to 15	Up to 10
G3	g3.4xlarge g3.8xlarge g3.16xlarge	c8g.4xlarge	16	32	EBS-only	Up to 15	Up to 10
I2	i2.xlarge i2.2xlarge i2.4xlarge i2.8xlarge	c8g.8xlarge	32	64	EBS-only	15	10
M1	m1.small m1.medium m1.large m1.xlarge	c8g.12xlarge	48	96	EBS-only	22.5	15
M2	m2.xlarge m2.2xlarge m2.4xlarge	c8g.16xlarge	64	128	EBS-only	30	20
M3	m3.medium m3.large m3.xlarge m3.2xlarge	c8g.24xlarge	96	192	EBS-only	40	30
M4	m4.large m4.xlarge m4.2xlarge m4.4xlarge m4.10xlarge m4.16xlarge	c8g.48xlarge	192	384	EBS-only	50	40
P2	p2.xlarge p2.8xlarge p2.16xlarge	c8g.metal-24xl	96	192	EBS-only	40	30
R3	r3.large r3.xlarge r3.2xlarge r3.4xlarge r3.8xlarge						
R4	r4.large r4.xlarge r4.2xlarge r4.4xlarge r4.8xlarge r4.16xlarge						
T1	t1.micro						



ルールはちゃんとある

[Documentation](#) > [Amazon EC2](#) > Instance Types

Amazon EC2 instance type naming conventions



Instance Family: 用途ベースのカテゴリ (t / c / r / g 他)

Generation: ハードウェア世代 (大きいほど新しい・高性能・高効率)

Processor Family: CPU 種別 (AMD / ARM / Xenon / Intel / Graviton 他)

Additional Capability: 性能強化や特殊構成 (EBS / Network 最適化 他)

Instance Size: vCPU 数とメモリ量のスケールリング (large / 2xlarge 他)

順番に見ていく

出典: <https://docs.aws.amazon.com/ec2/latest/instancetypes/instance-type-names.html>

複雑化する Processor Family + Additional Capability

【古いインスタンスタイプ】

- 例 : m3.large / c4.xlarge / r2.2xlarge
- 常に「Family + Gen.Size」で規則的な形
- “.”の前は常に数字で理解しやすい

【近年のインスタンスタイプ】

- 例 : m5a.large / m5d.large / c6gn.large
- (拡張機能を表す) 英字を前後に挿入
- 英数字の混在表記で複雑化してきた

m3.large

→ 第3世代の汎用インスタンス

c4.2xlarge

→ 第4世代のコンピュータ最適化



m5d.large

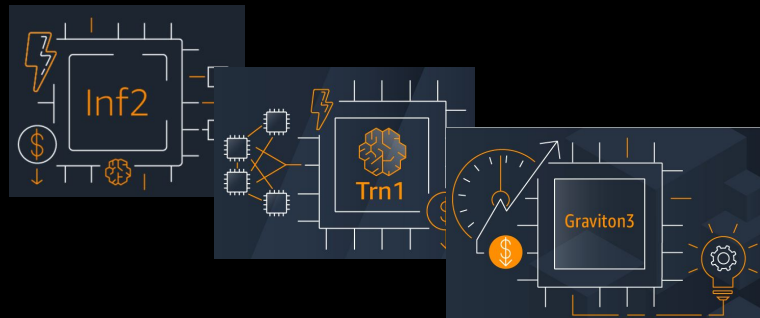
→ 第5世代 + NVMe ディスクあり

c6gn.large

→ 第6世代 + Graviton + 高帯域

ハイブリット世代になってくると...

- ✓ Graviton 系 (記号:g) の登場
- ✓ Trainium 系 (記号:trn) の登場
- ✓ Inferentia 系 (記号:inf) の登場



- 複数の機能が混ざった記号列 idn, gn, in, 等 が登場 (順番は AWS が決める)
 - 例: m6idn.large / c6in.2xlarge / r6id.large
- 「専用チップ + 特化ファミリ」で命名体系が独立したものが登場
 - 例: inf2.24xlarge / p4d.24xlarge

バリエーションの数が増えるごとに組み合わせパターンが爆発的に増加

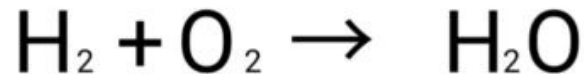
化学式みたい 

【化学反応式のつくり方】

1.化学変化を物質名と式で表す

水素 + 酸素 → 水

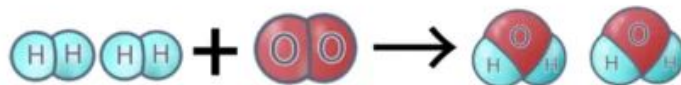
2.物質名を化学式にする



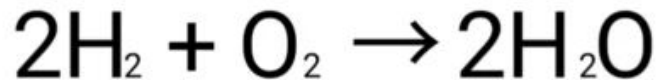
3.化学式をモデルにする



4.矢印の左右で原子の数が等しくなるように分子をふやす



5.モデルを化学式にする



再掲

これは何を表しているでしょう 🙋

u-12tb1.metal

u-12tb1.metal

コレナニ ??



あれ、命名規則に当てはまらない？

Amazon EC2 High Memory (U-1)

[Amazon EC2](#) > [Instance types](#)

Amazon EC2 High Memory (U-1) Instances

Built on AWS Nitro System with up to 6TiB of memory in a single instance

u-3tb1.56xlarge	224	3,072
u-6tb1.56xlarge	224	6,144
u-6tb1.112xlarge	448	6,144
u-6tb1.metal**	448	6,144
u-9tb1.112xlarge	448	9,216
u-9tb1.metal**	448	9,216
u-12tb1.112xlarge	448	12,288
u-12tb1.metal**	448	12,288
u-18tb1.112xlarge	448	18,432
u-18tb1.metal	448	18,432
u-24tb1.112xlarge	448	24,576
u-24tb1.metal	448	24,576

【概要】

- AWS が提供する超大容量メモリ特化型インスタンスファミリー第 1 世代
- SAP HANA などのインメモリデータベース向けに設計された専用インスタンス

出典: <https://aws.amazon.com/ec2/instance-types/high-memory>

合理的な例外

「**Family**+**Gen**(Processor Family + Additional Capability).**Size**」でしたよね

u-12tb1.metal

- 通常の命名規則に当てはまらない
- インスタンス名に「12TB」が直書き
- 命名構成が事実をそのまま書いただけ

High Memoryなのに m でも r でも x でもない → **u-** という Family

12TB 積んでるのに .384xlarge ではない → **12tb1** が 容量+世代 を代弁

u-12tb1.metal



僕は 12TB の超大規模メモリ(多分 Ultra memory)を搭載してますよ！

命名規則というか自己紹介

※ もはや呪文

【てんこ盛りパターン】

- x2idn.32xlarge: 高 IO + Local SSD + 100GB/s interconnects
- x2iedn.32xlarge: 強化メモリ + 高速キャッシュ + 高性能インメモリ型
- x2iezn.12xlarge: 4.5GHz Turbo-Clock + 高速・高メモリ構成

【カオスパターン】

- trn1.32xlarge: NeuronCore × 16 (vCPU はちゃんと 128)

インスタンスタイプの数：2025年7月現在

なんと...

850種類超！

直近では“ほぼ毎月”リリース

(Google Cloud は 約400種類)

Amazon Web Services Pricing More Tools ▾

Hi Instance Types Locations

Instance Types (All) 922 Intel 475 AMD 163 AWS 280 Apple 4 G

Amazon EC2 Instance Types

Amazon Web Services (AWS) offers 922 Amazon Elastic Compute Cloud (EC2)

Instance Type				
Name	vCPU	Base Clock	Memory	Network Performance
u7in-32tb.224xlarge	896	1.9 GHz	32768 GB	200 Gigabit
u7in-24tb.224xlarge	896	1.9 GHz	24576 GB	200 Gigabit
u7in-16tb.224xlarge	896	1.9 GHz	16384 GB	200 Gigabit
u7i-12tb.224xlarge	896	1.9 GHz	12288 GB	100 Gigabit
u7i-8tb.112xlarge	448	1.9 GHz	8192 GB	100 Gigabit
u7i-6tb.112xlarge	448	1.9 GHz	6144 GB	100 Gigabit
u-6tb1.112xlarge	448	2.1 GHz	6144 GB	100 Gigabit
u-6tb1.56xlarge	224	2.1 GHz	6144 GB	100 Gigabit
u-3tb1.56xlarge	224	2.1 GHz	3072 GB	50 Gigabit
x8g.metal-48xl	192	2.7 GHz	3072 GB	50 Gigabit
x8g.48xlarge	192	2.7 GHz	3072 GB	50 Gigabit
r8gd.metal-48xl	192	2.7 GHz	1536 GB	50 Gigabit
r8gd.48xlarge	192	2.7 GHz	1536 GB	50 Gigabit

出典: <https://aws-pricing.com/instances.html> / <https://aws.amazon.com/ec2/instance-explorer/> / <https://instancetyp.es/> / <https://gcloud-compute.com/instances.html>

これだけあると…

- 命名規則に一貫性を維持しきれていないケースも
- 膨大なラインナップで命名表現が **狂詩曲** と化す
- さらにインスタンスタイプが増えたらどうなる？

Ask the Bedrock: ちょっと聞いてみた

※(まだ)存在してません

- **c7abcde.2xlarge**: Additional Capability 肥大化
 - a: AMD
 - b: ベンチマーク強化 (Benchmark-Optimized)
 - c: キャッシュ強化 (L3 共有)
 - d: Disk (NVMe)
 - e: Energy efficient (消費電力最適化)
- **ai100g.64xlarge**: エッジ AI + 低エネルギー
 - ai: AI 特化ファミリー (NeuronCore v3 搭載)
 - 100g: Graviton + 100GB/s interconnects

Amazon Bedrock

The easiest way to build and scale generative AI applications with foundation models

Get started with Amazon Bedrock

Try free demo

What is Amazon Bedrock?

Amazon Bedrock is a fully managed service that offers a choice of high-performing foundation models (FMs) from leading AI companies like [AI21 Labs](#), [Anthropic](#), [Cohere](#), [DeepSeek](#), [Luma](#), [Meta](#), [Mistral AI](#), [Poolside](#), [Stability AI](#), [TwelveLabs](#), [Writer](#), and [Amazon](#) through a single API, along with a broad set of capabilities you need to build generative AI applications with security, privacy, and responsible AI. Using Amazon Bedrock, you can easily experiment with and evaluate top FMs for your use case, privately customize them with your data using techniques such as fine-tuning and Retrieval Augmented Generation (RAG), and build agents that execute tasks using your enterprise systems and data sources. Since Amazon Bedrock is serverless, you don't have to manage any infrastructure, and you can securely integrate and deploy generative AI capabilities into your applications using the AWS services you are already familiar with.



Amazon Bedrock

- インスタンスタイプはいっぱいある！
- 大事なものは名前じゃなくて中身だよね！
- ユースケースに最適なインスタンスタイプを選ぼう！



2025 July 16th

Ren Goto

それ、覚えられる？ インスタンスタイプ ラブソディ は 狂詩曲

~ CyberAgent AWS GameDay 2025 小ネタ LT ~

おしまい