

AI Game Forge

AI が 3分でマダミスを作り、 AI と一緒に遊ぶ



AI が作る、 AI と遊ぶ

Murder Mystery

01 約 3 分でフル生成

テキスト・画像・音声・動画・ 3D まで一式

02 NPC は AI が即興で議論

Bedrock Claude が容疑者を演じて応答

03 AWS フルマネージドのサーバーレス

Lambda + Step Functions + EKS Auto Mode

04 scale-to-zero でアイドル時 ¥ 0

KEDA + Karpenter で GPU は使う時だけ



押しポイント — 速い・議論する・安い

01 — SPEED

3

分でフル生成

- 脚本 (Bedrock Claude)
- シーン画像 / 手がかり画像
- TTS ナレーション / BGM / SE
- ボイスドラマ動画 + 3D キャラ

02 — INTERACTION

AI と議論

NPC は即興で応答

- AppSync Events で WebSocket 配信
- SQS FIFO で 1 ターンずつ順次
- Bedrock tool-use で容疑者を演じる
- TTS + 口パクで音声化

03 — COST

¥ 0

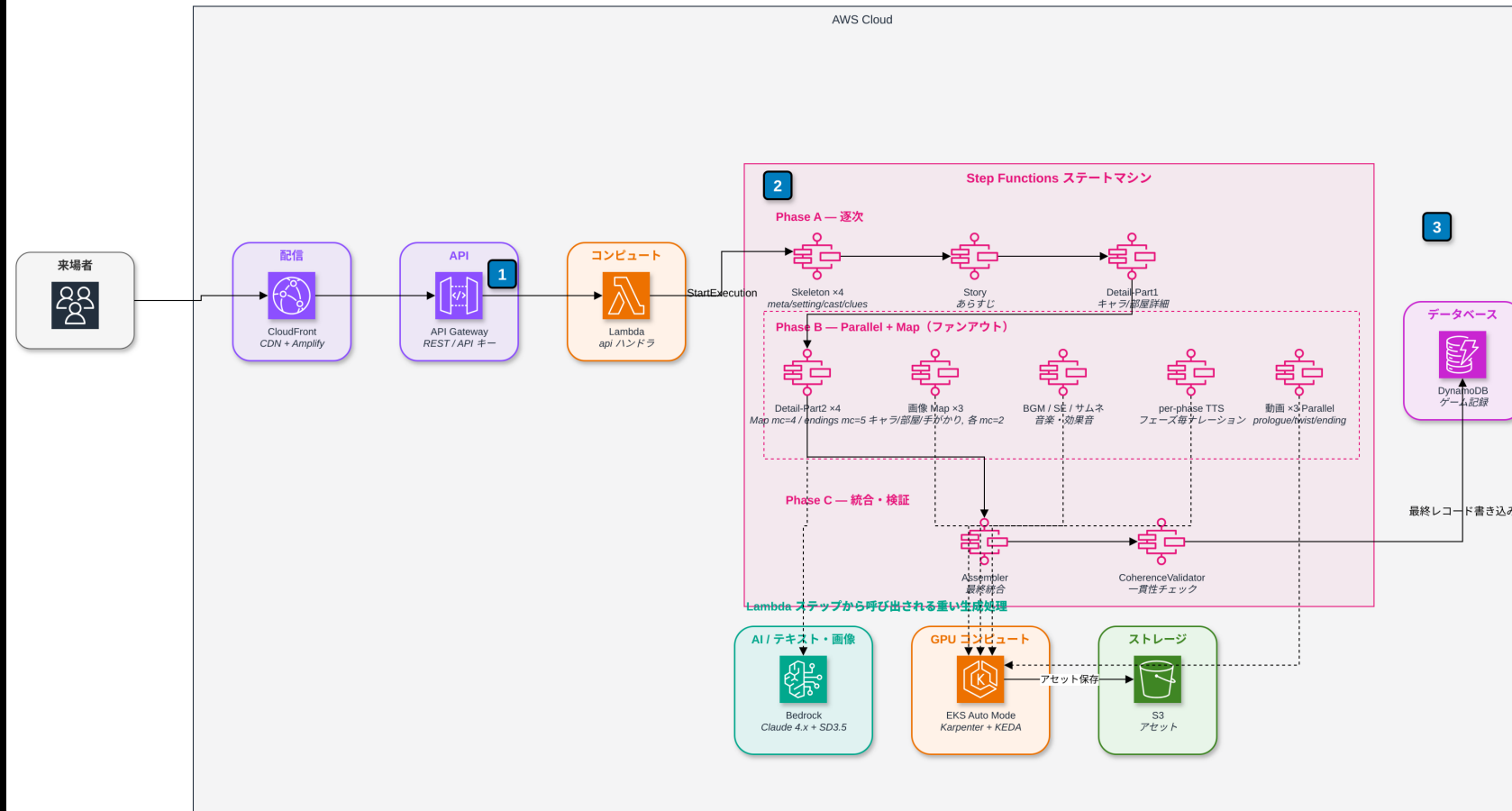
アイドル時のコスト

- KEDA で pod を 0 まで縮退
- Karpenter で GPU node も解放
- Lambda + DDB は従量課金
- イベント終了後は ¥ 0/h で待機

技術詳細① — ゲーム生成パイプライン

AI Game Forge — 生成パイプライン

API Gateway + Lambda + Step Functions が、約3.5分のゲーム生成を統括するサーバーレス制御プレーン



生成の流れ

1. リクエスト
ブラウザ → CloudFront → API Gateway (REST/APIキー)
→ Lambda が Step Functions を起動。

2. 生成 (約3.5分)
Phase A (逐次: Skeleton × 4 → Story → Detail-Part1) の後、Phase B が Parallel + Map でファンアウト。Detail-Part2 はフェーズ毎 Map (mc=4) と endings Map (mc=5) に分裂、画像は char/room/clue の 3 Map (各 mc=2)。Lambda ステップが Bedrock と EKS GPU を呼び。

3. 統合・検証・保存
Phase C は Assembler が最終統合 → CoherenceValidator が一貫性チェック → DynamoDB に書き込み。アセットは S3 経由で CloudFront 配信。

EKS GPU ワークロード
Karpenter NodePool (g5/g6/g6e, on-demand)、各 pod 1 GPU、KEDA で scale-to-zero。

- 画像 vllm-omni (FLUX.1-schnell): max 6
- Irodori TTS v2/v3: 各 max 6
- 口バク MuseTalk: max 2
- BGM ACE-Step: max 1
- サムネ Qwen-Image: max 1

上限: gpu 12 GPU + gpu-low (irodori 専用) 4 GPU

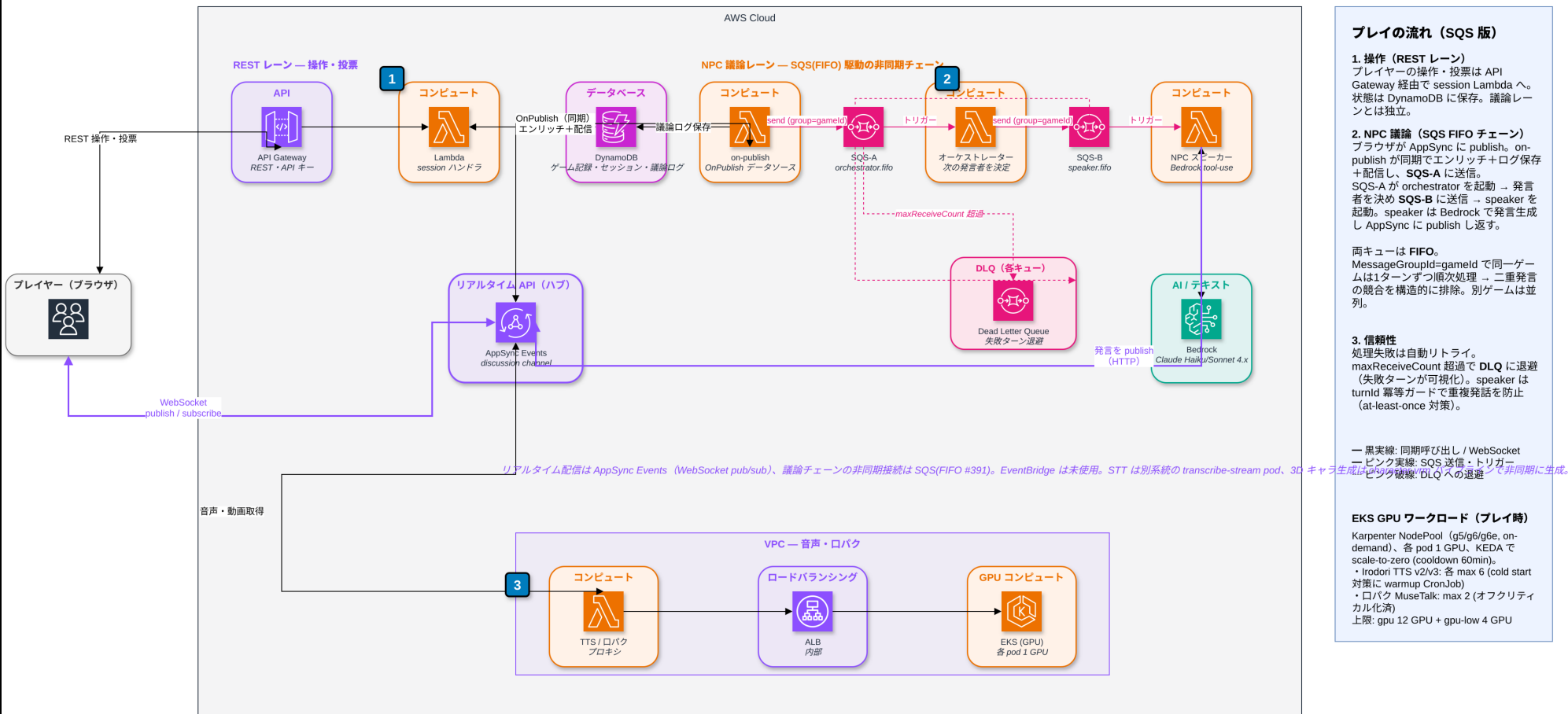
Step Functions Phase A (Skeleton × 4 → Story → Detail-Part1) → Phase B (Parallel + Map × 5) → Phase C (Assembler → CoherenceValidator → DynamoDB)



技術詳細② — リアルタイムプレイ (NPC 議論)

AI Game Forge — リアルタイムプレイ (現行)

REST 操作・投票は session レーン、NPC 議論は AppSync Events を中心に SQS(FIFO) で進む



REST レーン (操作・投票) + NPC 議論レーン (AppSync Events + SQS FIFO 2 段) + 音声・ロパク (VPC 内 ALB → EKS GPU)



技術詳細④ — Bedrock と OSS のハイブリッド

選定軸 — レイテンシ要件 × 品質 × コスト。テキストは Bedrock、リッチメディア (画像 / 音声 / 動画 / 3D) は OSS on EKS。

用途	モデル	実行場所	選んだ理由	代表的レイテンシ
脚本・NPC 対話	Claude Haiku/Sonnet/Opus 4.x	Bedrock	品質を 3 段で切替・prompt cache・最大規模	数秒 / call
シーン・手がかり画像	FLUX.1-schnell (vllm-omni)	EKS GPU (L4)	高速 8-step 推論・OSS でコスト最適	約 3 秒 / 枚
キャラクターポートレート	SD3.5 Large	Bedrock	マネージドで運用負荷ゼロ	約 6 秒 / 枚
ナレーション / NPC 音声	Irodori TTS v2 / v3	EKS GPU (L40S)	日本語話者の品質・声の一貫性	実時間比 ~0.3x
BGM / SE / サムネ	ACE-Step / Qwen-Image	EKS GPU	ジャンル指定で雰囲気付け	数秒 / 曲
口パク動画 / 3D キャラ	MuseTalk / Hunyuan3D-2	EKS GPU (別 namespace)	リップシンク + VRM 出力	~10 秒 / クリップ



技術詳細⑤ — スケール戦略とコスト最適化



LEVER 01 — POD SCALE

KEDA で 0 → N pod

CloudWatch GameForge/GPU/Activity メトリクス連動。
minReplicaCount=0、cooldownPeriod=3600s。
アイドル 60 分で全 pod が完全停止。

→ pod 課金がデモ時間のみに



LEVER 02 — CLUSTER OPS

EKS Auto Mode + Karpenter

コントロールプレーンも node も AWS 任せ。
GPU は要件 (L4/L40S) を指定するだけで Karpenter が選定・調達。
ノードのライフサイクル・パッチ・OS 更新も気にしない。

→ クラスタ運用に手が要らない



LEVER 03 — TOKEN COST

Bedrock prompt cache

Skeleton × 4 (meta/setting/cast/clues) で
共通の世界観プロンプトを cache hit。
Detail-Part2 の各 Map 呼び出しでも再利用。

→ 入トークンコスト約 1/10

THANK YOU

ブースで
遊んでください。

AI Game Forge Bedrock Claude × OSS GPU on EKS — 3分マダミス、AI と議論

Step Functions + Lambda + DynamoDB + AppSync + SQS FIFO + Karpenter + KEDA on EKS Auto Mode / ap-northeast-1

