

# Games in the Cloud 2



Juan Davalos

Purva Choudhary

Mayur Rahangdale

Narbeh Movsesian

# Table of Contents

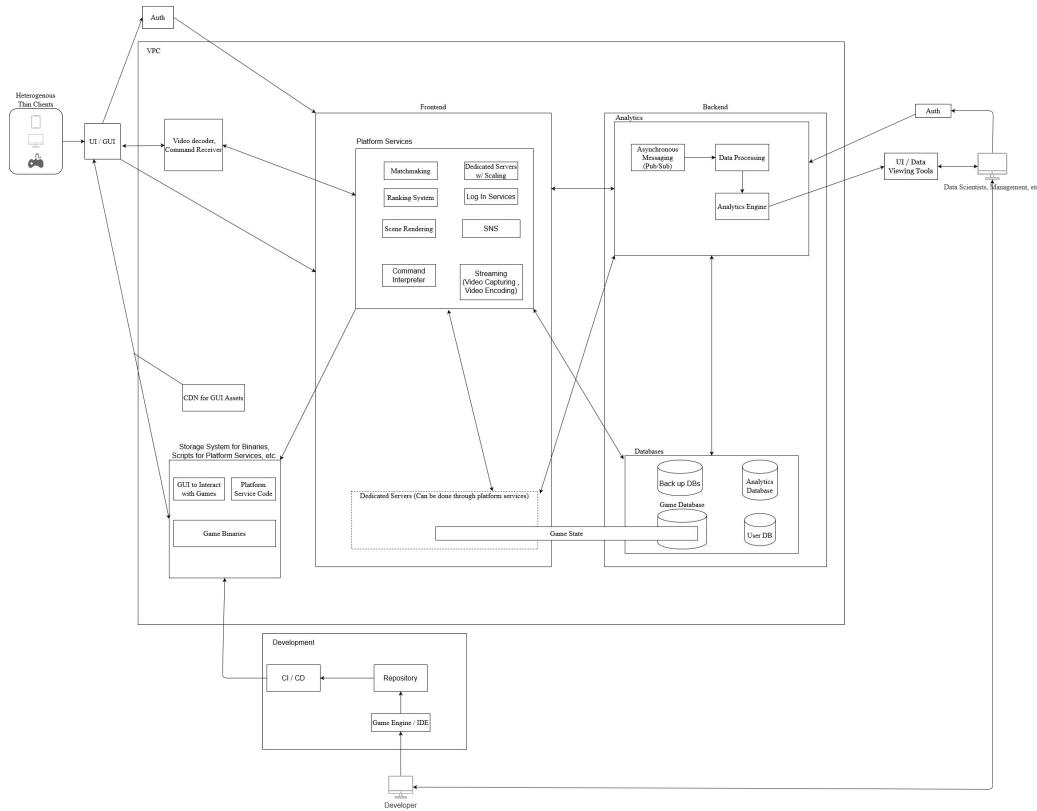
- Backtracking: What makes a good game?
- A Look Back at Our Reference Architecture
- Implementing using AWS
  - How and Why?
- Alternatives to this Solution
  - Using other Cloud Providers
  - Using Stadia and similar services

# Time Traveling Back - What Makes a Good Game?

- To put it in the most simple way, it's subjective.
  - If we all took a survey we would see some pretty hefty differences in what we call a "good" game.
- Although there are number of subjective factors that play into a good game, there are four factors to always keep in mind:
  - Gameplay
  - Story
  - Art design/ Graphics
  - Challenge
- Even these four factors will be different depending on perspective
  - From an economic point of view we might say a game that has larger profits is better.



# A Look Back at the World Map



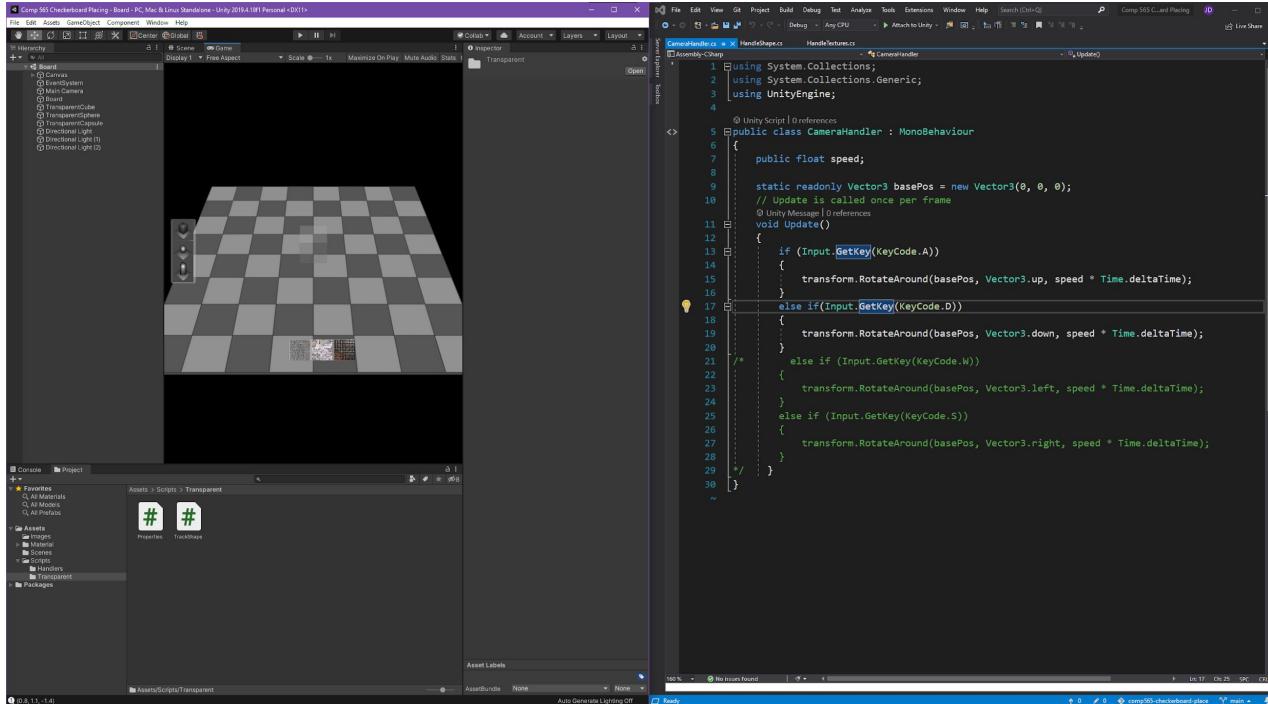
# The Crafting Table - Game Engines

- Game Engines number in the tens with support for different platforms, graphics library, and dimensions.
- We leave the game engine selection to the developer since every game has different needs.
- Three notable and popular engines to keep in mind:
  - Unreal Engine 4 by Epic Games: support for 3D games.
  - Unity Engine by Unity Technologies: support for 2D and 3D games.
  - AWS Lumberyard by Amazon.com Inc. for its integration with AWS.

# The Crafting Table - Game Engines

- Unreal Engine 4
  - Highly optimized 3D game engine used in many AAA games including Borderlands 3 (2019), Shin Megami V (TBA), Minecraft: Dungeons (2020), Valorant (2020), and Final Fantasy VII Remake (2020).
  - Requires License to develop once revenue exceeds \$1,000,000 USD.
- Unity
  - Simple to understand with practice and optimized for 2D and 3D games including Battletoads (2020), League of Legends: Wild Rift (2020), Phasmophobia (2020), GTFO(2019), and Untitled Goose Game (2019)
  - Requires license to develop after a \$100,000 USD threshold in revenue or funding.
- AWS Lumberyard
  - Complex in nature but based off of the Crytek CryEngine for 3D games. Notable games include Star Citizen (TBA) and New World (2021).
  - No license required but AWS expects users to develop with AWS services.
  - Integrates well with AWS services.

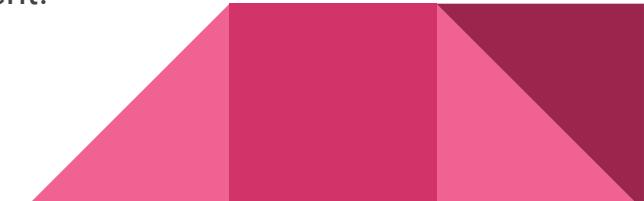
# The Crafting Table - Game Engines: A Look at Unity



# A Backpack for Our Items: The Repository



- Storing the Client-side UI code in a Github repository.
  - Github Organisation for Version Control and Administrative roles
- This practice helps in separating game code and UI code.
  - Game code stored as binaries (Discussed later)
  - Github-LFS for vital assets required for UI side
- Git for asset files(art).
  - Github allow you to "diff" the art and see what's changed. Others will just treat the assets as binary data, so you'll have to check it out to see what's different.



# A Backpack for Our Items: The Repository (Cont.)

master ▾ 2 branches 0 tags

Go to file Add file ▾ Code ▾

mayurcybercz test change	6376562 13 days ago	10 commits
dist	added more graphics	13 days ago
images	initial files	13 days ago
src	test change	13 days ago
.eslintrc.json	initial files	13 days ago
.gitignore	added dist, recovered paths	13 days ago
.stickler.yml	initial files	13 days ago
Procfile	initial files	13 days ago
babel.config.js	initial files	13 days ago
babelrc.js	initial files	13 days ago
package-lock.json	initial files	13 days ago
package.json	initial files	13 days ago
server.js	added dist, recovered paths	13 days ago
webpack.config.js	initial files	13 days ago

About

<http://checkers.us-west-1.elasticbeanstalk.com>

---

Releases

No releases published  
[Create a new release](#)

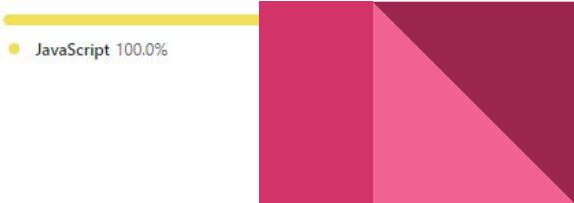
---

Packages

No packages published  
[Publish your first package](#)

---

Languages



JavaScript 100.0%

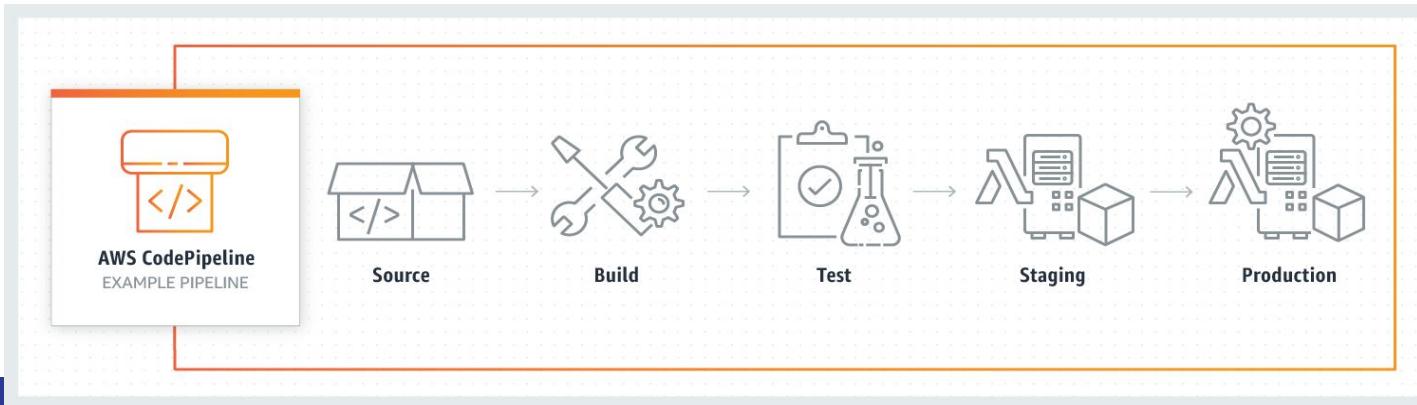
# Creating Key Quest Items - CI/CD for Our Games

- CodePipeline automates the build, test, and deploy phases of your release process every time there is a code change, based on the release model we define.
  - This enables us to rapidly and reliably deliver features and updates



AWS CodePipeline

- Easily integrate AWS CodePipeline with third-party services such as GitHub.



# Creating Key Quest Items - CI/CD for Our Games (Cont.)

Developer Tools > CodePipeline > Pipelines > checkers

## checkers

**Source** Succeeded  
Pipeline execution ID: 06a935e6-fa3b-40a0-9aff-c394069fa13d

Source GitHub (Version 1) ⓘ  
Succeeded - 12 days ago 6376562f ⓘ  
6376562f ⓘ Source: test change

Disable transition

**Deploy** Succeeded  
Pipeline execution ID: 06a935e6-fa3b-40a0-9aff-c394069fa13d

Deploy AWS Elastic Beanstalk ⓘ  
Succeeded - 12 days ago 6376562f ⓘ Source: test change

Notify ▾ Edit Stop execution Clone pipeline Release change

# Creating Key Quest Items - CI/CD for Our Games (Cont.)

Developer Tools > CodePipeline > Pipelines > checkers > Execution history > 06a935e6

Pipeline execution: 06a935e6

[Stop execution](#)

[Previous execution](#)

[Next execution](#)

## Execution summary

[Copy pipeline execution ID](#)

[View revisions](#)

Status

Succeeded

Started

12 days ago

Completed

12 days ago

Duration

1 minute 12 seconds

Trigger

**Webhook** - arn:aws:codepipeline:us-west-1:205882551351:webhook:checkers--Source--mayurcyberczcomp680website--2076088357

Latest action execution message

-

[Timeline](#)

[Visualization](#)

## Actions

[View execution details](#)

	Action name	Stage name	Status	Action provider	Started	Completed	Duration
<input type="radio"/>	Deploy	Deploy	Succeeded	AWS Elastic Beanstalk	12 days ago	12 days ago	1 minute 6 seconds
<input type="radio"/>	Source	Source	Succeeded	<a href="#">GitHub (Version 1) </a>	12 days ago	12 days ago	5 seconds

# A Backpack for Our Key Items - Storing the Binaries

- As we mentioned before there are many ways to store different binaries of our game code, platform scripts, and infrastructure code.
  - Once the CI/ CD pipeline is done running all of its processes it's always good to store the output somewhere.
- For this we use AWS S3 buckets to easily integrate with all of the other parts in the infrastructure.
- Storage is relatively simple in S3 from a pipeline and from the UI
  - Just a matter of setting up pipeline with the UI provided by CI/CD software or dragging and dropping files in the AWS Console.
  - Additionally policies for selected roles can be set to determine which pipeline can push to a specific bucket.

# A Backpack for Our Key Items - Storing the Binaries(Cont.)

Upload succeeded  
View details below.

## Upload: status

The information below will no longer be available after you navigate away from this page.

### Summary

Destination	Succeeded	Failed
s3://megafrogaceserverbucket	1 file, 2.4 KB (100.00%)	0 files, 0 B (0%)

**Files and folders** Configuration

Files and folders (1 Total, 2.4 KB)

Name	Folder	Type	Size	Status	Error
MegaFrogRaceServer.zip	-	application/x-zip-compressed	2.4 KB	Succeeded	-

Feedback English (US) © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

File Edit Selection View Go Run Terminal Help MegaFrogRaceServer.js - Visual Studio Code

JS MegaFrogRaceServer.js

```
C:\> Users > Juan > Documents > Temp > JS MegaFrogRaceServer.js ...  
22 // Any positive op code number can be defined here. These should match your client code.  
23 const OP_CODE_CUSTOM_OP1 = 111;  
24 const OP_CODE_CUSTOM_OP1_REPLY = 112;  
25 const OP_CODE_PLAYER_ACCEPTED = 113;  
26 const OP_CODE_DISCONNECT_NOTIFICATION = 114;  
27  
28 // Example groups for user-defined groups  
29 // Any positive group number can be defined here. These should match your client code.  
30 // When referring to user-defined groups, "-1" represents all groups, "0" is reserved.  
31 const RED_TEAM_GROUP = 1;  
32 const BLUE_TEAM_GROUP = 2;  
33  
34 // Called when game server is initialized, passed server's object of current session  
35 function init(rtSession) {  
36     session = rtSession;  
37     logger = session.getLogger();  
38 }  
39  
40 // On Process Started is called when the process has begun and we need to perform any  
41 // bootstrapping. This is where the developer should insert any code to prepare  
42 // the process to be able to host a game session, for example load some settings or set state  
43 //  
44 // Return true if the process has been appropriately prepared and it is okay to invoke the  
45 // GameLift ProcessReady() call.  
46 function onProcessStarted(args) {  
47     onProcessStartedCalled = true;  
48 }
```

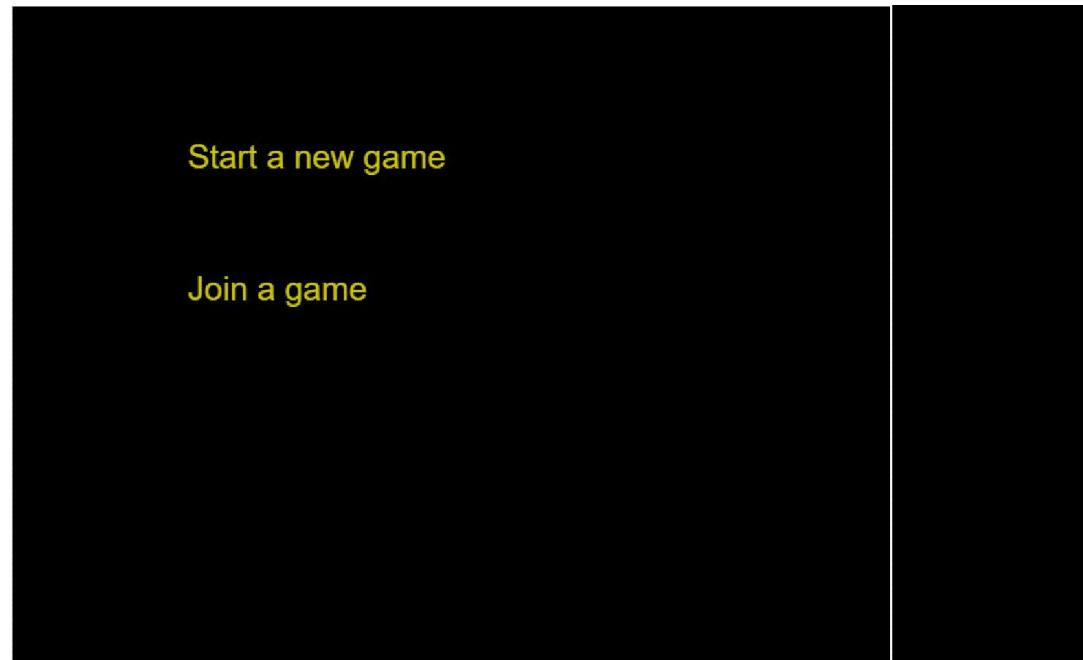
Ln 29, Col 4 Spaces: 2 UTF-8 CRLF JavaScript ✓ ESLint ✓ Prettier Colorize Colorize: 0 variables

# Tutorial: Using the Items and Skills - The Thin Client and UI

- Client side website/UI for users to access the gaming platform.
  - Relies heavily on remote server for Security, Performance, Game rendering
- A thin client is more on the hardware side, such as the Stadia Controller, and is out of scope for this presentation.
  - It is still a vital part for low latency cloud gaming architecture.

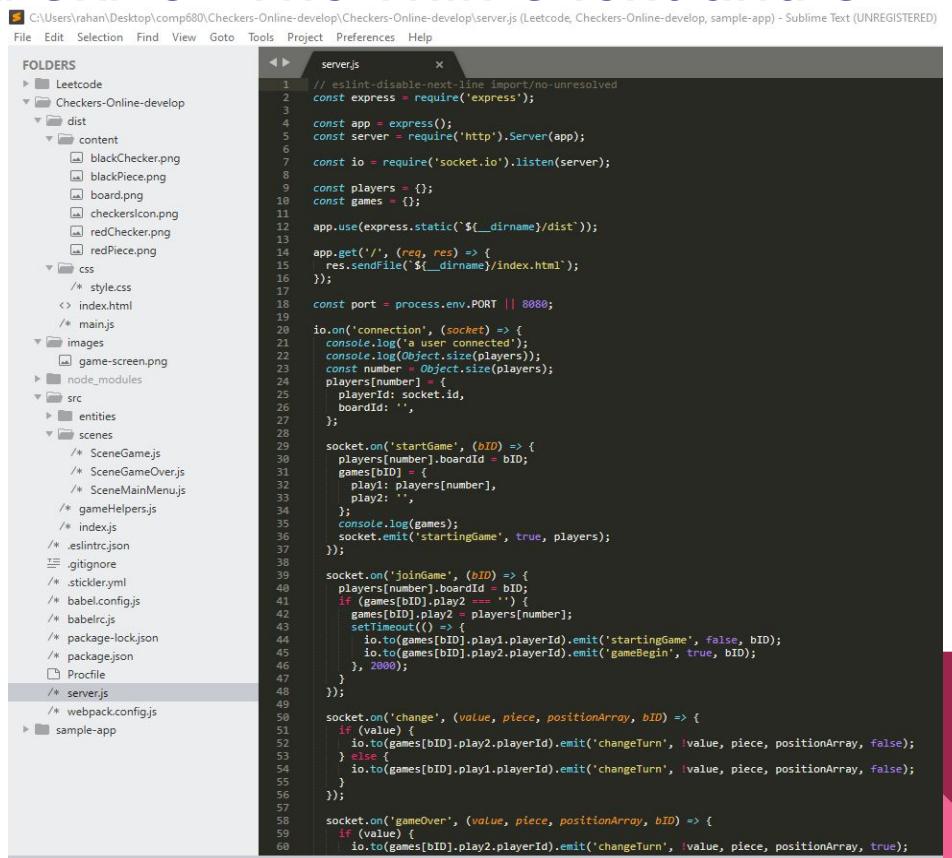
# Tutorial: Using the Items and Skills - The Thin Client and UI

- Allows user to log-in to platform
- Captures user-input when playing the game
- Updates game as per user interaction
- On a stream based platform, streams game onto the user device



# Tutorial: Using the Items and Skills - The Thin Client and UI

- User Authentication
- User-input capturing
- Video Decoding
- Video Streaming



The screenshot shows a Sublime Text window with the following details:

- File Path:** C:\Users\rham\Desktop\comp680\Checkers-Online-develop\Checkers-Online-develop\server.js
- File Type:** Sublime Text (UNREGISTERED)
- Code Language:** JavaScript
- Code Content:** The code is a Node.js application using Express and Socket.io. It handles game logic, player connections, and socket events like 'startGame', 'joinGame', 'change', and 'gameOver'.
- File Structure:** The sidebar shows a file tree with folders for Leetcode, Checkers-Online-develop (containing dist, content, css, images, node\_modules, src, entities, and scenes), and sample-app.

# Tutorial: Using the Items and Skills - Decoding and Encoding

- Video Encoding: Video frames rendered from game engine are compressed.
- Create and configure a channel for each user using an AWS Elemental Live encoder on the ground, with Real-time Transport Protocol (RTP) and forward error correction (FEC), and configure AWS Media Services for processing and packaging in the cloud.
  - Set up **AWS Elemental Live**, an on-premises appliance, as a contribution encoder using RTP with FEC enabled.
  - Configure **AWS Elemental MediaLive** to receive the incoming contribution stream and encode it into a set of adaptive bitrate (ABR) streams.
  - Configure **AWS Elemental MediaPackage** to connect to AWS Elemental MediaLive outputs, for further processing and packaging, to create a channel that can be viewed on connected devices.

# Tutorial: Using the Items and Skills - Decoding and Encoding (Cont.)

## About AWS Elemental products and services

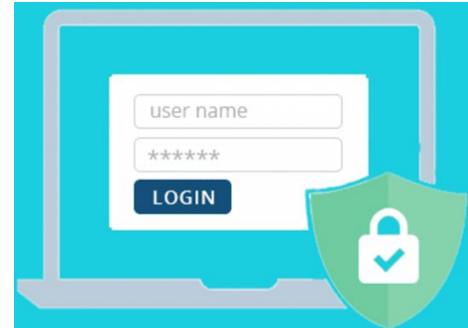
- **AWS Elemental Live** provides real-time video and audio encoding for linear TV broadcast and live streaming to new media platforms. The software-based solution simultaneously processes multiple video outputs, delivering high-quality, high-efficiency performance to format live video for any device.
- **AWS Elemental MediaLive** is a broadcast-grade live video processing service. It lets you create high-quality video streams for delivery to broadcast televisions and internet-connected multiscreen devices, like connected TVs, tablets, smart phones, and set-top boxes.
- **AWS Elemental MediaPackage** reliably prepares and protects your video for delivery over the Internet. From a single video input, AWS Elemental MediaPackage creates video streams formatted to play on connected TVs, mobile phones, computers, tablets, and game consoles.

# Tutorial: Using the Items and Skills - Decoding and Encoding (Cont.)



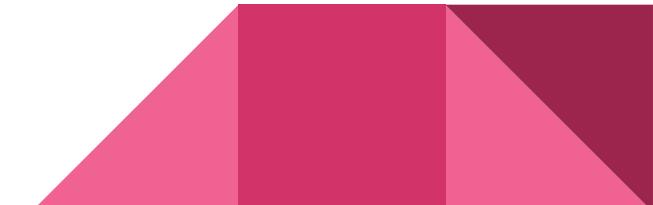
# User Authentication and Authorization

- Authentication is the process of verifying a user's identity. Most commonly, users authenticate with a username and a password.
- Authorization, in contrast, is the process of granting users access to specific resources after they have been authenticated.



# Tutorial: Using the Items and Skills - Authentication: User Side

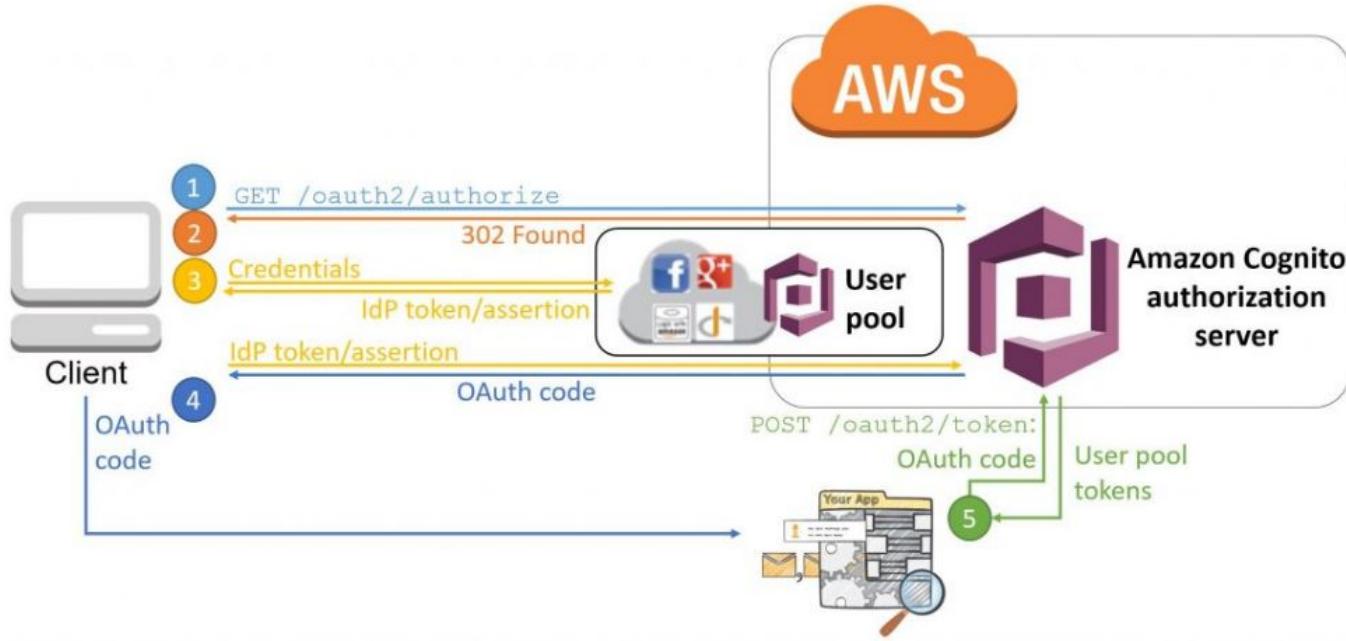
- **Amazon Cognito** allows you to easily integrate with a variety of third-party authentication providers and takes care of integrating with AWS Security Token Service (STS) in order to fetch temporary IAM credentials.
- When using Amazon Cognito, you manage the users for your application using an identity pool.
- These let you define one or more identity providers that will take care of user authentication as well as an IAM role that any authenticated user can assume.



# Amazon Cognito

- User Pools provide a user directory for your application, including user management tools like sign-up, sign-in, group management, etc.
- Identity Pools, in contrast, are used to assign IAM roles to authenticated users.
- OAuth 2.0 to implement single sign-on

# Amazon Cognito: Authorization Code Grant Flow



# The Skill Tree - Game Platform Services

- In order to add any kind of online functionality in-game we need some kind of service.
- Creating a set of services internally can be costly from a development standpoint.
  - Creating the services along with any APIs and/ or SDKs and their documentation is extremely time consuming.
- Using AWS GameLift solves the issue by letting developers use the GameLift SDK in their games.

# The Skill Tree - Game Platform Services (Cont.)

- There are a number of benefits when using the GameLift SDK for both the server (if multiplayer) and the game itself.
  - Supports a number of languages from Javascript to C++ to C#.
  - Development environments supported are the 3 game engines previously mentioned along with any development environments supporting the languages mentioned.
- GameLift also supports the two different operating systems for the game server: Amazon Linux (1 and 2) and Windows Server 2012 R2.
  - Typically we want to right servers on a Linux-based platform to as it is more cost effective in AWS.

# The Skill Tree - Game Platform Services (Cont.)

- To setup the server using the current process of using CI/CD
  - Need to give GameLift access to our scripts in S3 based on our region.
    - We create an IAM role for the AWS service.
  - Need to tell GameLift what bucket its looking for along with the zip file containing the script, the version of our server code, and the ARN value for the IAM role.
  - Need to set up the a client service using AWS Lambda and Amazon Cognito (for user authentication).
    - When creating the policy the needed services under GameLift must be selected to ensure the server can give access to the needed online functionality.
    - A Lambda function with the destined GameLift fleet must be created to listen for user sessions.
  - Keep in mind that the code must reference the GameLift Realtime Client SDK.

# The Skill Tree - Game Platform Services (Cont.)

The screenshot shows the 'Create script' configuration page for Amazon GameLift. At the top, there's a search bar and navigation links for 'Services', 'Search for services, features, marketplace products, and docs [Alt+S]', 'arct117', 'Oregon', and 'Support'. Below the header, the 'Amazon GameLift' and 'Scripts' tabs are selected. The main area is titled 'Create script' with a sub-section 'Script configuration'. It asks for 'Enter metadata for the new script' and provides fields for 'Name\*' (SampleServer) and 'Version\*' (1.00). The 'Script code' section is described as 'Specify how you want to upload your Realtime scripts to Amazon GameLift and provide the required information.' It includes fields for 'Script type' (User storage), 'S3 bucket\*', 'S3 key\*', 'S3 role ARN\*', and 'S3 object version'. A 'Cancel' button is at the bottom left, and a 'Submit' button is at the bottom right.

- Setting up the Scripts needed for GameLift
  - Note that you need to have a role to access the bucket's content.
  - We can also set up versions used.
  - Note this can be done via the AWS CLI for any users with access to the resources.
  - If using a IaC service like Terraform, we need to write policies for buckets and what resources are needed and/ or given to GameLift in the .tf file.

# The Skill Tree - Game Platform Services (Cont.)

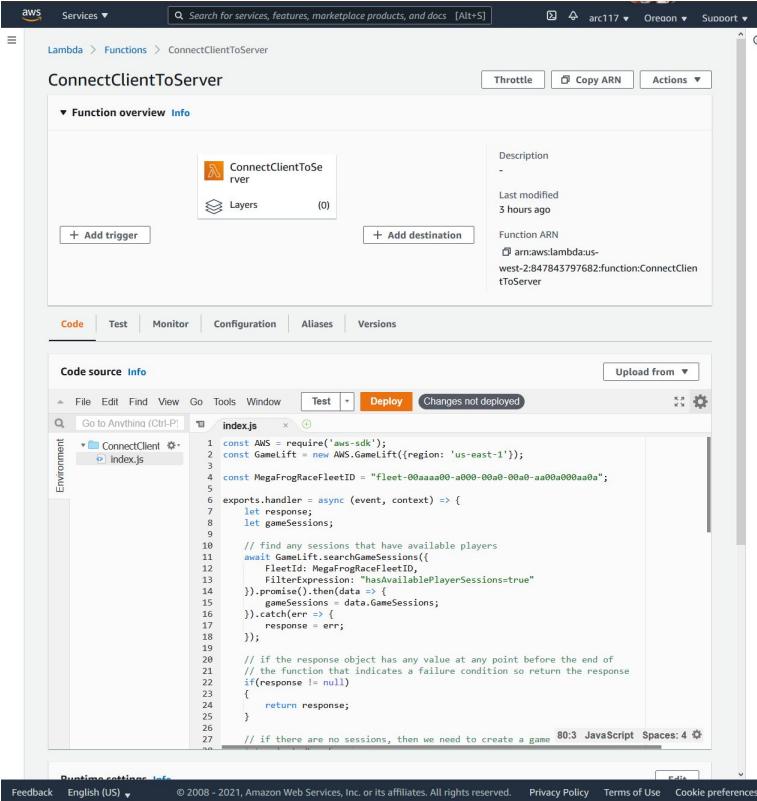
The screenshot shows the AWS GameLift 'Create fleet' interface. At the top, there are fields for 'Name' (Sample Fleet), 'Description' (Sample fleet for the Mega Frog Race Game), 'Fleet type' (On-Demand), 'Metric group' (empty), 'Instance Role ARN' (empty), 'Certificate type' (Disabled), 'Binary type' (Script), and 'Script' (Mega). Below these, a 'Script ID' field contains a long script ID and a 'Script size' of 2.35 KB. Under 'Instance type', a table lists various EC2 instance types grouped by family:

Family	Instance type	vCPU	Memory (GB)	Network performance	Spot history
Compute optimized	c5.large (Free tier)	2	4	Up to 10 Gigabit	Pricing history
Compute optimized	c5.xlarge	4	8	Up to 10 Gigabit	Pricing history
Compute optimized	c5.2xlarge	8	16	Up to 10 Gigabit	Pricing history
Compute optimized	c5.4xlarge	16	32	Up to 10 Gigabit	Pricing history
Compute optimized	c5.8xlarge	36	72	10 Gigabit	Pricing history
Compute optimized	c5.12xlarge	48	96	12 Gigabit	Pricing history
Compute optimized	c5.16xlarge	72	144	25 Gigabit	Pricing history
Compute optimized	c5.24xlarge	96	192	25 Gigabit	Pricing history
Compute optimized	c5a.large	2	4	Up to 10 Gigabit	Pricing history
Compute optimized	c5a.xlarge	4	8	Up to 10 Gigabit	Pricing history

At the bottom left, there's a 'Location management' section with a 'Feedback' link and language settings for English (US). The footer includes copyright information and links for Privacy Policy, Terms of Use, and Cookie preferences.

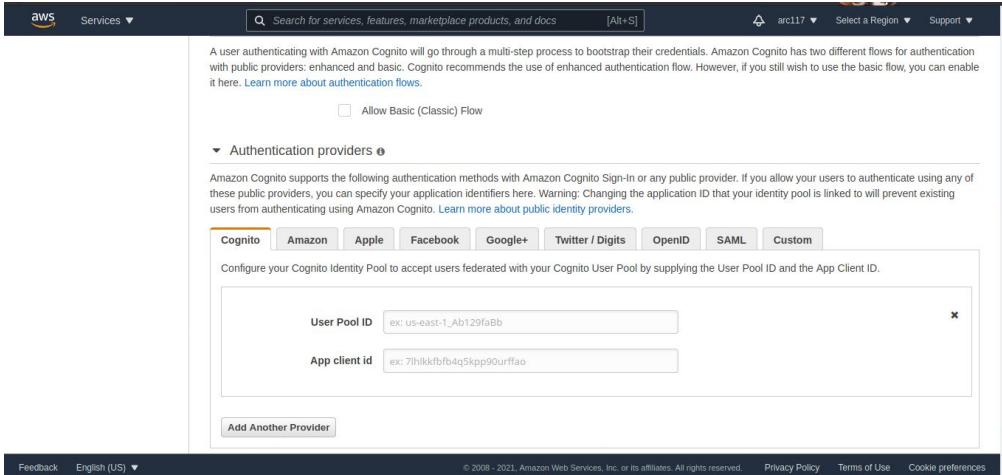
- Fleets can be configured with their instance type, region, and number of server processes in one instance.
  - We can, again, do this via the AWS CLI and in any file containing infrastructure code as previously mentioned.
  - Deciding on regions can depend on a number of factors.
  - Same can be said for deciding on instance type.

# The Skill Tree - Game Platform Services (Cont.)



- Now we create the Lambda to actually handle calls to GameLift server and handle.
- A number of steps must happen:
  - The Lambda must have the proper policy from GameLift.
  - The script for the Lambda must be added via a storage system or manually.
  - Note the scripts is in Javascript and uses the AWS GameLift SDK.

# The Skill Tree - Game Platform Services (Cont.)



A user authenticating with Amazon Cognito will go through a multi-step process to bootstrap their credentials. Amazon Cognito has two different flows for authentication with public providers: enhanced and basic. Cognito recommends the use of enhanced authentication flow. However, if you still wish to use the basic flow, you can enable it here. [Learn more about authentication flows.](#)

Allow Basic (Classic) Flow

▼ Authentication providers ⓘ

Amazon Cognito supports the following authentication methods with Amazon Cognito Sign-In or any public provider. If you allow your users to authenticate using any of these public providers, you can specify your application identifiers here. Warning: Changing the application ID that your identity pool is linked to will prevent existing users from authenticating using Amazon Cognito. [Learn more about public identity providers.](#)

Cognito    Amazon    Apple    Facebook    Google+    Twitter / Digits    OpenID    SAML    Custom

Configure your Cognito Identity Pool to accept users federated with your Cognito User Pool by supplying the User Pool ID and the App Client ID.

User Pool ID: ex: us-east-1\_Ab129faBb

App client id: ex: 7hlkkfb4q5kpp90urffao

Add Another Provider

Feedback    English (US) ▾    © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.    Privacy Policy    Terms of Use    Cookie preferences

- The platform service must also include authentication to ensure users are not randomly accessing other users game data.
  - AWS Cognito will help us set up the allowed users.
  - We will be using Identity Pools since users will be accessing an AWS service.
- We can set up a number of authentication providers and/ or create a custom authentication service.

# The All-Powerful Skill - The Streaming Service

- In order to stream any kind of graphics a GPU is needed.
- Setting up a server farm with GPU can be costly.
  - GPUs randomly fluctuate in price due to crypto currency mining.
  - Server GPUs are a much more costly expense than standard consumer GPUs, ranging in the thousands per GPU.
- An affordable service available is using Cloud Providers GPU instances.

# The All-Powerful Skill - The Streaming Service (Cont.)

	P40	M60	P4	P100	T4	V100	A100	RTX 6000/8000	NGC
Alibaba Cloud			✓	✓		✓	✓		✓
AWS		✓			✓	✓	✓		✓
Baidu Cloud	✓		✓		✓				
Google Cloud			✓	✓	✓	✓	✓		✓
IBM Cloud		✓		✓		✓			
Microsoft Azure	✓	✓		✓	✓	✓	✓		✓
Oracle Cloud				✓		✓	✓		✓
Tencent Cloud	✓		✓				✓		
NPN CSPs		✓		✓	✓	✓		✓	

Nvidia Certified Providers List  
Provided at

<https://www.nvidia.com/en-us/data-center/gpu-cloud-computing/>

# The All-Powerful Skill - The Streaming Service (Cont.)

- How do we decide between all of these providers?
  - Pricing: (We want the best bang for our buck)
    - How much do different GPU services cost per hour used?
  - Compatibility:
    - Does it work with our other services used?
  - Processing Power:
    - What are the limitations?
    - Can we upgrade easily if we find it's not enough?
    - What generations are supported?
    - Does it allow for modern features?
  - Region Support:
    - Is there support in the regions my games will be released in?

# The All-Powerful Skill - The Streaming Service (Cont.)

- We went with an AWS approach for the product.
  - AWS has three different EC2 instances using three different GPU models.
- This guide's approach uses EC2 G4 which utilizes Nvidia Tesla T4 GPUs.
  - T4 models run both CUDA and Tensor cores to speed up calculations and even give the added bonus of ray tracing.
- From a pricing perspective AWS charges \$0.526 per hour on On-Demand Linux instances and \$0.71 for Window Instances for the lowest tier.
- The regions supported total to 24.
- Support for 60 FPS with 4K display output with pre-built images and upto two 4K displays with custom drivers.

# The All-Powerful Skill - The Streaming Service (Cont.)

- Since it is an AWS product integration with other parts of the architecture will be much simpler.
  - Our infrastructure code no longer has to have multiple sets of files for every provider. Instead we can configure for AWS only.
- What about the other two EC2 instances?
  - EC2 P4d using A100 models
    - 4 Regions supported worldwide
    - Single pricing plan of \$32.77 per hour on a Linux instance only.
  - EC2 P3 using V100 models
    - 14 regions supported worldwide
    - Pricing plan of \$3.06 per hour on a Linux and \$3.428 per hour on a Windows instance.

# The All-Powerful Skill - The Streaming Service (Cont.)

Country	Less than 1 hour a week	1-2 hours a week	2-4 hours a week	4-7 hours a week	7-12 hours a week	12-20 hours a week	More than 20 hours a week	Average Hours Each Week
France	14.7%	13.9%	19.7%	20.9%	14.7%	10.5%	5.6%	<b>6.22</b>
Germany	10.8%	16.8%	15.0%	21.0%	16.4%	13.0%	6.8%	<b>6.92</b>
India	12.4%	17.2%	21.2%	16.6%	15.0%	11.6%	6.0%	<b>6.35</b>
Italy	10.0%	18.4%	20.2%	19.8%	16.2%	9.4%	6.0%	<b>6.26</b>
Japan	22.0%	12.6%	15.6%	17.4%	11.6%	8.6%	12.2%	<b>6.64</b>
South Korea	19.8%	20.2%	19.4%	18.6%	10.4%	6.8%	4.8%	<b>5.04</b>
Singapore	15.5%	16.6%	19.8%	19.0%	16.0%	6.1%	7.0%	<b>5.85</b>
United Kingdom	11.4%	17.0%	19.6%	16.0%	16.2%	9.8%	10.0%	<b>6.89</b>
United States	13.6%	15.2%	17.4%	18.2%	16.4%	11.4%	7.8%	<b>6.76</b>
Global	14.5%	<b>16.4%</b>	<b>18.7%</b>	<b>18.6%</b>	<b>14.8%</b>	<b>9.7%</b>	7.4%	<b>6.33</b>

Hours Spent per Week Survey in 2020  
Courtesy of Limelight Network.

# The All-Powerful Skill - The Streaming Service (Cont.)

- To set up the service we need a number of steps:
  - As always we need to be logged into the AWS Console, have the AWS CLI ready, or an IaC file based on an IaC provider.
  - Need to first find out what Amazon Machine Image (AMI) we need.
    - In our case if we want a cheaper instance we can potentially go Linux but if the game needs to be streamed on Windows then a Window Image can be used.
  - Need configure the instance type as the **EC2 G4 Instance** along with its network settings, auto scaling group, storage settings, and security group.
  - Of course you also want to configure a way for the instances to spin up on when a game is requested and make sure authenticated users access the service.
    - This requires more policy and role set up.

# The All-Powerful Skill - The Streaming Service (Cont.)

Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Search for an AMI by entering a search term e.g. "Windows"

Cancel and Exit

1. Choose AMI   2. Choose Instance Type   3. Configure Instance   4. Add Storage   5. Add Tags   6. Configure Security Group   7. Review

Quick Start

My AMIs

AWS Marketplace

Community AMIs

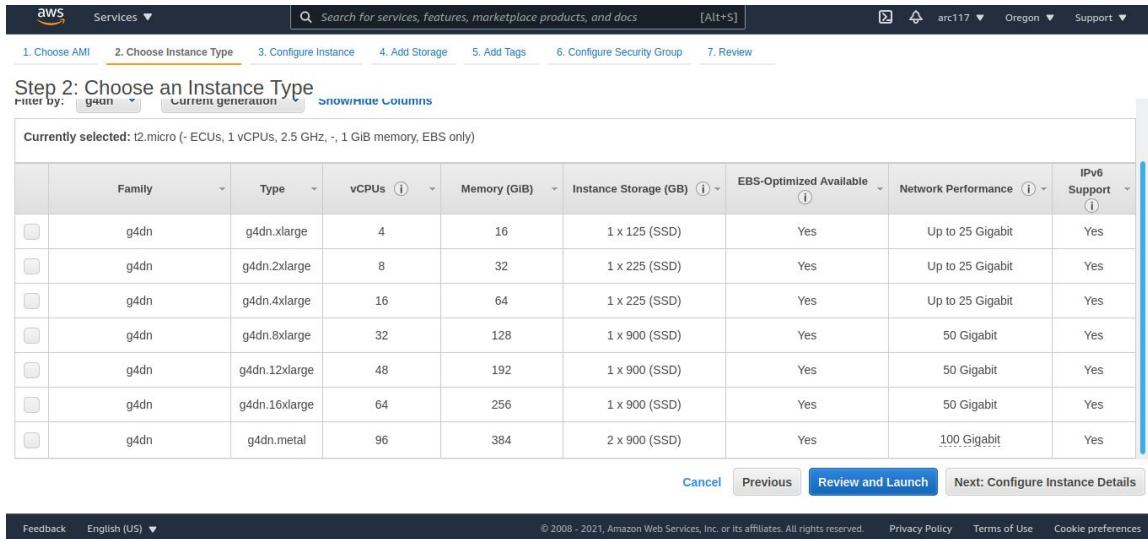
Free tier only ⓘ

Feedback English (US) ▾

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

- Selecting an appropriate image will boil down to the game developers.
  - Some Game Engine frameworks might not be available in certain OS systems.
  - When selecting an image always check if the engine framework supports the OS.

# The All-Powerful Skill - The Streaming Service (Cont.)



The screenshot shows the AWS CloudFormation console with the title "Step 2: Choose an Instance Type". The search bar at the top contains "Search for services, features, marketplace products, and docs [Alt+S]". Below the search bar, there are tabs: 1. Choose AMI, 2. Choose Instance Type (which is selected), 3. Configure Instance, 4. Add Storage, 5. Add Tags, 6. Configure Security Group, and 7. Review. The main content area is titled "Step 2: Choose an Instance Type" and shows a table of instance types. The table has columns: Family, Type, vCPUs, Memory (GiB), Instance Storage (GB), EBS-Optimized Available, Network Performance, and IPv6 Support. The table lists several g4dn instances:

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	g4dn	g4dn.xlarge	4	16	1 x 125 (SSD)	Yes	Up to 25 Gigabit	Yes
<input type="checkbox"/>	g4dn	g4dn.2xlarge	8	32	1 x 225 (SSD)	Yes	Up to 25 Gigabit	Yes
<input type="checkbox"/>	g4dn	g4dn.4xlarge	16	64	1 x 225 (SSD)	Yes	Up to 25 Gigabit	Yes
<input type="checkbox"/>	g4dn	g4dn.8xlarge	32	128	1 x 900 (SSD)	Yes	50 Gigabit	Yes
<input type="checkbox"/>	g4dn	g4dn.12xlarge	48	192	1 x 900 (SSD)	Yes	50 Gigabit	Yes
<input type="checkbox"/>	g4dn	g4dn.16xlarge	64	256	1 x 900 (SSD)	Yes	50 Gigabit	Yes
<input type="checkbox"/>	g4dn	g4dn.metal	96	384	2 x 900 (SSD)	Yes	100 Gigabit	Yes

At the bottom of the table are buttons: Cancel, Previous, Review and Launch (which is highlighted in blue), and Next: Configure Instance Details.

- Selecting an appropriate an appropriate EC2 instance will boil down to the game itself as well.
  - Typically going with a G4 instance as we mentioned with a single GPU and as many vCPUs as needed.
  - Remember vCPU ≠ CPU. One is virtual the other is physical.

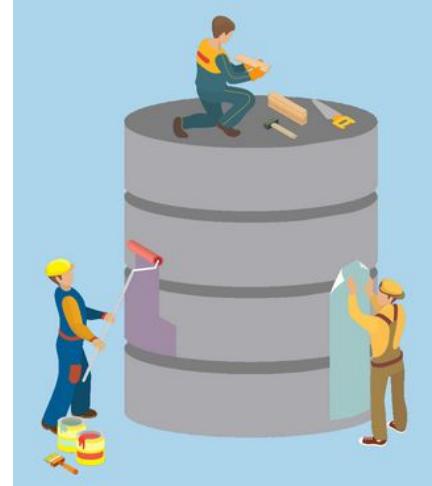
# Checkpoints - Databases

Databases in game development

Differences in databases

Major differences in databases are structure and modeling

Be able to access your data quickly and securely

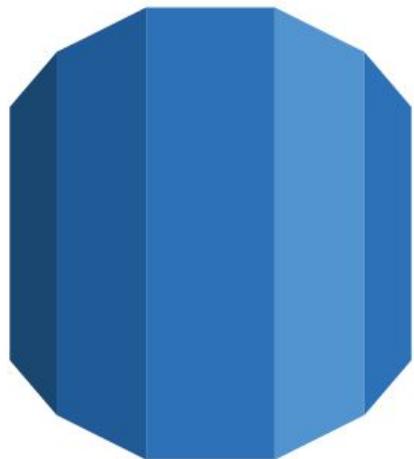


# Checkpoints - Relational Databases

- Oracle is a multi-model database and is commonly used for data warehousing, online transaction processing or mixed workloads
- MySQL is an open-source database widely used for websites, e-commerce, logging applications etc.
- Microsoft SQL Server supports mostly big corporate IT environments
- PostgreSQL is an object-relational open-source database used by many enterprises



- Amazon Aurora is the relational database built for the cloud
- Aurora performs well by having separate data storage and compute layers
- Player and Game Data
  - Inventory
- Keep track of matchmaking rating
  - High Scores



# Amazon Aurora

Screenshot of the AWS Amazon RDS console showing the creation of a new database named "database-1".

The screenshot shows the following details:

- Summary:**
  - DB cluster ID: database-1
  - CPU: -
  - Info: Available
  - Current capacity: 8 capacity units
  - Role: Serverless
  - Current activity: -
  - Engine: Aurora MySQL
  - Region & AZ: us-east-2
- Connectivity & security:**
  - Endpoint & port:
    - Endpoint: database-1.cluster-cngdbo48ivlb.us-east-2.amazonaws.com
    - VPC: vpc-6c67e807
  - Networking:
  - Security:
    - VPC security groups: (empty)
    - Create new DB Subnet Group (sg-01a7a11f2317208)

Navigation and status bar at the bottom:

- Feedback
- English (US) ▾
- © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.
- Privacy Policy
- Terms of Use
- Cookie preferences

# Checkpoints: Non-Relational Databases

- items can be in some sort of object notation like JSON
- MongoDB is the most popular document store database and is also among the top database management systems.
- Apache Cassandra is a free and open-source, distributed, wide-column store, NoSQL database management system



- DynamoDB which is Amazon's footprint in the NoSQL world
- you don't have to manage any of the underlying instances or servers
- pay per request for the data reads and writes that your application performs
- managed and maintained security flaws and patches
- join two different tables together by appending keys into existing items
- sort keys which allow you to quickly sort items

The preview of the new DynamoDB console is now available  
We are redesigning the DynamoDB console. The preview of the new console is a work in progress, but we encourage you to [try it](#) and let us know what you think.

Kinesis Data Streams for DynamoDB is now available  
You now can capture item-level changes in your DynamoDB tables as a Kinesis data stream and start taking advantage of Kinesis services to build advanced streaming applications.

**Create table** **Delete table**

**Player\_stats** **Close**

**Overview** **Items** **Metrics** **Alarms** **Capacity** **Indexes** **Global Tables** **Backups** **Contributor Insights** **Triggers** **More**

**Recent alerts**  
No CloudWatch alarms have been triggered for this table.

**Kinesis data stream details**  
Use Amazon Kinesis Data Streams for DynamoDB to capture item-level changes in your table as a Kinesis data stream. [Learn more](#)

Stream enabled	No
Stream name	-

**DynamoDB stream details**

Stream enabled	No
View type	-
Latest stream ARN	-

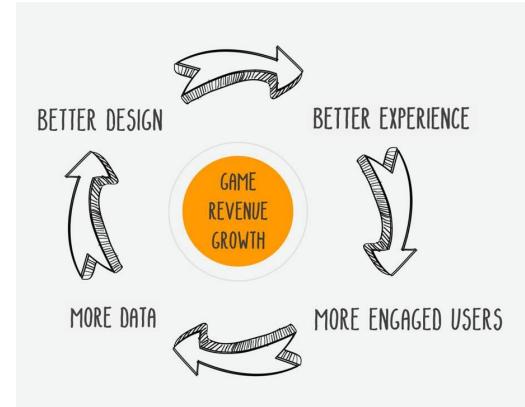
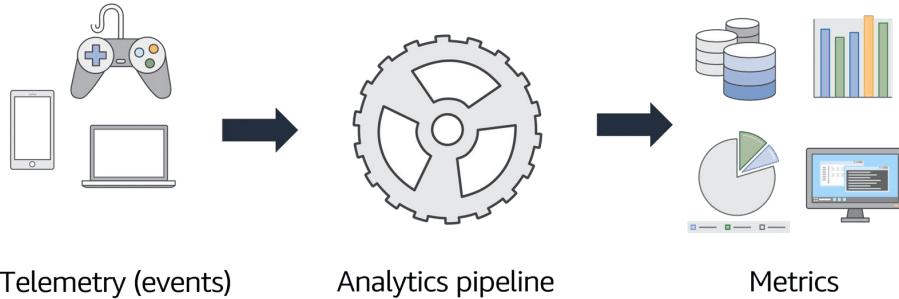
[Manage streaming to Kinesis](#)



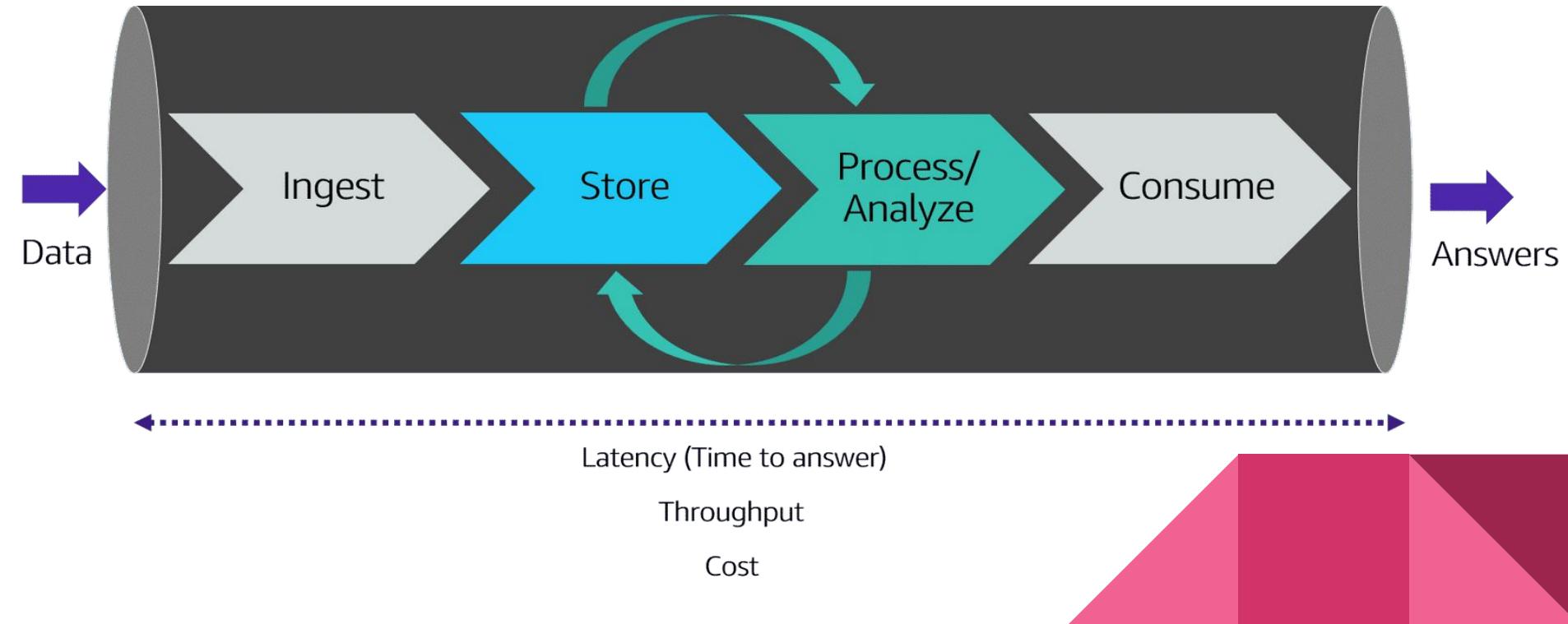
# DynamoDB

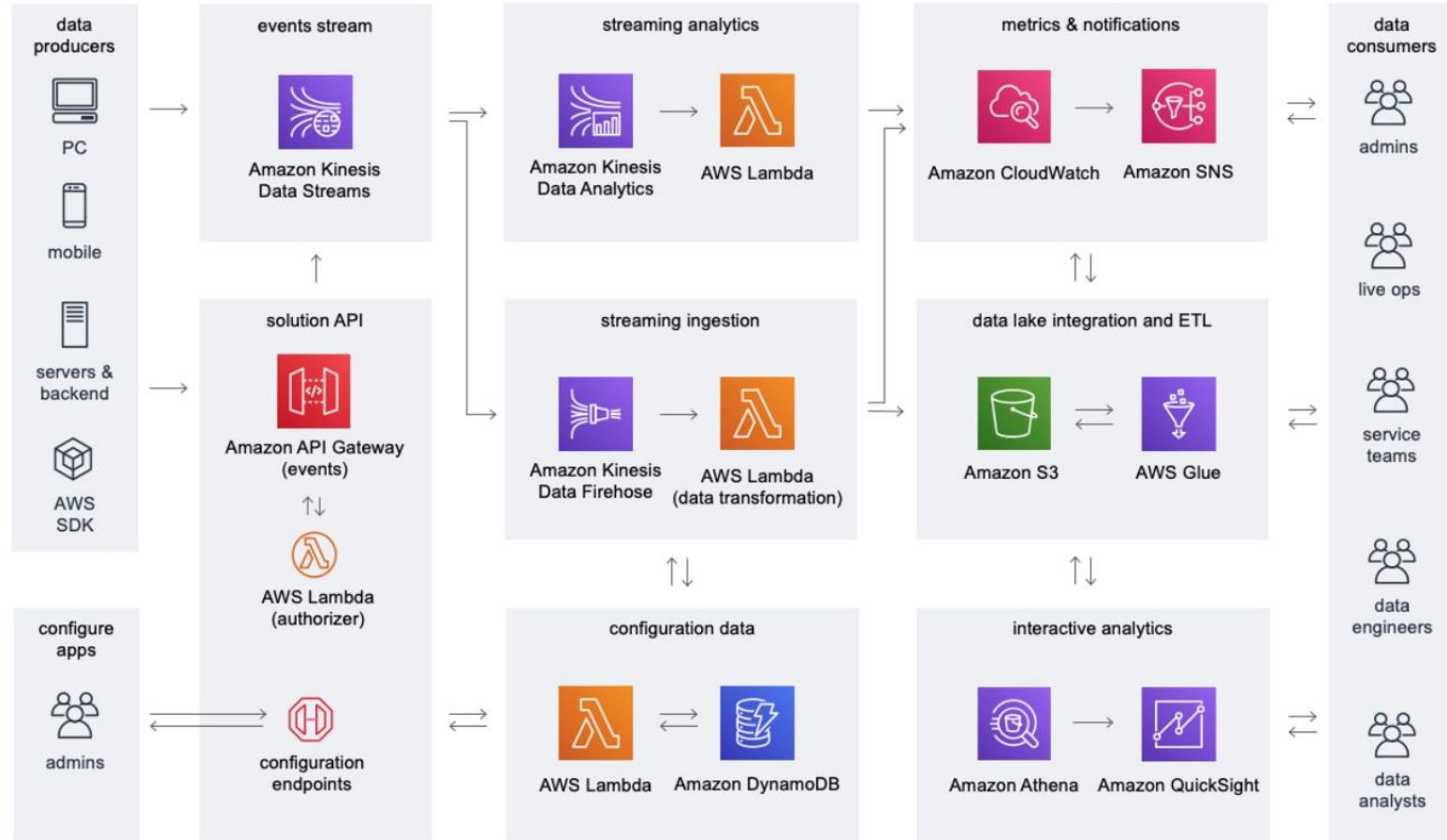
# Health, Mana, and Armor - Analytics of Our Architecture

- Measure and track data elements to monitor performance
- Have the right information at the right time
- Using analytics to improve game design
- Analytics in the Cloud



The image below shows a generalized analytics pipeline





**Figure 1: Game Analytics Pipeline architecture**

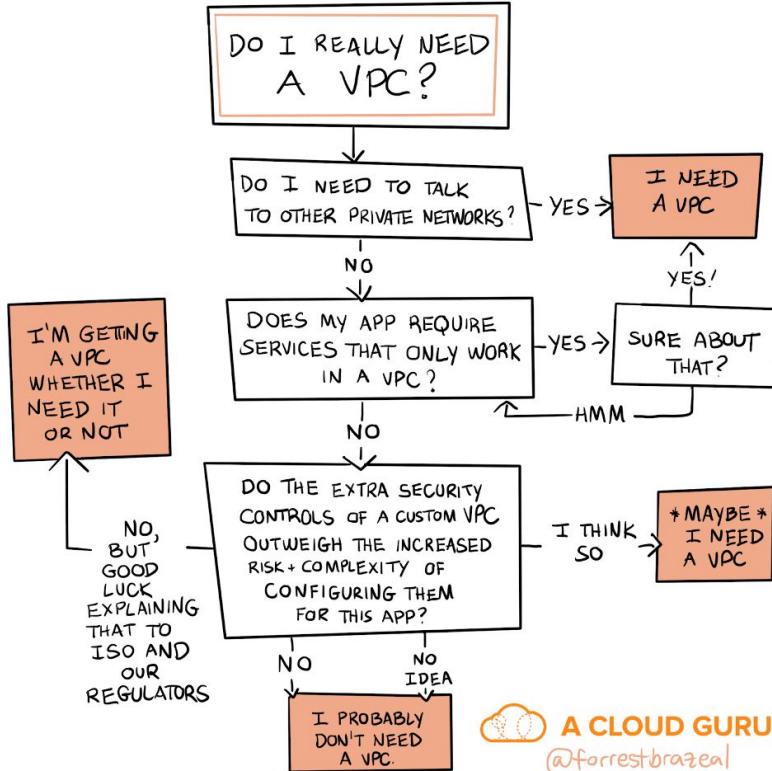
# Fast Travel - Route 53

- What about serving multiple regions?
  - This part can be tricky if a provider does not have some way of routing traffic to the nearest region's-worth of resources.
  - AWS Route 53 can help solve this problem.
    - A Domain Name System (DNS) web service to connect users to desired resources inside or outside of AWS.
  - We want to always have users get as close as possible to a region's resources while also keeping the lowest possible latency.
    - Remember low-latency is key in the cloud gaming world.
    - We don't want a user in Asia to have to wait for an EC2 instance to spin up in a North American region.

## Fast Travel - Route 53 (Cont.)

- We will not be going too much into the specifics of how to set up Route 53 in this seminar as it goes through a similar process as other AWS services with the inclusion of more specific policies dedicated to routing.
  - This must be done for the different regions where the game services are set.
  - Remember just like any other AWS service, Route 53 does not know all of the infrastructure only that its policies tell it who can access the service and what they can do with the service.
- We will bring up that in order to get the nearest region we do need to choose a Routing Policy as Geolocation
  - Keep in mind Route 53 policies can be set based on latency, failure of resources, and geoproximity.

# GameObject.CatchyVPCName.get() - VPC in Our Arch.



- Luckily and unluckily VPCs are tied to EC2 instances so we don't have to worry about wrapping those around a VPC.
- Do we need to wrap everything up in a VPC?
  - Well not necessarily: VPCs are meant to easily access private network resources but if we're already within a VPC for our services we don't need to add additional costly services.



# Comm Links - Communication Between Devs

- Although most organizations have email as a developer it can be nearly impossible to keep track of conversations with other developers.
  - Emails might get lost when there are tens of emails coming in at any given time.
  - Creating custom rules to sort email can take forever.
- A good alternative is to use an application such as Discord.
  - Roles for developers can be set.
  - Bots can be added similar to Slack.
  - Desktop streaming comes with Discord with no price attached to it.
  - Channels for voice and chat can be created.
  - We can connect with fans over Discord as it is a popular application among the gaming community, giving quicker access to feedback.



# Alternate Universes - Other Cloud Providers

- In the implementation we used AWS as the main cloud service provider (not including Game Engines and Repo)
- Like always there are alternatives such as Google Cloud and Heroic Labs.
  - A big part of the decision making process came from the streaming service, the amount of tutorials offered, the game platform service, and pricing.
  - As developers we don't want to choose a service that will make us repeat code or add more work to an already time consuming project.
- What about using Stadia, Luna, or any other Cloud Gaming Service?
  - Unfortunately you need to ask for permissions to obtain SDKs for current cloud gaming services.
  - As time progresses there will be a shift in how gaming is presented, potentially making it easier to develop cloud games.

Questions?

