
2025年度 情報工学実験第2

クラウドコンピューティング実験 概要説明資料

理工学部情報工学科

近藤 正章

cloud_jikken@am.ics.keio.ac.jp

2025年10月6日更新

担当教員の紹介: 近藤正章

• 経歴および専門

- 2021年4月: 慶應大学理工学部情報工学科着任
- (2018~: 理化学研究所計算科学研究センター、チームリーダー兼務)
- 専門: 計算機アーキテクチャ, 高性能計算, AI処理, 量子コンピュータ



コンピュータ アーキテクチャ

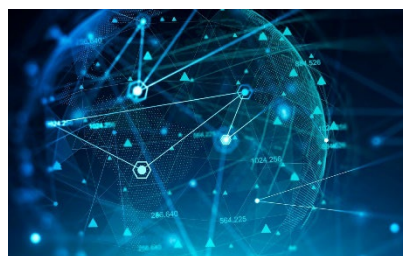


AI処理・ロボット制御向け
計算機アーキテクチャ

3次元積層メモリを用いた
プロセッサシステム

LSI/FPGAでの実装評価

スマート社会向け CPS処理基盤

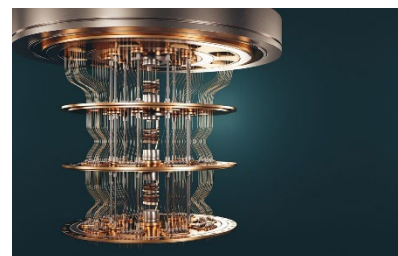


AIによる数理最適化

オンデバイス学習による
異常検知手法

エッジ・クラウド連携システム

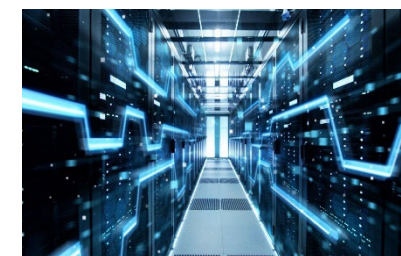
量子 コンピューティング



量子コンピューティングによる
強化学習

ゲート型量子計算機の
エラー訂正機構

次世代スーパー コンピュータ



大規模並列システムの電力
管理とジョブスケジューリング

次世代スーパーコンピュータの
アーキテクチャ探索

TAの紹介

- 安藤 大地さん (ANDO, Daichi)
- 齋藤 楓さん (SAITO, Kaede)
- 高木 響貴さん (TAKAGI, Hibiki)
- 樋渡 雅樹さん (HIWATASHI, Masaki)
- <補助> 杉本 寛直さん (SUGIMOTO, Hirotada)

実験の概要

- 実験の目的

- エッジデバイス等によるデータ収集や入力受付、クラウド環境へのデータ転送、クラウド上でのデータ解析、結果のWEBインターフェース表示、という一連のシステムをエッジデバイス(ラズパイやスマホ、PCなど)とAmazon Web Service(AWS)を用いて構築し、クラウドコンピューティングに関する基礎とそのシステム実装技術を習得する。

- 実験の内容

1. 実験手引書に従って各自でAWSサービスを利用してテストシステムを構築しクラウドコンピューティングを体験・理解する
2. グループ毎にそれぞれのアイデアに基づいて上記の実験目的を達成するシステムを構築する
3. グループ毎に開発したシステムに関する紹介のプレゼン資料を作成し発表する

注意事項

- 実験の形態・場所

- 原則として対面での受講に限る。ただし体調に懸念があるなどの場合はリモート(Zoom)での受講も可能
 - リモート受講の場合は授業開始時にはZoomに接続していること
 - ラズパイ作業を担当する場合はキャンパスでの受講が必須
- 授業時間中は14棟213(セミナールーム3)を占有的に使用可能
- 昼食時・休憩時は貴重品を必ず携行すること

- 質問など

- 教員／TAはセミナールーム3にいたので適宜質問すること
 - 教員／TAも自身の環境で試してみないとわからないことも多いので回答に多少時間がかかる
- Slackで質問してもよい

クラウドコンピューティングとは

クラウドとは

- 自身でリソースを持たずにネットワークから必要なサービスを必要な量だけ利用するコンピューティングの形態
- ネットワークを雲（Cloud）の絵で表すことが多く、繋がったネットワークの先の実態を意識せずにそのサービスやデータを利用することから「クラウド」という言葉が用いられるようになった（所説あり）
- IT機器のネットワーク化が進み、現在はとても重要な技術になっている
- クラウドサービスの例
 - メールサービス（Gmail, Yahoo!メール, ...）
 - オンラインストレージサービス（Dropbox, Google Drive, ...）
 - 仮想マシン（Amazon EC2, さくらサーバ, ...）

クラウドコンピューティングの利点

- コスト的な利点
 - 自身でサーバなどを持つ必要がなくIT機器や設置場所などへの初期投資が必要ない
 - IT機器の維持にかかる経費負担がない
 - 使った分だけの従量課金も可能
- スケーラビリティ
 - 必要になった際にリソースの増強（あるいは必要なくなった際のリソース削減）が容易
- 耐障害性や高いセキュリティ
 - データや機器のバックアップが基本的に自動で行われる
 - （サービスによるが）最新のOS／ソフトウェア環境が提供される
- IT機器のメンテナンスが必要ない
 - 故障対応もクラウド業者が行い、機器も最新に保たれる
- 導入が容易
 - Web上のインタフェースを用いて簡単に高度なシステムを構築することができる

代表的なクラウドサービスプロバイダー

- パブリッククラウド
 - Amazon Web Services (AWS)
 - Google Cloud Platform
 - Microsoft Azure
 - IBM Cloud
 - Alibaba Cloud
- プライベートクラウド
 - Enterprise Cloud (NTTコミュニケーションズ株式会社)
 - KDDIクラウドプラットフォームサービス
 - :

Amazon Web Services (AWS)の成り立ち

- Amazonはもともと書籍のオンライン販売会社
- Amazon社ではITインフラやソフトウェアのメンテナンス高効率化のために様々なウェブサービス(API)を開発
 - 社外にも提供したところ好評化
- 通常サーバはピーク時の処理要求に合わせて導入するがオフピーク時には計算処理能力が相当に余る
 - 処理能力を他者に貸し出したい
- ウェブサービスや計算資源を貸し出すビジネスとしてAWSが誕生
 - ちなみにAmazon社の利益の半分以上はAWSによるもの

クラウドのサービスモデル

- クラウドサービス構築における役割分担による分類
 - IaaS (Infrastructure as a Service) : インフラをサービスとして提供[例: Amazon EC2, Sakura]
 - PaaS (Platform as a Service) : プラットフォームをサービスとして提供[例: Google App Engine]
 - SaaS (Software as a Service) : ソフトウェアをサービスとして提供[例: Gmail, Office 365]



出典: 総務省 ICTスキル総合習得教材, http://www.soumu.go.jp/ict_skill/pdf/ict_skill_2_2.pdf

※本実験では主にPaaSに分類されるサービスを利用してシステムを構築する

実験の進め方

実験スケジュール

第1週目

	実験内容
2限 (10:45-12:15)	・出席確認、実験概要説明 ・グループ分け ・AWSへのログイン確認
(12:15-13:00)	(昼休み)
3限 (13:00-14:30)	・AWS利用のチュートリアル (テストシステム構築)
4限 (14:45-16:15)	・チュートリアルの続き ・開発システムの構想を練る ・グループ内分担分け
5限 (16:30-18:00)	・第1週目の内容が終了していない場合は続ける

第2週目

	実験内容
2限 (10:45-12:15)	・出席確認 ・システム構想の説明 ・担当部のシステム開発
(12:15-13:00)	(昼休み)
3限 (13:00-14:30)	・担当部のシステム開発
4限 (14:45-16:15)	・担当部のシステム開発
5限 (16:30-18:00)	・(必要なら)担当部のシステム開発を続ける

第3週目

	実験内容
2限 (10:45-12:15)	・出席確認 ・グループ毎の統合システム開発
(12:15-13:00)	(昼休み)
3限 (13:00-14:30)	・グループ毎の統合システム開発
4限 (14:45-16:15)	・デモ発表準備
5限 (16:30-18:00)	・デモ発表

- ・ 時間は目安なので、必ずしもこの通りに進める必要はない
- ・ 適宜各自で休憩をとっても良い

実験手引書について

- 実験手引書
 - https://docs.google.com/presentation/d/1CzWLZUWppYzkThPhjf7DFiOM7pOW96uMs94aLF_k93A
 - 閲覧にはkeio.jpアカウントが必要
- 手引書の内容
 - AWSへのログインの方法
 - テストシステム構築のチュートリアル
 - その他AWSサービスの簡単な説明など
- 質問内容などをふまえて常時アップデート予定
 - 常に最新版が参照できるように、ダウンロードせずにWeb上で見ることを勧める

グループ分けについて

- グループ分けを確認(グループはこちらで振り分け)
 - 各グループ4～5人の合計5グループ
 - Pythonのrandomモジュールで初回日の日付(例えば251006)をseedとして指定し乱数で決定
 - グループ作業時にリモート受講の人がいる場合はZoom等で会話をしてください
- 後ほどグループ内で担当を決める
 - エッジデバイス(ラズパイ・スマホなど)側担当: 1～2人
 - クラウド側担当: 残り的人
 - ただし皆でできるだけ協力して実施すること
- クラウド側担当の役割分担の例
 - IoT Coreとその出力周辺担当
 - データ解析・処理部担当
 - Web表示側担当

AWSの利用について

- 本実験ではAWSのパブリッククラウドサービスを利用する
- 大学で契約中のAWS Academy環境を使うことができる
 - keio.jpアドレスでアカウントを作成するので個人でアカウントを作成する必要はない
 - 既に自身のアカウントを持っている人は間違えないように注意すること
(間違えると自分のクレジットカードに請求されてしまいます)
 - 各自50ドル分のクレジット(AWSのサービス利用に使える予算)が利用可能
 - ある程度まで無料で使えるサービスもいくつかある
 - AWS Academyでは権限上使えないサービスを使いたい場合は相談すること
 - **あくまで学習用のアカウントであり公序良俗に反するような使い方は厳禁**
- AWS Academyへの受講生ログイン
 - <https://awsacademy.instructure.com/login/canvas>
 - ログイン情報は@keio.jpアドレスにメールで送信される
 - ログインに成功するとCanvas画面になる(慶應大学のCanvasとは関係ない)
 - 詳細は実験手引書を参照

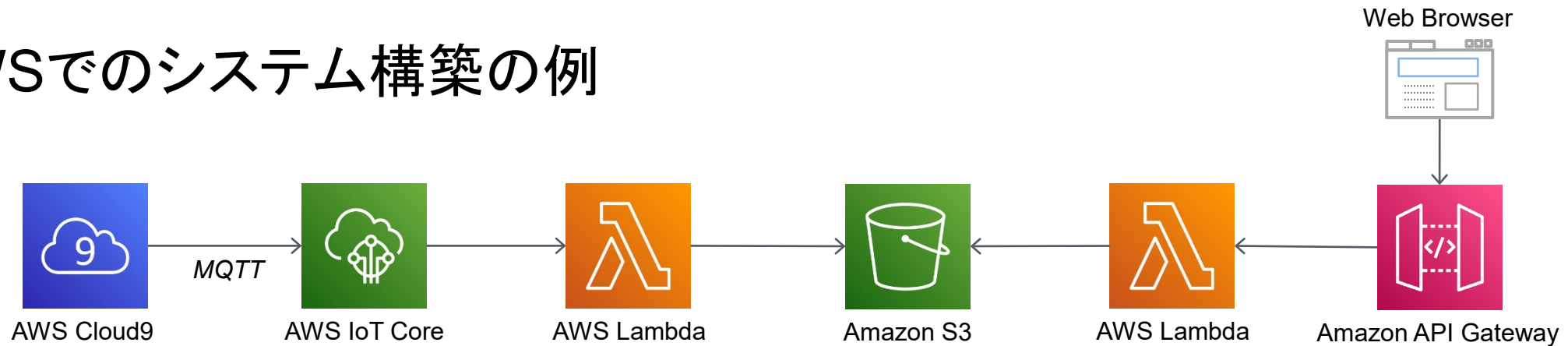
AWS利用に際しての注意点

- 本実験の目的以外にサービスを利用しないこと
- 「クラウド破産」にならないように気を付けること
 - クラウド破産: クラウドサービスの利用料金が意図せずに高額請求されてしまうこと
 - 思わぬ高額な課金が発生してしまう例
 - プログラムのバグ・暴走などで大量のリクエスト／データ送受信を行ってしまう
 - イベントトリガによるサービス起動と当該イベント発生との間に無限ループが生じる
(S3オブジェクト生成でLambda関数起動→その関数内でS3オブジェクトを生成するなど)
 - エンドポイントなどの情報が外部に漏れて第三者に悪用されてしまう
 - 性能保証(I/O性能など)があるサービスをデプロイしてしまう(アクセス等に関係なく課金)
- だからと言って遠慮しすぎ過ぎて十分な実験ができないのもNG
 - 注意はしつつも積極的に様々なことを試してみてください
 - ただしGPUを用いるような機械学習サービスを使うと結構高コスト...
 - もし50ドル分を使い切ってしまった場合は別途アカウントを作成するので申し出ること

クラウドサービスを利用したシステム構築

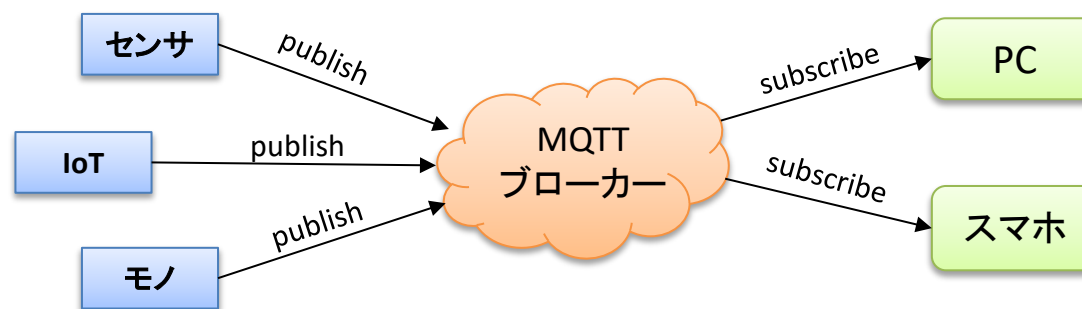
- パブリッククラウドでは様々なサービスが提供されている
 - ストレージ (Amazon S3など)
 - コンピューティング (AWS Lambda、EC2など)
 - データベース (Amazon DynamoDBなど)
 - データ解析・人工知能
 - :

- AWSでのシステム構築の例



技術的な補足

- MQTT (Message Queueing Telemetry Transport) プロトコル
 - TCP/IPで利用できるプロトコルの一種でシンプルかつ軽量 (IoTに適している)
 - パブリッシュ/サブスクライブ型モデル
 - 非同期型の通信のため相手を気にせずに送信・受信が可能

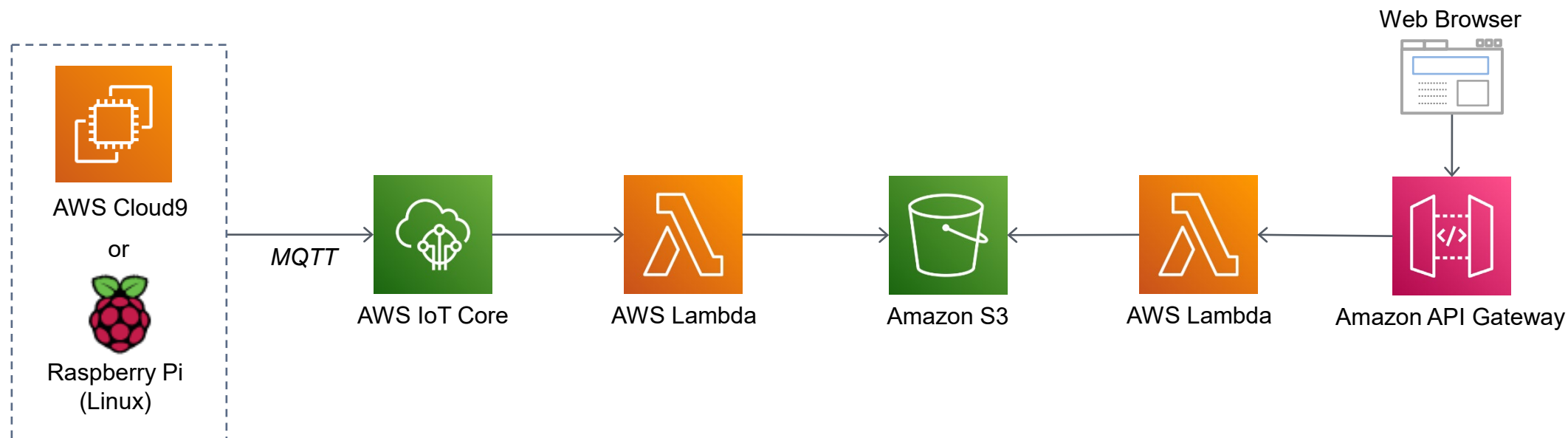


- マイクロサービス
 - 小さな独立した複数のサービスをAPIベース連携させソフトウェアを構成する方式
 - 単位ごとの修正や変更が容易でスケーラビリティに優れる
 - 例えばサービス毎にコンテナを利用してプロセスを起動する
 - 例: AWS Lambda

クラウドコンピューティング実験を始めよう

ステップ1: テストシステムの構築

- 実験手引書に従って各自でテストシステムをAWS上に構築
 - クラウドコンピューティングを体験するためラズパイ担当者も含め全員が行う
 - IoTデバイスからデータを収集し保存＋ウェブに収集データを表示する簡単なシステム
 - 詳細は実験手引書を参照
 - <https://docs.google.com/presentation/d/1Fig9cl9HDeiWt33l7XENx9Hqw7DM-SULu-w015ajvJo>



ステップ2: グループ毎でのより高度なシステム構築

- AWSが提供する様々なサービスやAPIを利用してグループ毎にそれぞれのアイデアに基づいて実験目的を達成するシステムを構築しよう
- 進め方
 - 1週目4限: グループ内でどのようなシステムを作るかを話し合い、決まったらグループ内での担当箇所の分担を行う
 - 2週目2限: 教員／TAに構築予定のシステム概要を資料1枚程度で説明する
 - 前ページのようなアーキテクチャダイアグラムを使って説明すること
 - 作業の分担も示して必要に応じてアドバイスを受けること
 - 2週目2～4限: 担当部分を個別に開発・テストする
 - 3週目2～3限: 個々の担当部分を統合して統合システムを構築する
 - 3週目4限: デモ発表の準備をする
 - 3週目4限: デモ発表

ステップ2: グループ毎でのより高度なシステム構築

- システム構築におけるルール
 - 仮想サーバ(Amazon EC2)は利用しない
 - 基本は\$50のクレジット内で最後のデモ発表まで実施
 - 4人のグループであれば合計200ドル分のクレジットが使えるので通常の利用なら十分
 - それぞれ担当部分を決めて環境構築やデバッグを協力して行う
 - クレジットがどうしても足りなくなった場合は別途アカウントを割り当てるので申し出ること
 - サービス利用において権限上どうしても必要な場合は別途有償アカウントを配布する
 - 最後の統合システムはグループ内の誰かのアカウントで代表して構築する
 - アカウントの貸し借りはしない→自身のパスワードは他の人に教えない
 - (残念ながらAWS Academy受講生アカウントではグループでの共同開発はできない)

ステップ2: グループ毎でのより高度なシステム構築

- グループ内で担当者分けについて
 - エッジデバイス(ラズパイ等)担当: 1~2人
 - クラウド側担当: 残り的人
 - クラウドのどこを担当するかはアイデア次第なので、グループ内で良く相談してください
- ラズパイを使う人へ
 - ラズパイ自体は既に持っている自身のボード、あるいは実験準備室で貸与したものを利用
 - センサキットは貸し出します
 - <https://docs.sunfounder.com/projects/sensorkit-v2-pi/en/latest/>
 - 自宅持ち帰り希望者は機材借用書を提出すること
 - 機材借用書: <https://forms.gle/pYkq9aNkx2aHodEi8>
 - 持ち帰らない場合は毎週返却すること

ステップ2: グループ毎のシステム構築の補足

- ラズパイ作業について
 - ラズパイ用OSのセットアップを自身ですること
 - ラズパイのネットワーク接続は以下を利用すること
 - SSID: jouhoukougakujikken2
 - パスワード: 部屋で掲載
 - センサー部品などの紛失や破損に注意すること
 - カメラやマイク、ICカードリーダーもあるので必要に応じて利用してよい
- クラウドの様々なサービスを積極的に利用してみよう
 - (見た目でインパクトがなくても) 違うサービスを積極的に使ったことも評価対象
 - ネット上には多くのドキュメントがあるので検索してみよう

ステップ3: 統合システムのデモ発表

- 3週目の4限に構築したシステムを紹介するデモ発表の準備をする
 - AWSのアーキテクチャダイアグラムを用いつつシステムの概要を紹介
 - アイコンなどはここからダウンロード可能→ <https://aws.amazon.com/jp/architecture/icons/>
 - 動作の様子を見せる
 - 各担当箇所の工夫点・改善点・**うまくいかなかった点**なども説明する
- 3週目の5限に発表会を開催する
 - Zoom上でプレゼン
 - 各グループ持ち時間10分(時間厳守)
 - デモ発表の中身
 - スライドを用いてプレゼンする(スライドは個別ではなくグループ共通で作る)
→**スライドは後の実験の参考に以下にアップロードをしてください**
https://drive.google.com/drive/folders/1CCRk2un__1TV1nLSYIJ2vQ0f2YF5Qjos
 - 各自の実装箇所は本人が説明する(極簡単でも構わないので全員で協力して発表する)
 - デモとして実際に動作させWebブラウザ上などで表示を見せる

ステップ4: レポート作成

- 表紙などの形式や提出方法の詳細は全体のガイダンス資料を参照
- レポートの内容について
 - 分量の目安: 表紙を含めて4ページ程度
 - 1ページ目: 構築した全体システム概要
 - アーキテクチャダイアグラムと説明 → ここはグループ共通で良い
 - 最終的な表示結果のスクリーンショットなど
 - 2～3ページ目: 担当箇所の説明
 - 担当箇所のAWSサービスの概要や構築したシステムの詳細
 - サービスの設定の仕方やソースコードを含めても良い
 - スクリーンショットなども用いわかりやすく説明すること
- AWS Academyアカウントに関する注意
 - レポート提出 **×** 切以降はアカウントを削除する (次のグループのアカウント作成のため)
- アンケートも協力をお願いします

グループ分けについて

- グループ分け

当日決めます