

01_概要説明と環境構築

AWS(講義名)

今日やること

- シラバス・授業内容の説明
- なぜクラウド(AWS)を使うのか
- 次回以降に向けた環境構築
 - アカウント作成→IaC 環境構築
 - IAMと権限
 - 予算消化アラートの設定
 - S3バケットの作成

概要/アカウント設定

クラウドとは

- オンプレミスとの違い
- メリット
 - 従量課金
 - 伸縮性
 - リソース調達の容易さ
 - 俊敏性向上
 - 世界展開

AWSの特徴

- 伸縮性
 - 需要に応じてサーバ性能や台数を柔軟に変えられる
- 規模の経済
 - 多数のユーザーの使用量集約によるコスト削減
- 俊敏性
 - リソース調達の容易さによるシステム改修・拡大スピードの向上
- グローバルインフラストラクチャ
 - 地理的な制約なくシステムを構築出来る
 - 高速なネットワークにより高いパフォーマンスを発揮

AWSの特徴

- リージョン (Regions)
 - 複数のデータセンターがまとまった世界中にある物理的な場所
- アベイラビリティゾーン (AZ)
 - リージョンの中に存在する1つ以上のデータセンターの集まり
 - リージョンよりも小さな単位
 - それぞれ地理的に隔離
 - 災害やハードウェア故障による障害の影響を最小限に
- リージョン/AZはよく使うので概念として理解しておくといい

責任共有モデル

- クラウド(AWS)における責任共有モデルの基本概念
- AWSの責任範囲
 - ハードウェア、ソフトウェア、ネットワーク
 - 基盤部分
- 利用者の責任範囲
 - AWS基盤上に構成された全ての要素
 - アプリケーション、データ、OS/ミドルウェア設定 (EC2の場合)
 - 暗号化、アクセス管理 (IAM)

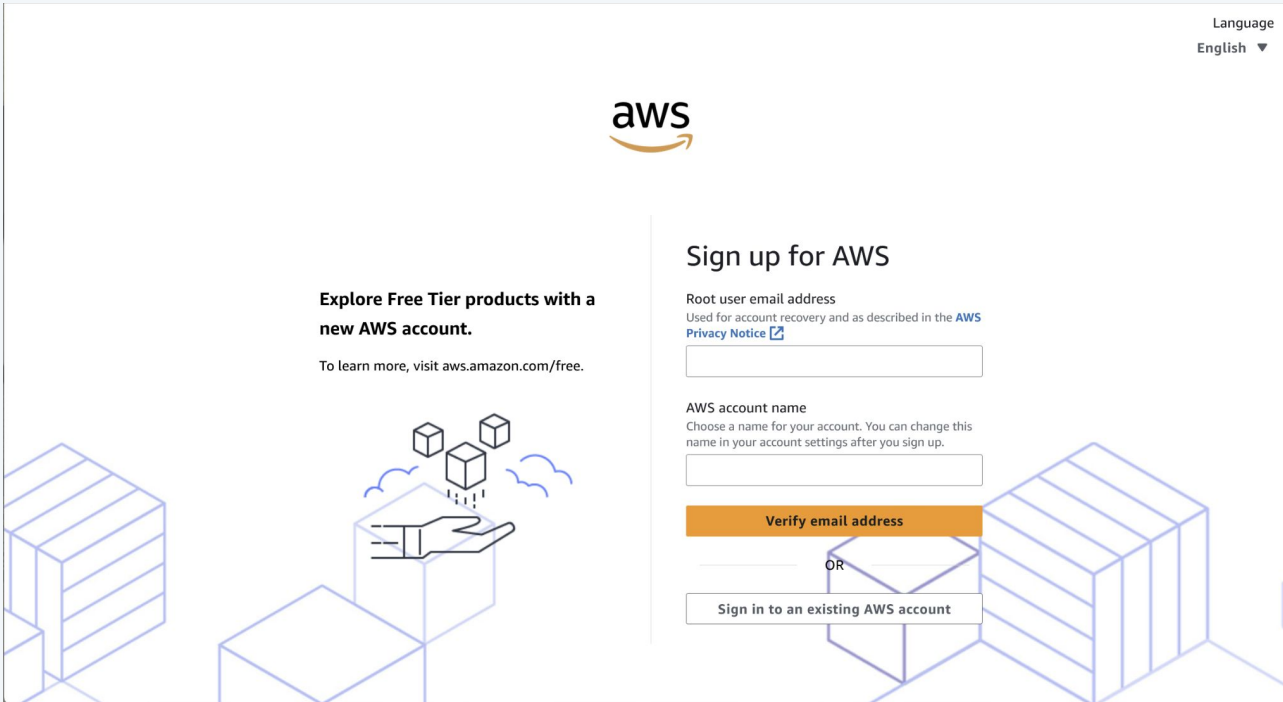
環境構築

アカウント作成

- 新規にアカウントを作成 → [作成ページリンク](#)
 - 過去に授業等でアカウントを作成し、すでに持っている人でも新規に作ることを **推奨**する
 - 後述する Terraform での IaC 環境構築の結果、既存のリソースが消える可能性がある
 - ガイドもあるのでこれをみて進めても良い → [アカウント作成ガイド記事のリンク](#)

アカウント作成

- 左上の *Language* から日本語を選択できる



The screenshot shows the AWS sign-up page. At the top right, there is a 'Language' dropdown menu currently set to 'English'. The AWS logo is centered at the top. On the left, there is a section titled 'Explore Free Tier products with a new AWS account.' with a link to 'aws.amazon.com/free'. Below this is an illustration of a hand placing a cube on a stack of cubes. On the right, the 'Sign up for AWS' section contains two input fields: 'Root user email address' and 'AWS account name'. Below the email field is an orange button labeled 'Verify email address'. Below that is a link 'OR' and a button labeled 'Sign in to an existing AWS account'.

Language
English ▼

aws

Explore Free Tier products with a new AWS account.
To learn more, visit aws.amazon.com/free.

Sign up for AWS

Root user email address
Used for account recovery and as described in the [AWS Privacy Notice](#)

AWS account name
Choose a name for your account. You can change this name in your account settings after you sign up.

Verify email address

OR

Sign in to an existing AWS account

アカウント作成

- ルートユーザーの E メールアドレス
 - github に登録しているメールアドレスと同一のものが良い
 - ここから確認できる→ [リンク](#)
 - 「メールアドレスの形式が無効です。」と言われたら空白文字などが入っていないか確認
- 新しいアカウント名
 - 無難に Yosuke Yamakawa など本名にしておくことをおすすめする
 - honahuku とかネットで使ってるアカウント名でも問題ない気はする(要検証)

アカウント作成

- ルートユーザーパスワード
 - 忘れないようにパスワードマネージャーに保存しておくことを推奨する
 - 山川は 1password を使っているが、ブラウザのパスワードマネージャー等でも今回は問題ない

アカウント作成

- AWS の利用用途を選択ください。
 - ビジネス
- フルネーム
 - 本名。ローマ字推奨
- 組織名
 - KADOKAWAドワンゴ情報工科学院
- 国コード
 - +81
 - 続けて電話番号を入れる
- 国または地域
 - 日本

AWS にサインアップ

連絡先情報

AWS の利用用途を選択ください。

- ☒ ビジネス - 職場、学校、組織向け
☐ 個人 - ご自身のプロジェクト向け

本アカウントの連絡先情報を半角英数字で入力ください。

フルネーム

組織名

国コード

電話番号

 +1 ▼

222-333-4444

国または地域

アメリカ ▼

アカウント作成

- 住所入力
 - アメリカ式の記入順になっている
- 例 (KDG東京校の住所)
 - 住所 1
 - 西五反田2-14-9
 - 住所 2
 - 五反田ケイズビル 5階
 - 市区町村
 - 品川区
 - 州/都道府県または地域
 - Tokyo
 - 郵便番号
 - 141-0031

国または地域
アメリカ ▼

住所 1

住所 2
アパート、棟、ビル、階など

市区町村

州/都道府県または地域

郵便番号

☐ [AWS カスタマーアグリーメント](#) の条項を読み、同意します

同意して次へ (ステップ 2/5)

アカウント作成

- 請求先の国
 - 日本
- カード情報
 - 各自で入力する
 - 基本無料枠で収まる量の使用を想定
 - 数百円/月の請求が飛ぶことが稀にある

AWS にサインアップ

請求情報

請求先の国

請求先の国によって、AWS サービスの支払いに使用できる支払い方法が決まります。

日本 ▼

カード情報



カード番号

有効期限日

MM/YY 形式を使用してください。

セキュリティコード



CVV/CVC

カードの名前

アカウント作成

- SMS がおすすめ
 - 日本(+81)
 - 携帯電話番号は先ほど入力した
ものと同じのもので良い

AWS にサインアップ

本人確認

AWS アカウントを使用する前に、電話番号を認証する必要があります。続行すると、AWS の自動システムから、お客様に認証コードをお知らせします。

認証コードの受け取り方法

- ☒ テキストメッセージ (SMS)
☐ 音声通話

国または地域コード

アメリカ (+1)

携帯電話番号

SMS を送信する (ステップ 4/5)

アカウント作成

- ベーシックサポート - 無料を選択
- サインアップを完了

AWS にサインアップ

サポートプランを選択

ビジネスアカウントまたは個人アカウントのサポートプランを選択します。[プランと料金の例を比較](#)します。プランは、AWS マネジメントコンソールでいつでも変更できます。

☒ ベーシックサポート - 無料

- AWS の使用を開始したばかりの新規ユーザーにお勧め
- AWS リソースへの 24 時間 365 日対応のセルフサービスアクセス
- アカウントと請求の問題のみ
- Personal Health Dashboard と Trusted Advisor へのアクセス



☐ デベロッパーサポート - 29 USD/月から

- AWS を試用するデベロッパーにお勧め
- 営業時間中の AWS サポートへの E メールでのアクセス
- 12 (営業) 時間の応答時間



☐ ビジネスサポート - 100 USD/月から

- AWS での実稼働のワークロードの実行にお勧め
- E メール、電話、チャットによる 24 時間年中無休のテクニカルサポート
- 1 時間の応答時間
- Trusted Advisor のベストプラクティスに関するすべての推奨事項



エンタープライズレベルのサポートが必要です?

1 か月あたり 15,000 USD から、担当のテクニカルアカウントマネージャーが 15 分間対応し、コンシエルジュ形式のサービスをご利用いただけます。[詳細はこちら](#)

サインアップを完了

アカウント作成

- 作成完了 🚀



おめでとうございます。

AWS にサインアップしていただきありがとうございます。

数分でアカウントが有効になります。アカウントが有効になり次第、E メールでお知らせいたします。

AWS マネジメントコンソール にお進みください

[別のアカウントにサインアップするか、お問い合わせください](#)

権限とIAM

IAMについて

- なぜ作業用IAMユーザーが必要か
 - 最小権限の原則
- IAM (Identity and Access Management) とは
 - IAMユーザー
 - IAMグループ
 - IAMポリシー
 - IAMロール

IAMグループの作成

- グループ作成のメリット
 - 複数ユーザーの管理容易化
- ポリシーのアタッチ



ユーザーグループを作成

グループに名前を付ける

ユーザーグループ名

このグループを識別するためのわかりやすい名前を入力します。

最大 128 文字です。英数字と「+','=','@','-」文字を使用してください。

許可ポリシーを添付 - オプション(1/1045) [情報](#)

このユーザーグループに最大 10 個のポリシーをアタッチできます。このグループのすべてのユーザーには、

絞り込み **タイプ**

🔍 検索

すべてのタイプ

[-] | ポリシー名

▲ | タイプ

☒  [AdministratorAccess](#)

AWS 管理 - ジョブ機能

IAMユーザーの作成

- IAMユーザーの作成
- ユーザー名、コンソールアクセス
- パスワード設定
- グループへの追加
- 作成ユーザーでのログイン確認

- ステップ 1
ユーザーの詳細を指定
- ステップ 2
許可を設定
- ステップ 3
確認して作成
- ステップ 4
パスワードを取得

ユーザーの詳細を指定

ユーザーの詳細

ユーザー名

yamakawa

ユーザー名には最大 64 文字を使用できます。有効な文字: A～Z、a～z、0～9、+、=、.、@、_、- (ハイフン)

☒ AWS マネジメントコンソールへのユーザーアクセスを提供する - オプション

人にコンソールアクセスを提供している場合、それは IAM Identity Center でその人のアクセスを管理するための ベストプラクティス になります。

① 人にコンソールアクセスを提供していますか?

ユーザータイプ

☐ Identity Center でユーザーを指定する - 推奨

Identity Center を使用して、人にコンソールアクセスを提供することをお勧めします。Identity Center では、AWS アカウントおよびアプリケーションへのユーザーアクセスを一元管理できます。

☒ IAM ユーザーを作成します

IAM ユーザーを作成するのは、アクセスキー、AWS CodeCommit または Amazon Keyspaces のサービス固有の認証情報、または緊急アクセス用のバックアップ認証情報を使用してプログラムによるアクセスを有効にする必要がある場合のみ推奨されます。

コンソールパスワード

☒ 自動生成されたパスワード

パスワードは、ユーザーを作成した後に表示できます。

☐ カスタムパスワード

ユーザーのカスタムパスワードを入力

• 8 文字以上にしてください

• 大文字 (A～Z)、小文字 (a～z)、数字 (0～9)、!@#\$%^&*()_+- (ハイフン) = [] { } ' といった記号のいずれかの文字タイプの組み合わせ

☐ パスワードを表示

☒ ユーザーは次回のサインイン時に新しいパスワードを作成する必要があります - 推奨

ユーザーは、自分のパスワードの変更を許可する `IAMUserChangePassword` ポリシーを自動的に取得します。

① アクセスキー、または AWS CodeCommit や Amazon Keyspaces のサービス固有の認証情報を使用してプログラムによるアクセス。 [詳細はこちら](#)

ステップ 1

● ユーザーの詳細を指定

ステップ 2

● 許可を設定

ステップ 3

○ 確認して作成

ステップ 4

○ パスワードを取得

許可を設定

既存のグループにユーザーを追加するか、新しいグループを作成します。グループを使用することは、職務機能別にユーザーの許可を管理するためのベストプラクティスの方法です。 [詳細はこちら](#)

許可のオプション

- ☒ **ユーザーをグループに追加**
ユーザーを既存のグループに追加するか、新しいグループを作成します。グループを使用して、職務別にユーザーの許可を管理することをお勧めします。

- ☐ **許可のコピー**
既存のユーザーから、すべてのグループメンバーシップ、アタッチされた管理ポリシー、およびインラインポリシーをコピーします。

- ☐ **ポリシーを直接アタッチする**
ユーザーにマネージドポリシーを直接アタッチします。ベストプラクティスとして、代わりにグループにポリシーをアタッチすることをお勧めします。次に、ユーザーを適切なグループに追加します。

ユーザーグループ (1/1)

[グループを作成](#)

Q 検索

< 1 > ⚙



グループ名



ユーザー



アタッチ

[kdg-aws](#)

0

[Administr](#)

▶ 許可の境界を設定 - オプション

[キャンセル](#)[前へ](#)[次へ](#)

ステップ 1
ユーザーの詳細を指定

ステップ 2
許可を設定

ステップ 3
確認して作成

ステップ 4
パスワードを取得

確認して作成

選択内容を確認します。ユーザーを作成した後、自動生成されたパスワード (有効になっている場合) を表示およびダウンロードできます。

ユーザーの詳細

ユーザー名
yamakawa

コンソールパスワードのタイプ
Autogenerated

パスワードのリセットが必要
はい

許可の概要

< 1 >

名前 	▲ タイプ ▼	次として使用: ▼
IAMUserChangePassword	AWS 管理	許可ポリシー
kdg-aws	グループ	許可グループ

タグ - オプション

タグは AWS リソースに追加できるキーと値のペアで、リソースの特定、整理、検索に役立ちます。このユーザーに関連付けるタグを選択します。

キー

Q Name



値 - オプション

Q yamakawa



削除

新しいタグを追加する

最大 49 個のタグを追加できます。

キャンセル

前へ

ユーザーの作成

CSVファイルをダウンロードする と良い

≡ IAM > ユーザー > ユーザーの作成

✓ ユーザーが正常に作成されました

ユーザーを表示

ユーザーのパスワードと、AWS マネジメントコンソールにサインインするための手順が記載された E メールを表示してダウンロードできます。

ステップ 1

● ユーザーの詳細を指定

ステップ 2

● 許可を設定

ステップ 3

● 確認して作成

ステップ 4

● **パスワードを取得**

パスワードを取得

以下のユーザーのパスワードを表示およびダウンロードするか、AWS マネジメントコンソールにサインインするための手順を E メールでユーザーに送信できます。これは、このパスワードを表示およびダウンロードできる唯一の機会です。

コンソールサインインの詳細

E メールでのサインイン手順

コンソールサインイン URL

https://405539349004.signin.aws.amazon.com/console

ユーザー名

yamakawa

コンソールパスワード

***** 表示

キャンセル

.csv ファイルをダウンロード

ユーザーリストに戻る



パスワードのリセット ⓘ

アカウント (405539349004) のパスワードの有効期限が切れているか、リセットが必要です。

続行するには、古いパスワードを確認し、**yamakawa** の新しいパスワードを設定してください。 (お客様ではありませんか?)

古いパスワード

☐ パスワードを表示

新しいパスワード

新しいパスワードの確認

☐ パスワードを表示

パスワード変更の確認

[別のアカウントにサインインする](#)

Infrastructure as Code

Infrastructure as Code

- IaCとは何か
 - コードでインフラを定義・管理
- IaCのメリット
 - 自動化、再現性、変更管理、ドキュメント化
 - 環境の容易な複製
 - 誤操作の防止
 - レビューのしやすさ

IaC と Terraform

- IaCツールの種類
 - CloudFormation
 - etc..
- Terraformとは
- Terraformでできることの概要


terraform を実行するための リポジトリ作成

次ページに続く

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

 [Single sign-on](#) to see results in the pixiv organization.

Repository template

No template ▾


Start your repository with a template repository's contents.

Owner *

 kdg-server-2025 ▾

Repository name *

aws-honahuku

 aws-honahuku is available.

Great repository names are short and memorable. Need inspiration? How about [automatic-barnacle](#) ?

Description (optional)



Public

Anyone on the internet can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Initialize this repository with:



Add a README file

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: None ▾

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: None ▾

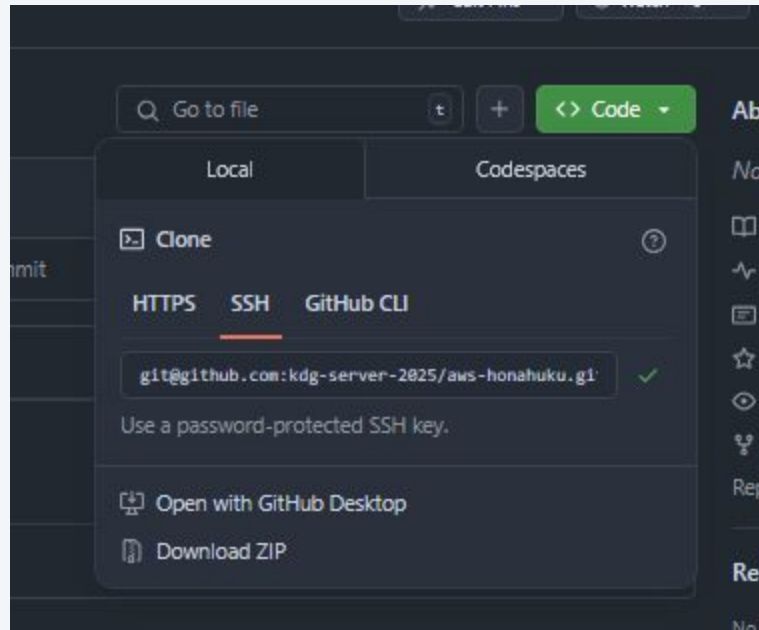
A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

This will set  main as the default branch. Change the default name in kdg-server-2025's [settings](#).

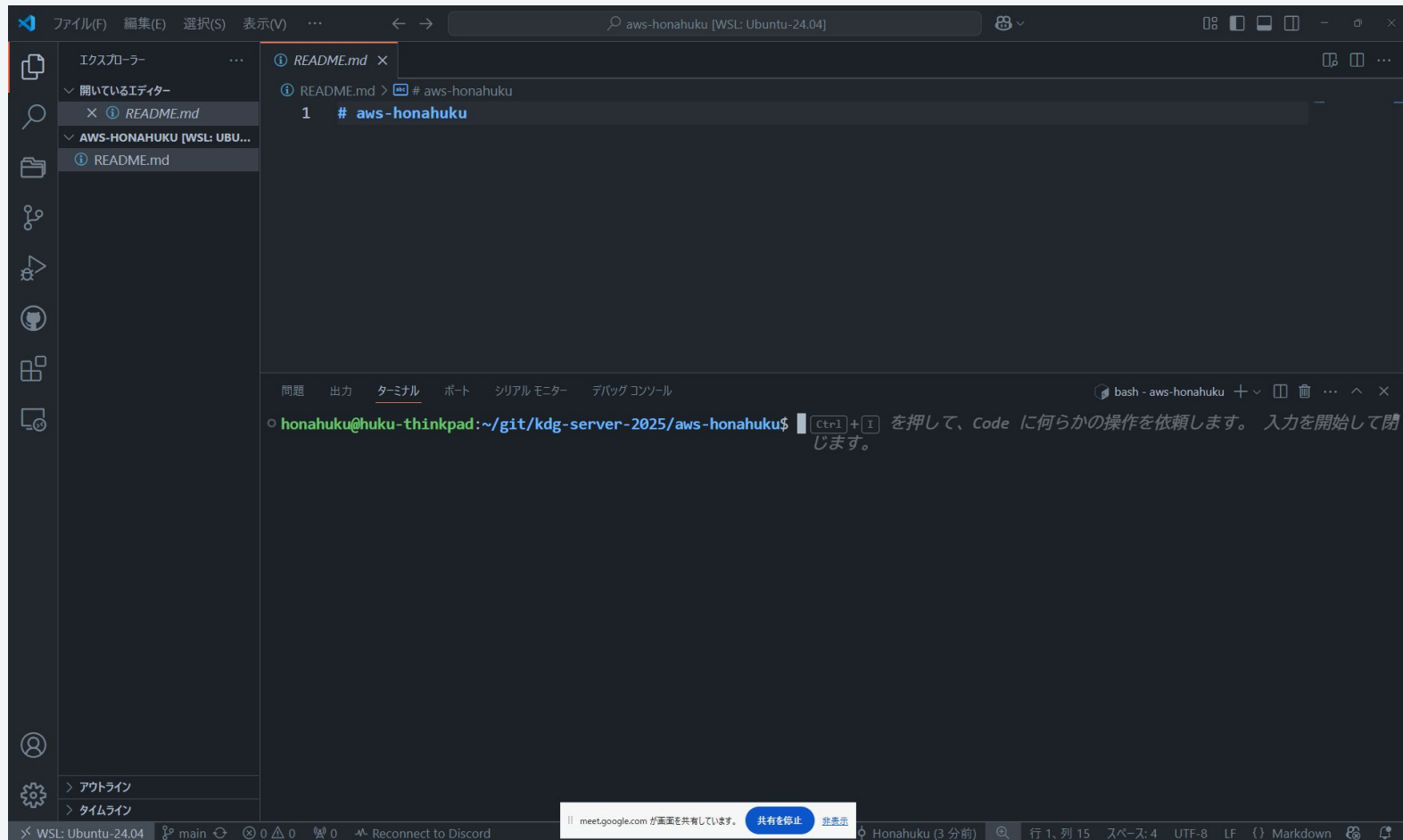
 You are creating a public repository in the kdg-server-2025 organization.

Create repository

リポジトリをclone



vscodeで開く



Terraform環境構築

- Terraformのインストール
- aqua で入れる
 - まだ主流な方法としてはtfenvがある
 - この授業ではまとめてツールを入れたい
 - <https://aquaproj.github.io/docs/install>
 - PATH を通す必要がある
 - <https://aquaproj.github.io/docs/install#2-set-the-environment-variable-path>
 - ~/.zshrc の末尾に↑の export から始まるコマンドを追加する

aqua init

- (terraform を実行するリポジトリに移動しておく)
 - aws-自分のユーザー名 のディレクトリ
- aqua init
- aqua g -i
 - (terraform と type して hashicorp/terraform を選択する)
- aqua i -l
- terraform -v
 - Terraform v1.11.4 みたいに出たら成功

Terraformの認証認可

- AWSとの連携
 - Credentialの設定
- TerraformがAWSリソースを操作するために必要な認証情報
- 認証情報の指定方法
 - 環境変数
 - クレデンシャルファイル
 - IAMロールなど
- 今回は簡単な方法で実施(ファイル作成)

Terraformの基本要素

- HashiCorp Configuration Language (HCL) 構文
- Providerブロック
 - 利用するクラウドプロバイダーの指定
- Resourceブロック
 - 作成するリソースの定義

tfstate について

- Terraformが管理しているリソースの状態を記録するファイル
- 今回はまずローカルにファイルが生成されることを確認する
- 詳細は次週以降で取り扱う

init, plan, apply

- init の役割
 - プロバイダーのダウンロード、作業ディレクトリの初期化
- plan の役割
 - コード実行による変更内容の確認
- apply
 - hoge

预算管理

予算使用アラートを設定する

- AWS Budgets とコスト管理
- AWS利用料金について（従量課金制）
- 意図しない課金のリスクとコスト管理の重要性
- AWS Budgets（予算設定とアラート機能）の紹介
- 今回はこのAWS BudgetsをTerraformで設定する

予算使用アラートを設定する

- Terraformコードによる予算アラート設定
- AWS Budgets定義コードの解説
- `resource "aws_budgets_budget" "..."`
`{ ... }`
- 必要な設定項目
 - 金額、期間、通知先など

S3

S3 の概要

- オブジェクトストレージサービス
- S3バケットとは
 - データの入れ物
- 耐久性・拡張性
- ストレージクラスの種類
 - 標準、IAなど
- 様々な用途

S3バケットの作成

- terraformで使うS3バケットの作成(コンソールで)
- AWSマネジメントコンソールでのS3バケット作成手順
- バケット名の指定
 - グローバルで一意
- リージョンの選択
 - 東京リージョン
- パブリックアクセスの設定
- バケット作成の実行
- 作成されたバケットの確認

名前空間

- linux や k8s, 今回の s3 でも名前空間という概念が出てくる
- その名前が使用できる範囲のこと
- 例) 日本の山田さんと言うと衝突する
 - 品川区五反田hogeのpiyolに住む山田さん、なら一意に定まるかもしれない
 - 「日本の」や「品川区五反田hogeのpiyolに住む」の部分を名前空間と呼ぶ

名前空間

- s3 のリソース名においては全世界の aws で共通の名前空間となっている
- 今回は prefix として kdg-aws-2025- をつけることにする

tfstate

- tfstate の場所と管理
- applyした際にローカルに.tfstateファイルが生成されたことを確認
- tfstateのローカル管理の問題点
 - 共有が難しい、競合リスク
- tfstateをremoteに置くことの概念説明
 - S3バケットなどでのState管理
 - メリット（チーム開発向き、安全性の向上など）

tfstate の移行

- tfstateのS3バケットへの移行
- 作成したS3バケットをTerraformのリモートStateバックエンドとして設定する方法
- コードにバックエンド設定 (`backend "s3" { ... }`) を追加
- `terraform init -migrate-state` コマンドの実行と解説 (ローカルStateをリモートへ移行)
- S3バケットにStateファイルが保存されたことを確認

まとめ/ふりかえり

まとめ(今日の内容)

- シラバス・授業内容の説明
- なぜクラウド(AWS)を使うのか
- 次回以降に向けた環境構築
 - アカウント作成→IaC 環境構築
 - IAMと権限
 - 予算消化アラートの設定
 - S3バケットの作成