
rm rf root - Task Tracker 222

Maintenance Manual

4th December 2022

Amrin Sandhar

Michelle Chan

Brian Frey

Anish Gautam

Visoth Cheam

Noel Ian Paulite

Tobechi Maduchukwu

Marouf Zaman

Table Of Contents

[1.0 Introduction](#)

[2.0 Implementation Tools & Technologies](#)

[2.1 IDE, Programming languages, Libraries, Frameworks](#)

[2.2 Web services](#)

[2.3 Database](#)

[2.4 APIs](#)

[3.0 Runbook](#)

[3.1 Server goes down](#)

[3.2 Database becomes corrupt](#)

[3.3 Third Party services go down](#)

[3.4 How to set up development environment](#)

[3.5 Make changes to code base](#)

[4.0 Deployment](#)

[4.1 Host Machine](#)

[4.2 Amazon Web Services - RDS](#)

[4.2.1 Table Creation](#)

[4.2.2 Set up an RDS instance:](#)

[4.2.2 View database instance information:](#)

[4.2.3 Change the database password:](#)

[4.2.4 Retrieve Cognito Credentials for Environment Variables](#)

[4.3 Amazon Web Services - Cognito](#)

[4.3.1 Set up AWS Cognito](#)

[4.3.1 Retrieve Cognito Credentials for Environment Variables](#)

[5.0 User Interaction](#)

[5.1 Authentication](#)

[5.2 Drill Schedule](#)

[5.3 Drill Management](#)

[5.4 User Management](#)

[5.5 User Registration](#)

[6.0 Database](#)

[6.1 ERDs](#)

[6.2 Other diagrams showing data flow](#)

1.0 Introduction



Task Tracker 222 is an application intended to be used for the purpose of scheduling events for users. As opposed to the document-style emails currently in use, the approach taken by this application allows for easy updates and edits to events, as well as a more readable format.

Events are served to recipients on the basis of whether or not the recipient is included in the participants list. This means that a recipient can be added as a participant not only individually, but also by rank, flight, workcenter, and team. The ability to add both individually and by group allows for a greater degree of flexibility than was currently available with the presently used system.

A dynamic calendar view, with an included intraday summary, allows for a quick “at a glance” visualization for the end user to quickly get a picture of how their schedule will look for any given week.

Management of both users and events was made easier with preset lists of both groups and locations that can be searched and drawn from. All functions required for management can be found on the frontend, requiring minimal technical knowledge for proper usage and operation.

2.0 Implementation Tools & Technologies

2.1 IDE, Programming languages, Libraries, Frameworks

- a. IDE
 - i. IntelliJ IDEA Ultimate - Note: this program is not free. Other sufficiently featured IDEs compatible with Java should work for the purpose of making changes, adjustments, and maintaining the application.
- b. Programming Languages
 - i. Backend
 - 1. Java - The backend of this application is written exclusively in Java.
 - ii. Frontend
 - 1. Javascript - Used heavily for adaptive features on the frontend.
 - 2. HTML - Not a programming language, defines front-end markdown.
 - 3. CSS - Not a programming language, handles front-end styles.
- c. Libraries
 - i. Backend
 - 1. JUnit - Used for unit testing.
 - 2. AWS - Used for all features that interface with AWS services.
 - ii. Frontend
 - 1. JQuery - Used for general frontend DOM manipulation.
 - 2. Chosen - Used for advanced, searchable selection fields.
 - 3. TimePicker - Used for time selection fields.
- d. Frameworks
 - i. Spring Boot - An extension of the Spring Framework, Spring Boot is a comprehensive package that allows for MVC (model, view, controller) development, including an integrated server. The entire backend is structured around the use of Spring Boot's functionality.
 - ii. Maven - Although not strictly a framework, Maven is used for dependency management.

2.2 Web services

- a. Cognito - Cognito is an AWS service used in order to handle user authentication. This takes the burden of security away from the application developer by providing APIs for authentication. Any management necessary that isn't possible from within the application can be done on the Cognito dashboard on AWS.

2.3 Database

- a. AWS RDS PostgreSQL - Deployed and managed through AWS, the database used is a PostgreSQL instance in RDS.

2.4 APIs

- a. No extra web services outside of the two listed in the previous two sections were used.

3.0 Runbook

3.1 Server goes down

- a. Restart the service - The first thing to try is a simple restart of the service through the service on which it is hosted. If this does not succeed, proceed to the next option.
- b. Check the console - If the primary service has ceased to work, check the console for errors. Spring prints detailed errors to the console, as well as stack traces in the event an error has occurred. Identify the error condition and resolve it (for example, hypothetically removing a malformed input from the database, even though this has never caused a crash in our testing). Once the error condition has been cleared, restart the service.

3.2 Database becomes corrupt

- a. For preventative purposes, we suggest setting up regular, automated backups through Amazon's RDS control panel. From this control panel, the database may be reverted to a previous backup in the event they become corrupt.
- b. For further reading, we suggest referencing Amazon's guide on [Backing up and restoring an Amazon RDS DB instance](#). *We feel it would be a disservice to describe the steps in this document, as these interfaces and the relevant information are subject to regular change, and thus, problems relating to them should be resolved by referencing the official documentation that is kept up to date.*

3.3 Third Party services go down

- a. Database
 - i. If the database service (RDS) is not working, verify that your environment variables are correct. For further reading, please reference section [**4.1d**](#).
 - ii. For further reading, we suggest referencing the official [Amazon Relational Database Service \(Amazon RDS\) User Guide](#).
- b. Cognito
 - i. If the authentication service (Cognito) is not working, verify that your environment variables are correct. For further reading, please reference section [**4.1d**](#).
 - ii. Cognito can be managed through the dashboard on AWS. In this control panel, error statuses can be found and used to resolve the issue causing the outage.
 - iii. For further reading, we suggest referencing Amazon's guide on [Troubleshooting Amazon Cognito identity and access](#) and [Managing error responses](#) for Cognito. Just as with the RDS documentation, we feel it would be a disservice to detail the steps here, as they are subject to change.

3.4 How to set up development environment

- a. An original copy of the source code will be provided to the organization. It is suggested that this copy is committed and pushed to GitHub under the organization's collection of repositories. The rest of this section is under the assumption this step has been completed.
- b. Clone a copy of the repository to your local machine. This can be done with the command **git clone <insert repository link here>** using the git bash command line.
- c. Open the source directory of the project in your IDE of choice. Jetbrains IntelliJ IDEA is recommended.
- d. This project is very lean on outside dependencies. As such, the aforementioned steps should be sufficient to begin making changes to the code base.

3.5 Make changes to code base

- a. The use of git is encouraged when making changes to the code base. The typical hierarchy of git branches used in this project were **main ← dev ← <all feature branches>**. It is recommended to first ensure that **dev** is up to date with **master**, then branch off of **dev** using the command **git checkout -b <branch name here>**.
- b. To further elaborate upon hierarchy, **master** should be used for a stable release of the product, whereas **dev** should be the experimental branch that contains recent changes.
- c. If an uncommitted change needs to be rolled back, use the command **git reset <path to file>**.
- d. If it is desired to revert to a previous commit, ensure that the working tree is clean with the command **git status**. If the tree is clean, use the command **git revert <desired commit hash>**.
- e. To commit changes, first add the files that are desired to commit by using either **git add <path to file>**, if adding individual files, or **git add -A**, if adding all files. Commit with **git commit -m "<commit message here>"**, using a descriptive commit message. Push changes to GitHub if desired with the command **git push origin <branch name>**.
- f. The ideal way to merge changes back to the primary branches is through the use of **pull requests** on GitHub. Before doing this process, ensure that the desired branch is pushed to the remote repository. When ready, navigate to the repository on GitHub and proceed to the next step.
- g. Navigate to the **branches** menu and click the **New Pull Request** button. Set your **base** branch as the branch that will receive the changes (usually **main** or **dev**), then set your **compare** branch to the branch that holds the new features. Write a comment if desired, then click **Create pull request**.

-
- h. Navigate to the **Pull requests** tab. Select the desired pull request. Resolve any conflicts, if present. Once conflicts are resolved (if any), **merge** your changes. With this step, merging is complete!

4.0 Deployment

4.1 Host Machine

- a. Note - Sections **4.2** and **4.3** should be completed before following the instructions in **4.1**.
- b. The client will be provided with a final version of the single-source executable in the form of a **JAR file**. In the event the client has made changes to the codebase and needs to build a new executable, navigate to the source directory of the project and use the command **mvn clean install**. Once the build is complete, navigate to the **/target** directory of the project and retrieve your newly-minted jar file.
- c. The aforementioned jar file should be placed in the directory from which its execution will occur. Ensure at least **Java 11** is present on the system hosting the application. Execute the jar file with the command **java -jar <name of jar file>**.
- d. Upon the first execution of the application, a **/config** directory will be made within the same directory as the jar file. Within this directory, there are five files detailing the different ranks, flights, workcenters, teams, and locations, as specified by the client at the time of development. If desired, any of these files may be modified as the user sees fit - they are merely line-separated text files that serve as unordered lists.
- e. For the application to work properly, the following **environment variables** must be set:
 - i. **AWS_REGION** : The AWS region used by your services
 1. This will likely take the value of **us-west-1**
 2. See section **4.3.1e**
 - ii. **USER_POOL_ID** : The ID of the user pool as defined by your instance of Cognito.
 1. See section **4.3.1e**
 - iii. **ACCESS_KEY** : The access key associated with your instance of Cognito.
 1. See section **4.3.1g**
 - iv. **SECRET_KEY** : The secret key associated with your instance of Cognito.
 1. See section **4.3.1h**
 - v. **CLIENT_ID** : The client ID associated with your instance of Cognito.
 1. See section **4.3.1f**
 - vi. **CLIENT_SECRET** : The client secret associated with your instance of Cognito.
 1. See section **4.3.1f**
 - vii. **COGNITO_CLIENT_NAME** : The client name for your user pool in Cognito.
 1. See section **4.3.1f**
 - viii. **COGNITO_ISSUER_URI** : The issuer uri associated with your instance of Cognito.
 1. See section **4.3.1i**
 - ix. **DATABASE_URL** : The url of the database as defined by your RDS instance.
 1. See section **4.2.4a**

-
- x. **DATABASE_USERNAME** : The database username in RDS.
 - 1. See section **4.2.4b**
 - xi. **DATABASE_PASSWORD** : The database password in RDS.
 - 1. See section **4.2.4c**
 - xii. **SUBDOMAIN** : The subdomain of your services.
 - 1. See section **4.2.4j**

4.2 Amazon Web Services - RDS

4.2.1 Table Creation

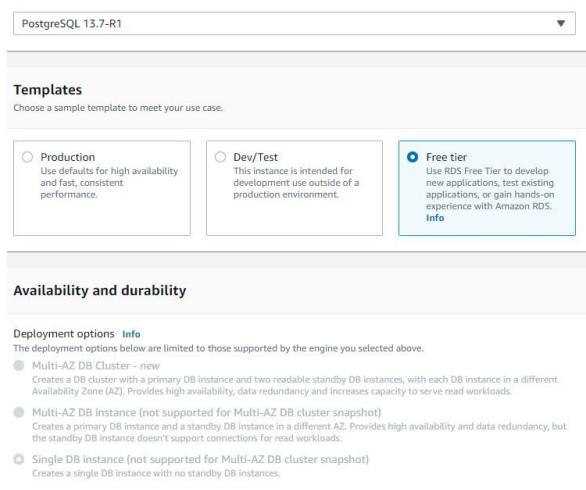
- a. See section 6.1 in this document for scripts used to create each table (Drill and User)

4.2.2 Set up an RDS instance:

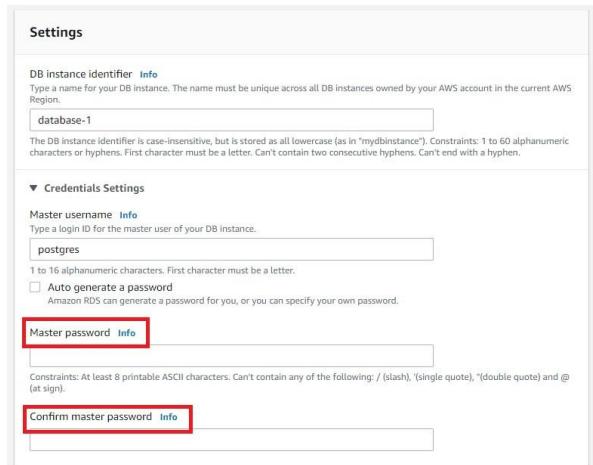
- a. Click **Create Database**



- b. Click **Free Tier**



- c. Enter the **master password** for the database instance



d. **Fill in your information** according to the following images:

Instance configuration

The DB instance configuration options below are limited to those supported by the engine that you selected above

DB instance class [Info](#)

- Standard classes (includes m classes)
- Memory optimized classes (includes r and x classes)
- Burstable classes (includes t classes)

db.m6i.large

2 vCPUs 8 GiB RAM Network: 10,000 Mbps

- Include previous generation classes

Storage

Storage type [Info](#)

General Purpose SSD (gp2)

Baseline performance determined by volume size

Allocated storage

200

GiB

The minimum value is 20 GiB and the maximum value is 65,536 GiB

Storage autoscaling [Info](#)

Provides dynamic scaling support for your database's storage based on your application's needs.

- Enable storage autoscaling

Enabling this feature will allow the storage to increase after the specified threshold is exceeded.

Maximum storage threshold [Info](#)

Charges will apply when your database autoscales to the specified threshold

1000

GiB

The minimum value is 220 GiB and the maximum value is 65,536 GiB

Choose existing
Choose existing VPC security groups

Create new
Create new VPC security group

Existing VPC security groups

default X

Availability Zone [Info](#)

No preference

RDS Proxy

RDS Proxy is a fully managed, highly available database proxy that improves application scalability, resiliency, and security.

[Create an RDS Proxy](#) [Info](#)
RDS automatically creates an IAM role and a Secrets Manager secret for the proxy. RDS Proxy has additional costs. For more information, see [Amazon RDS Proxy pricing](#)

► Additional configuration

Database authentication

Database authentication options [Info](#)

Password authentication
Authenticates using database passwords.

Password and IAM database authentication
Authenticates using the database password and user credentials through AWS IAM users and roles.

Password and Kerberos authentication
Choose a directory in which you want to allow authorized users to authenticate with this DB instance using Kerberos Authentication.

Monitoring

Performance Insights [Info](#)

Turn on Performance Insights [Info](#)

Retention period [Info](#)

7 days (free tier)

AWS KMS key [Info](#)

(default) aws/rds

Account
064743136929

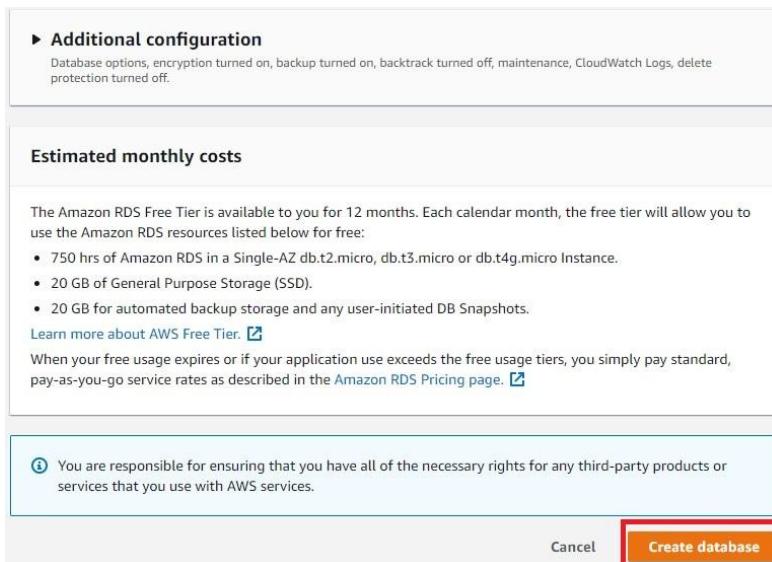
KMS key ID
alias/aws/rds

⚠ You can't change the KMS key after enabling Performance Insights.

► Additional configuration

Enhanced Monitoring

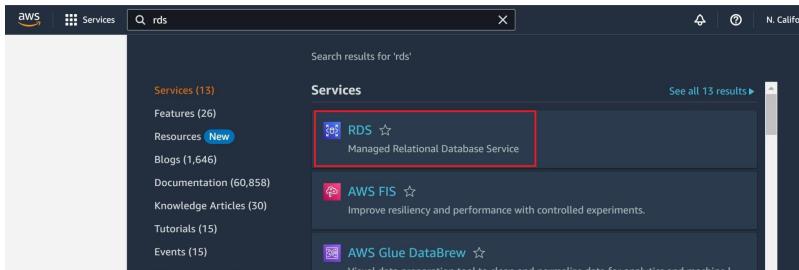
e. Click **Create Database**



f. Database is **created**.

4.2.2 View database instance information:

a. Log into the AWS console, click **search**, and search **RDS**



b. To view the database instance information, first click **database**, then click **database-1**

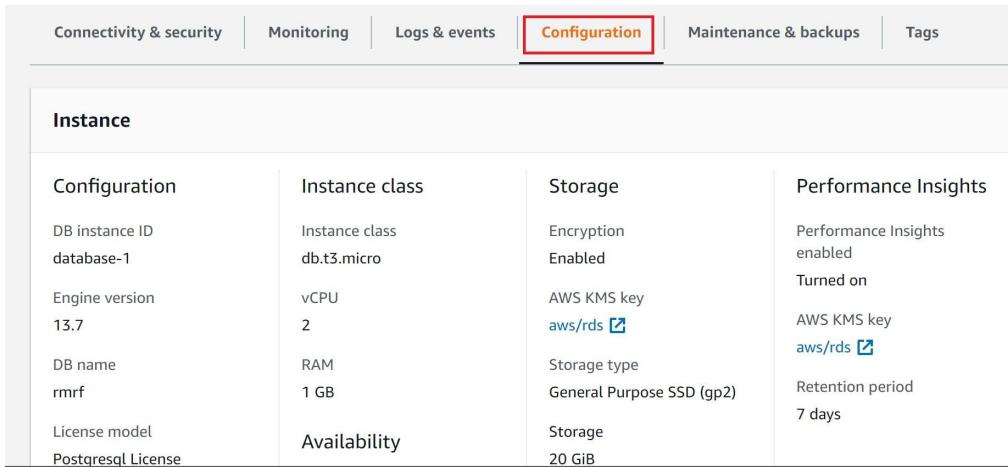
Databases 1

DB identifier	Instance	Engine	Region & AZ	Size
database-1	PostgreSQL	us-west-1c	db.t3.micro	

Connectivity & security

Endpoint & port	Networking	Security
Endpoint database-1.cuqrkcpdpixe.us-west-1.rds.amazonaws.com	Availability Zone us-west-1c	VPC security groups default (sg-00650820dc7775d5c) Active
Port 5432	VPC vpc-03184f780e300aa7d	Publicly accessible Yes
	Subnet group default-vpc-03184f780e300aa7d	Certificate authority rds-ca-2019
	Subnets subnet-09e521a9e6634b82e subnet-0c2c1d85954d05912	Certificate authority date August 22, 2024, 10:08 (UTC-07:00)

c. To view the database name, click **Configuration**

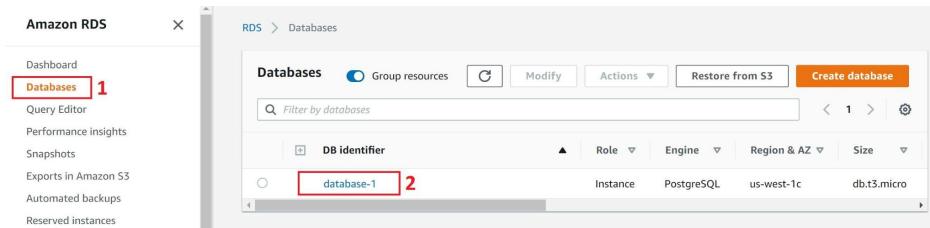


The screenshot shows the 'Configuration' tab selected in the top navigation bar. Below it, the 'Instance' section displays various database parameters:

Configuration	Instance class	Storage	Performance Insights
DB instance ID database-1	Instance class db.t3.micro	Encryption Enabled	Performance Insights enabled Turned on
Engine version 13.7	vCPU 2	AWS KMS key aws/rds	AWS KMS key aws/rds
DB name rmrf	RAM 1 GB	Storage type General Purpose SSD (gp2)	Retention period 7 days
License model Postgresql License	Availability	Storage 20 GiB	

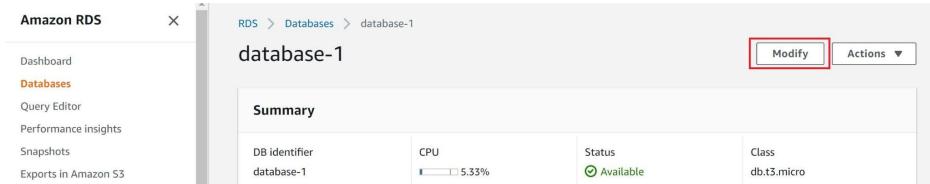
4.2.3 Change the database password:

a. Click **database**, then click **database-1**



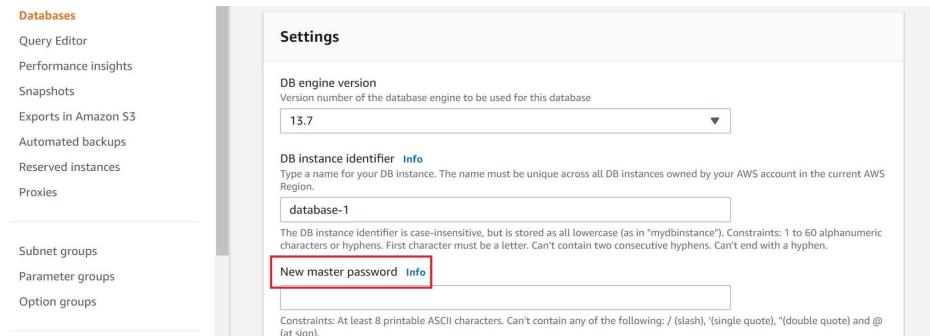
The screenshot shows the 'Databases' section of the AWS RDS console. A red box labeled '1' highlights the 'Databases' link in the left sidebar. Another red box labeled '2' highlights the 'database-1' entry in the list.

b. Click **modify**



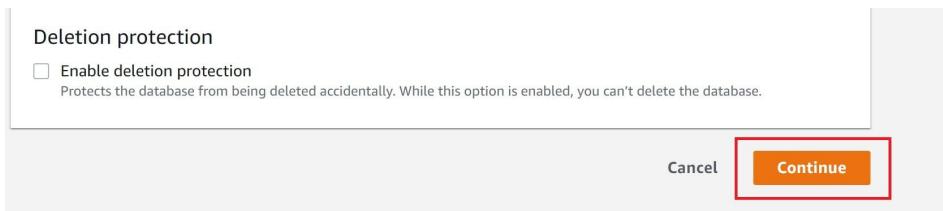
The screenshot shows the 'database-1' modification page. A red box labeled 'Modify' is highlighted in the top right corner.

c. Fill new master password



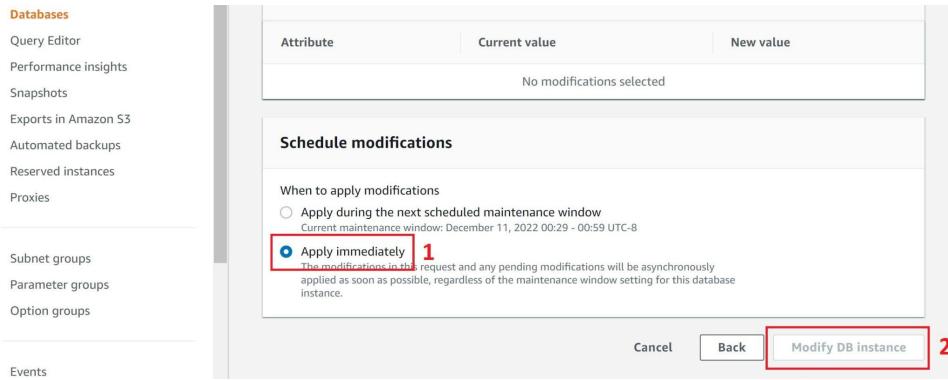
The screenshot shows the 'Settings' page for the database. A red box highlights the 'New master password' field.

d. Scroll to the bottom and click **continue**



The screenshot shows the final configuration step. It includes a 'Deletion protection' section with an unchecked checkbox for 'Enable deletion protection'. At the bottom right, a red box highlights the 'Continue' button.

- e. First, click **Apply immediately**, then click **Modify DB instance**



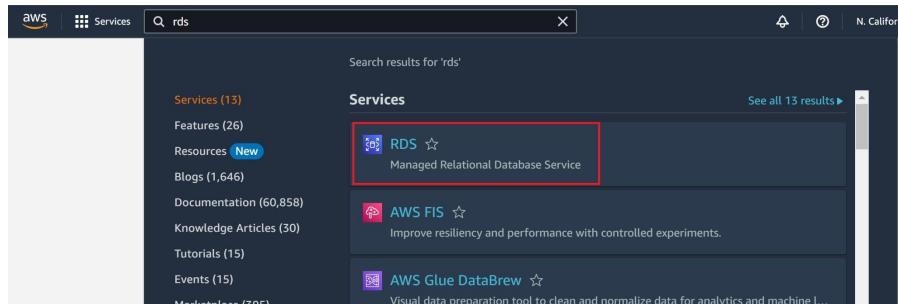
- f. Password change is complete

4.2.4 Retrieve Cognito Credentials for Environment Variables

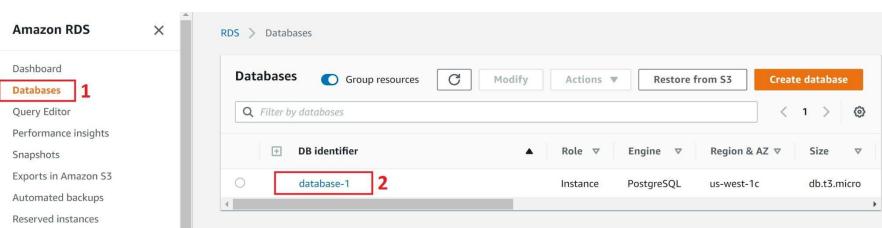
a. For **DATABASE_URL**:

i. To find **endpoint** and **port number**

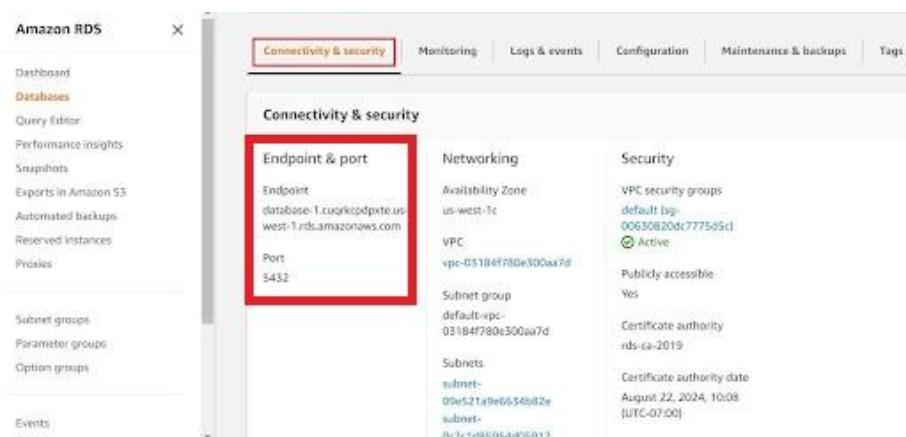
1. Log into the AWS console, click **search**, search for **RDS**, then click **RDS**



2. Click **database**, then click **database-1**

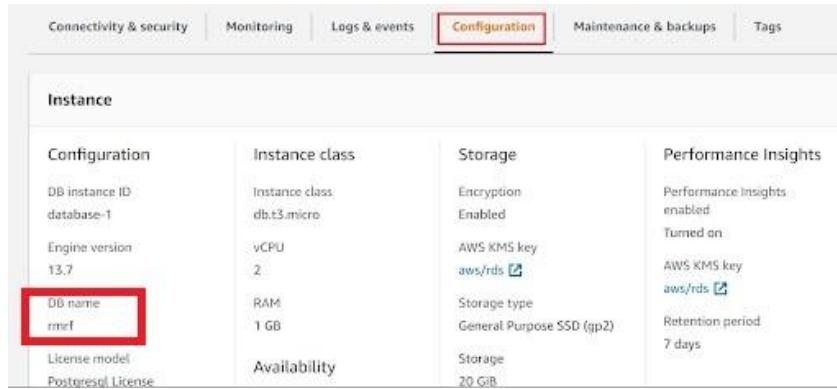


3. Scroll down, click **Connectivity & Security** and look for **Endpoint & Port Number**



ii. To find **DB name**:

1. On the same navigator bar as the previous step, click **Configuration** and look for **DB name**



Instance				
Configuration	Instance class	Storage	Performance Insights	
DB instance ID: database-1	Instance class: db.t3.micro	Encryption Enabled	Performance Insights enabled	Turned on
Engine version: 13.7	vCPU 2	AWS KMS key aws/rds 	AWS KMS key aws/rds 	
DB name rmrf	RAM 1 GB	Storage type General Purpose SSD (gp2)	Retention period 7 days	
License model PostgreSQL License	Availability	Storage 20 GiB		

- iii. Set **DATABASE_URL** equal to **postgresql://{{Endpoint}}:{portNumber}/{DB name}** with the appropriate fields replaced

b. For **DATABASE_USERNAME**:

- i. The database username is the **DB name** found in the previous step (**4.2.4a.ii**)

c. For **DATABASE_PASSWORD**:

- i. The database password is the **master password** that was set when creating the database. If this master password was not recorded, change the database password and record it (see **4.2.3**).

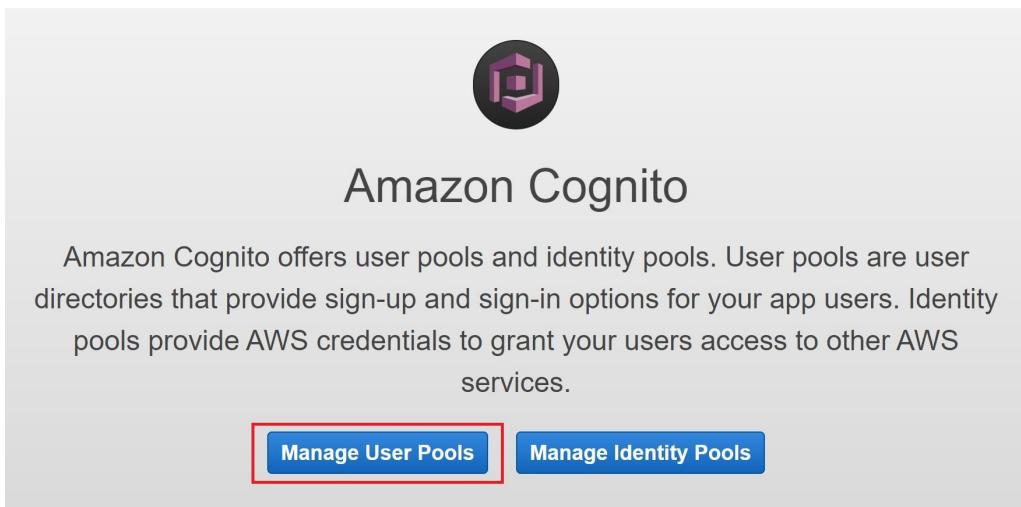
4.3 Amazon Web Services - Cognito

4.3.1 Set up AWS Cognito

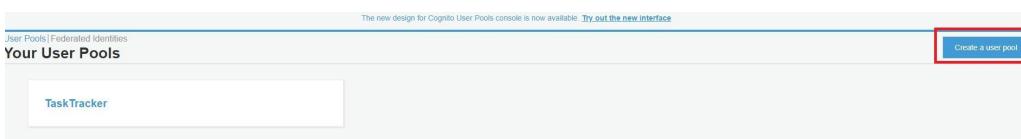
- a. Log into the **AWS Console**
- b. Click **search**, search for **Cognito**, then click on **Cognito** (inside where the red rectangle shows in the following figure)



- c. The Cognito console will be shown. Click **Manage User Pools**



- d. Click **Create User Pool**



- e. Click **Name**, then enter the pool name

The screenshot shows the 'Create a user pool' interface. On the left, a sidebar lists steps: Name, Attributes, Policies, MFA and verifications, Message customizations, Tags, Devices, App clients, Triggers, and Review. The 'Name' step is highlighted. The main area has a heading 'What do you want to name your user pool?' with a note: 'Give your user pool a descriptive name so you can easily identify it in the future.' A 'Pool name' input field is labeled 'Required'. Below are two buttons: 'Review defaults' (Start by reviewing the defaults and then customize as desired) and 'Step through settings' (Step through each setting to make your choices). A 'Cancel' button is in the top right.

- f. Click **Attributes**, then fill the information as below, or adjust it depending on your needs

The screenshot shows the 'Create a user pool' interface, specifically the 'Attributes' step. The sidebar shows steps: Name, Attributes, Policies, MFA and verifications, Message customizations, Tags, Devices, App clients, Triggers, and Review. The 'Attributes' step is highlighted. A note says: 'You can't change the sign-in and attribute options on this page after you've created your user pool. Make sure that you've decided on the settings that you want.' Below is a section titled 'How do you want your end users to sign in?'. It shows 'Username' selected, with options: 'Allow email addresses', 'Also allow sign in with verified email address', 'Also allow sign in with verified phone number', and 'Also allow sign in with preferred username (a username that your users can change)'. Other options like 'Email address or phone number' are also listed. A note says: 'You can choose to enable case insensitivity on the username input for the selected sign-in option. For example, when this option is selected, the users can sign in using either "username" or "Username".'. A checkbox '(Recommended) Enable case insensitivity for username input' is checked. Below is a section titled 'Which standard attributes do you want to require?'. It lists required attributes: 'address', 'birthday', and 'email'. It also lists optional attributes: 'nickname', 'phone number', and 'picture'. A note says: 'All of the standard attributes can be used for user profiles, but the attributes you select will be required for sign up. You will not be able to change these requirements after the pool is created. If you select an attribute to be an alias, users will be able to sign in using that value or their username. Learn more about attributes.' A 'Cancel' button is in the top right.

- g. Click **Policies**, then fill the information as below, or adjust it depending on your needs

The screenshot shows the 'Create a user pool' interface, specifically the 'Policies' step. The sidebar shows steps: Name, Attributes, Policies, MFA and verifications, Message customizations, Tags, Devices, App clients, Triggers, and Review. The 'Policies' step is highlighted. A note says: 'The new design for Cognito User Pools console is now available. Try out the new interface.' Below is a section titled 'What password strength do you want to require?'. It shows a 'Minimum length' input set to '8'. Checkboxes for 'Require numbers', 'Require special character', 'Require uppercase letters', and 'Require lowercase letters' are all checked. A note says: 'You can choose to only allow administrators to create users or allow users to sign themselves up. Learn more.' Below is a section titled 'Do you want to allow users to sign themselves up?'. It shows 'Only allow administrators to create users' selected. A note says: 'You can choose for how long until a temporary password set by an administrator expires if the password is not used. This includes accounts created by administrators.' Below is a section titled 'How quickly should temporary passwords set by administrators expire if not used?'. It shows a 'Days to expire' input set to '7'. A 'Cancel' button and a 'Save changes' button are at the bottom.

- h. Click **MFA and verification**, then fill the information as below, or adjust it depending on your needs

The new design for Cognito User Pools console is now available. [Try out the new interface](#)

[User Pools](#) | [Federated Identities](#)

Create a user pool

[Cancel](#)

MFA and verifications

Multi-Factor Authentication (MFA) increases security for your end users. If you choose "optional", individual users can have MFA enabled. You can only choose "required" when initially creating a user pool, and if you do, all users must use MFA. Phone numbers must be verified if MFA is enabled. You can configure adaptive authentication on the Advanced security tab to require MFA based on risk scoring of user sign in attempts. [Learn more about multi-factor authentication](#).

Note: separate charges apply for sending text messages.

Off Optional Required

How will a user be able to recover their account?

When a user forgets their password, they can have a code sent to their verified email or verified phone to recover their account. You can choose the preferred way to send codes below. We recommend not allowing phone to be used for both password resets and multi-factor authentication (MFA). [Learn more](#)

Email if available, otherwise phone, but don't allow a user to reset their password via phone if they are also using it for MFA
 Phone if available, otherwise email, but don't allow a user to reset their password via phone if they are also using it for MFA
 Email only
 Phone only, but don't allow a user to reset their password via phone if they are also using it for MFA
 (Not Recommended) Phone if available, otherwise email, and do allow a user to reset their password via phone if they are also using it for MFA
 None – users will have to contact an administrator to reset their passwords

Which attributes do you want to verify?

Verification requires users to retrieve a code from their email or phone to confirm ownership. Verification of a phone or email is necessary to automatically confirm users and enable recovery from forgotten passwords. [Learn more about email and phone verification](#).

Email Phone number Email or phone number No verification

Which attributes do you want to verify?

Verification requires users to retrieve a code from their email or phone to confirm ownership. Verification of a phone or email is necessary to automatically confirm users and enable recovery from forgotten passwords. [Learn more about email and phone verification](#).

Email Phone number Email or phone number No verification

You must provide a role to allow Amazon Cognito to send SMS messages

You are currently in a Sandbox environment for SMS messages. In order to send messages, go to Amazon SNS and follow the instructions to verify your phone numbers. You can then initiate your move to a production environment. [You may be redirected to the Amazon SNS console in a different region](#). Learn more.

Acknowledge to proceed

In order to send SMS messages to US phone numbers, you must set up an origination ID in Amazon Pinpoint. [You may be redirected to the Amazon Pinpoint console in a different region](#). Learn more.

Note: separate charges may apply for sending SMS text messages. See the [Worldwide SMS pricing page](#) for more details.

Amazon Cognito needs your permission to send SMS messages to your users on your behalf. [Learn more about IAM roles](#).

New role name

-SMS-Role

[Create role](#)

[Cancel](#) [Save changes](#)

- i. Click **Message customization**. You can leave it default, or adjust it depending on your needs

The new design for Cognito User Pools console is now available. [Try out the new interface](#)

[User Pools](#) | [Federated Identities](#)

Create a user pool

[Cancel](#)

Message customizations

You have chosen to have Cognito send emails on your behalf. Best practices suggest that customers send emails through Amazon SES for production User Pools due to a daily email limit. [Learn more about email best practices](#)

Do you want to customize your email address?

You can send emails from an SES verified identity. [Learn more about SES verified identities and domains](#).

SES Region
US West (N. California)

FROM email address ARN
Default

You must verify your email address with Amazon SES before you can select it. Verify an SES identity.

FROM email address
e.g. John Smith <john@smith.com>

REPLY-TO email address

Do you want to send emails through your Amazon SES Configuration?

Select Yes if you require higher daily email limits otherwise select No. Learn more about Cognito daily email limits. If you choose Yes, Cognito will send emails through your Amazon SES configuration. Refer to this documentation for additional steps.

Yes - Use Amazon SES No - Use Cognito (Default)

Do you want to customize your email verification messages?

You can choose to send a code or a clickable link and customize the message to verify email addresses. [Learn more about email verification](#).

Verification type
 Code Link

Email subject
Your verification code

Email message
Your verification code is #####.

You can customize the message above and include HTML tags, but it must include the "{#####}" placeholder, which will be replaced with the code.

Do you want to customize your user invitation messages?

SMS message
Your username is {username} and temporary password is #####.

You can customize the message above and include HTML tags, but it must include the "{username}" and "#####" placeholder, which will be replaced with the username and temporary password respectively.

Email subject
Your temporary password

Email message
Your username is {username} and temporary password is #####.

You can customize the message above and include HTML tags, but it must include the "{username}" and "#####" placeholder, which will be replaced with the username and temporary password respectively.

- j. Click **Tags**. You can leave it as default, or adjust it depending on your needs

The new design for Cognito User Pools console is now available. [Try out the new interface](#)

[User Pools](#) | [Federated Identities](#)

Create a user pool

[Cancel](#)

Tags

Do you want to add tags for this user pool?

You can create new tags by entering tag keys and tag values below.

[Add tag](#)

[Cancel](#) [Save changes](#)

- k. Click **Devices**. You can leave it as default, or adjust it depending on your needs

The new design for Cognito User Pools console is now available. [Try out the new interface](#)

Create a user pool

Do you want to remember your user's devices?

Always User Opt In No

Devices (highlighted)

Cancel Save changes

- l. Click **App Clients** and enter **App client name**. It is important to remember this step, as the application configuration needs it. You can leave it default, or adjust it depending on your needs.

The new design for Cognito User Pools console is now available. [Try out the new interface](#)

Create a user pool

Which app clients will have access to this user pool?

The app clients that you add below will be given a unique ID and an optional secret key to access this user pool.

App client name (Required)

Refresh token expiration (30 days and 0 minutes)
Must be between 60 minutes and 3650 days

Access token expiration (0 days and 60 minutes)
Must be between 5 minutes and 1 day. Cannot be greater than refresh token expiration

ID token expiration (0 days and 60 minutes)
Must be between 5 minutes and 1 day. Cannot be greater than refresh token expiration

Generate client secret

Auth Flows Configuration

Enable username password auth for admin APIs for authentication (ALLOW_ADMIN_USER_PASSWORD_AUTH) [Learn more](#).

Enable lambda trigger based custom authentication (ALLOW_CUSTOM_AUTH) [Learn more](#).

Enable username password based authentication (ALLOW_USER_PASSWORD_AUTH) [Learn more](#).

Enable SRP (secure remote password) protocol based authentication (ALLOW_USER_SRP_AUTH) [Learn more](#).

Enable refresh token based authentication (ALLOW_REFRESH_TOKEN_AUTH) [Learn more](#).

Security configuration

Prevent User Existence Errors [Learn more](#).

Legacy

Enabled (Recommended)

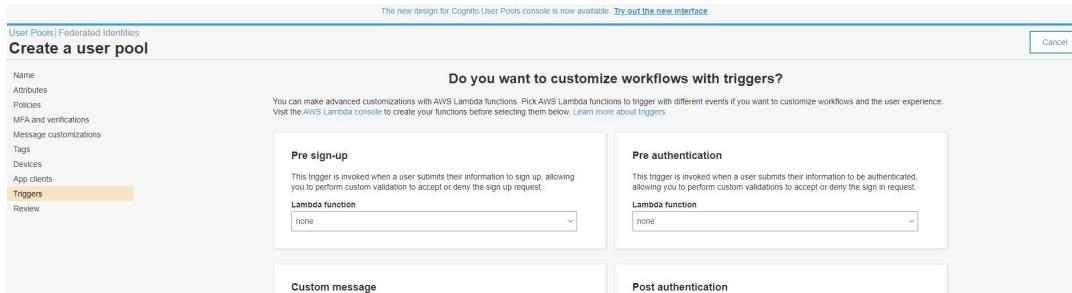
Advanced token settings

Enable token revocation
Enabling this feature adds new claims to access and id tokens, thereby increasing their size. [Learn more](#).

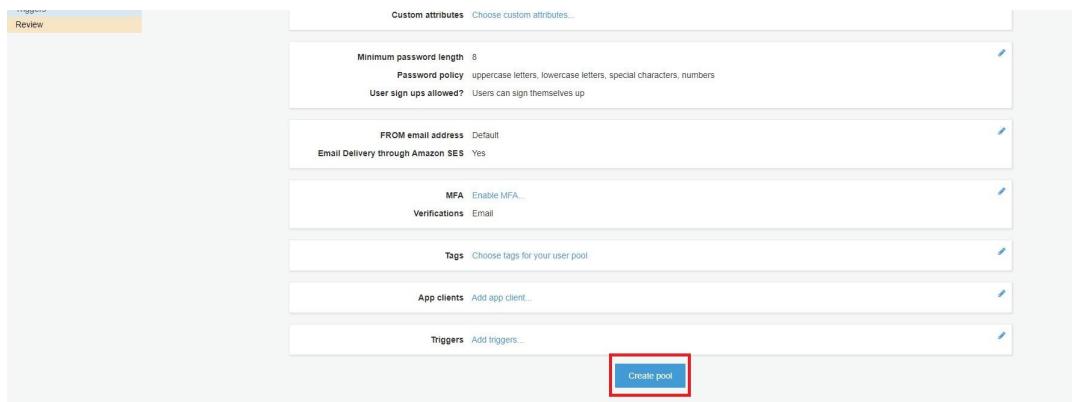
Set attribute read and write permissions

Cancel Create app client

- m. Click **Trigger**. You can leave it as default, or adjust it depending on what you need. It is required if you are using the AWS Lambda server



- n. Click **Review**, then click **Create Pool**



- o. Cognito setup is **complete**.

4.3.1 Retrieve Cognito Credentials for Environment Variables

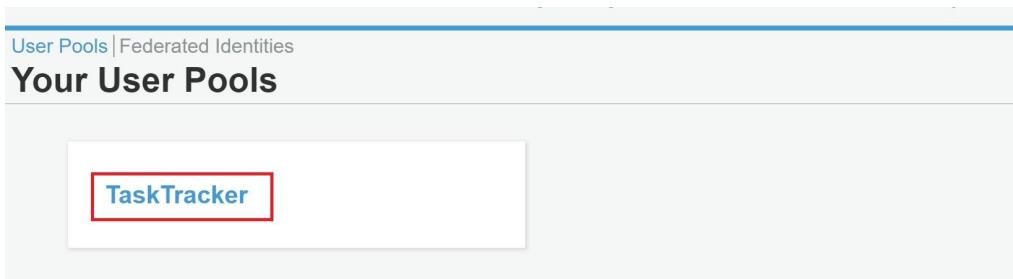
- Log into the AWS Console
- Click **search**, then search for **Cognito**. Select **Cognito** as pictured below



c. Click **Manage User Pools**

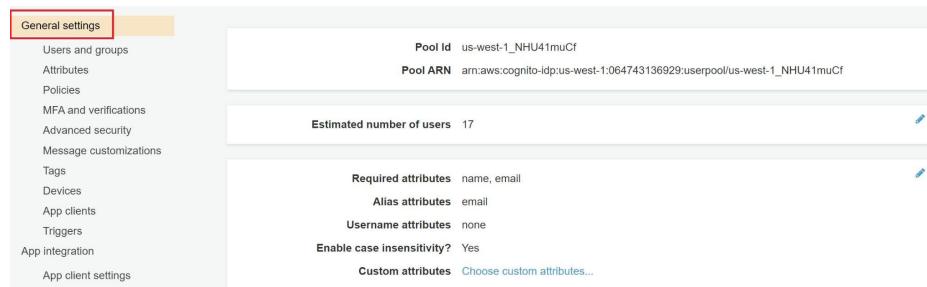


d. Click **TaskTracker**



e. For general information like **USER_POOL_ID** and **AWS_REGION**

i. Click **General Settings**



f. For **CLIENT_ID**, **CLIENT_SECRET**, and **COGNITO_CLIENT_NAME**

i. Click **App Clients**, then click **Show Details**



g. For **ACCESS_KEY**

- Log into the AWS console, click **Search**, then search **IAM**. Click **IAM**.

The screenshot shows the AWS search interface with 'Search results for 'iam'' at the top. On the left, there's a sidebar with categories like Services (8), Features (19), Resources (New), and Documentation (115,941). The main area shows four services: IAM, IAM Identity Center, Resource Access Manager, and Serverless Application Repository. The IAM card is highlighted with a red box. At the bottom, there's another 'See all 19 results' link.

- Click **Users**, then click **rnr191** (or your equivalent), then click **IAM**

The screenshot shows the IAM Users page. On the left, there's a sidebar with 'User groups' (highlighted with a red box) and 'Users' (highlighted with a red box). The main area shows a table with one user row, where 'rnr191' is highlighted with a red box. The table columns include User name, Groups, Last activity, MFA, Password age, and Active key age.

- Click **Security Credentials**

The screenshot shows the IAM Security Credentials page. On the left, there's a sidebar with 'Access management' (highlighted with a red box) and 'Users'. The main area has tabs for Permissions, Groups (2), Tags, Security credentials (highlighted with a red box), and Access Advisor. Under 'Security credentials', it shows 'Sign-in credentials' and 'Multi-factor authentication (MFA)'. At the bottom, it shows 'Access keys' with a table. One row in the table is highlighted with a red box, showing the Access key ID 'AKUAQ6EX532QR87WV3M6'.

- For **SECRET_KEY**, there is no way to find it on the user information board because AWS IAM only generates it one time. Every time you create an IAM user, you need to save the secret key in the last step of the process, as in the following example:

- Click **Add Users**

The screenshot shows the IAM Users page. On the left, there's a sidebar with 'Access management' and 'Users'. The main area shows a table with one user row. At the top right, there's a blue 'Add users' button, which is highlighted with a red box.

ii. Fill in **User name**, then click **Access Key**, then click **Next permissions**

Add user

Set user details

You can add multiple users at once with the same access type and permissions. Learn more

User name* simple

Add another user

Select AWS access type

Select how these users will primarily access AWS. If you choose only programmatic access, it does NOT prevent users from accessing the console using an assumed role. Access keys and autogenerated passwords are provided in the last step. Learn more

Select AWS credential type*

Access key - Programmatic access
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

Password - AWS Management Console access
Enables a **password** that allows users to sign-in to the AWS Management Console.

iii. Create or add **Group**, then click **Next: Tags**

Add user

Set permissions

Add user to group

Create group Refresh

Showing 4 results

Group	Attached policies
<input type="checkbox"/> accessCognito	AmazonCognitoDeveloperAuthenticatedIdentities and 1 more
<input type="checkbox"/> cognitoUsers	AmazonESCognitoAccess
<input checked="" type="checkbox"/> powerUser	AmazonCognitoPowerUser
<input checked="" type="checkbox"/> UsersPool	cognitoUsersPool

Set permissions boundary

Cancel Previous **Next: Tags**

-
- iv. Leave as default, or adjust based on needs, then click **Next: Review**

Add user

1 2 3 4 5

Add tags (optional)

IAM tags are key-value pairs you can add to your user. Tags can include user information, such as an email address, or can be descriptive, such as a job title. You can use the tags to organize, track, or control access for this user. [Learn more](#)

Key	Value (optional)	Remove
<input type="text" value="Add new key"/>	<input type="text"/>	Remove

You can add 50 more tags.

- v. Review, then click **Create user**

Add user

1 2 3 4 5

Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

User details

User name	simple
AWS access type	Programmatic access - with an access key
Permissions boundary	Permissions boundary is not set

Permissions summary

The user shown above will be added to the following groups.

Type	Name
Group	powerUser
Group	UsersPool

Tags

No tags were added.

Cancel Previous **Create user**

vi. **Save or copy the secret access key** somewhere safe before closing

Add user

1 2 3 4 5

Success
You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://064743136929.signin.aws.amazon.com/console>

Download .csv

User	Access key ID	Secret access key
simple	AKIAQ6EX532Q4XFLD24A	my9rOEx4EPJtnGa4i/o+20f ulnJpypafkAOGs6ln Hide

Close

i. For **COGNITO_ISSUER_URI**:

- Go to https://docs.aws.amazon.com/general/latest/gr/cognito_identity.html, and find the **Endpoint**.
- Set **COGNITO_ISSUER_URI** to [https://\[endpoint\]/\[pool-id\]](https://[endpoint]/[pool-id])

j. For **SUBDOMAIN**:

- On the **Cognito Dashboard**, navigate to **Domain name**. This value will be used for the **SUBDOMAIN** environment variable.

User Pools | Federated identities
TaskTracker

General settings

- Users and groups
- Attributes
- Policies
- MFA and verifications
- Advanced security
- Message customizations
- Tags
- Devices
- App clients
- Triggers
- App integration
- Domain name

What domain would you like to use?

Type a domain prefix to use for the sign-up and sign-in pages that are hosted by Amazon Cognito. The prefix must be unique across the selected AWS Region. Domain names can only contain lower-case letters, numbers, and hyphens. [Learn more about domain prefixes](#).

Amazon Cognito domain
Prefixed domain names can only contain lower-case letters, numbers, and hyphens. [Learn more about domain prefixes](#).

Domain prefix

.auth.us-west-1.amazoncognito.com

Your own domain
This domain name needs to have an associated certificate in AWS Certificate Manager (ACM). You also need the ability to add an alias record to the domain's hosted zone after it's

Delete domain

5.0 User Interaction

5.1 Authentication

Sign in with your username and password

Username

amrins

Password

•••••••

[Forgot your password?](#)

[Sign in](#)

Need an account? [Sign up](#)

a.

- i. Auth (including logout) is configured in the file
`/src/main/java/com.rmrroot.tasktracker222/configurers/SecurityConfiguration`
in the method **configure()**
- ii. User pool (for Cognito) is configured in the file
`src/main/java/com/rmrroot/tasktracker222/awsCognito/CreatePoolClient.java`
in the methods **createCognitoClient()**, **getUserInfo()**, and
deleteUserByUsername()

5.2 Drill Schedule

The screenshot shows a web-based drill schedule application. On the left is a calendar grid for the week of December 4, 2022. The grid shows various events scheduled across the days from Monday to Sunday. A specific event on Tuesday, December 6, from 07:00 to 12:00, is highlighted in red. To the right of the calendar is a panel titled "Event Details" which contains the following information:

Introductory Meeting for AB

Date: Tue, 2022-11-29
Time: 07:00 - 12:00
Location: Wg Training rm
Report To: Bruce Wayne
Participants: AB, Training

This is a sample description for an event. Not all events will have descriptions. Only users with the rank of AB or users who are part of the training team will be assigned this event.

Below this, there is a scrollable list of other events:

- Introductory Meeting for AB
- Concurrent Event 1
- Concurrent Event 2
- Concurrency 3
- A really long sample event

- The drill schedule page is served by the controller in the file [src/main/java/com/rmrroot/tasktracker222/controllers/DrillController.java](#) in the **drillScheduleRecipient()** method
- The template used for the drill schedule page is the file [src/main/resources/templates/DrillScheduler.html](#)
- The stylesheet used for the drill schedule page is the file [src/main/resources/static/js/user-management.js](#)
- The procedural generation for the calendar and the intraday summary is handled in the file [src/main/resources/static/js/drill-schedule-generator.js](#)
- The rendering of events on the calendar, including concurrency handling, is in the file [src/main/resources/static/js/drill-schedule-render.js](#)
- The selection of events (i.e. bringing an event into focus on the calendar and intraday summary, as well as displaying details of the event) is handled in the file [src/main/resources/static/js/drill-schedule-selection.js](#)
- The selection of drills shown to the user are calculated in the file [src/main/java/com/rmrroot/tasktracker222/services/DrillDaoImpl.java](#) in the method **findDrillsInWeekOfDateByID()**, which takes the user's ID and the currently selected week as inputs
- The algorithm handling concurrency (i.e., calculating when extra columns are needed if there are two concurrent events) can be found in the file [src/main/java/com/rmrroot/tasktracker222/services/DrillDaoImpl.java](#) in the method **findConcurrencyInDay()**

-
- i. Drill entities are defined in the file

src/main/java/com/rmrroot/tasktracker222/entities/Drill.java

5.3 Drill Management

The screenshot shows a web-based application for managing drills. On the left, there is a sidebar with icons for home, search, and refresh. The main area has a header "Week of: 12/04/2022". Below this is a table listing various events:

Title	Date	Time	Location
+ New Event			
Introductory Meeting for AB	2022-11-29	07:00-12:00	Wg Training rm
Concurrent Event 1	2022-11-29	08:00-14:00	The Cafeteria
Concurrent Event 2	2022-11-29	09:00-11:00	91S Lg conf
Concurrency 3	2022-11-29	13:00-14:00	DGS triangle
A really long sample event	2022-11-29	20:30-18:00	The Oasis
End of year performance review	2022-12-01	10:00-13:00	91S Sm conf
Uniform distribution	2022-12-02	10:00-11:00	Main Office
Team Picture	2022-12-02	11:00-14:00	SR-71
An edited event	2022-12-03	05:00-14:00	The Oasis

To the right is an "Event Editor" form:

Event Editor	
Title	<input type="text" value="Enter title..."/>
Color	<input type="text" value="Select one..."/>
Date	<input type="text" value="Choose date..."/>
Start Time	<input type="text" value="Choose start time..."/>
End Time	<input type="text" value="Choose end time..."/>
Location	<input type="text" value="Choose location..."/>
Report To	<input type="text" value="Select one..."/>
Participants	<input type="text" value="Choose participants..."/>
Description	<input type="text" value="Drill Description"/>

At the bottom are two buttons: "Submit" (green) and "Delete" (red).

- a. The drill management page is served by the controller in the file **src/main/java/com/rmrroot/tasktracker222/controllers/DrillController.java** in the method **drillScheduleManager()**
- b. The template used for the drill manager page is the file **src/main/resources/templates/DrillManagement.html**
- c. The stylesheets used for the drill management page are **all the files** found in the **src/main/resources/static/drill-management** directory
- d. The functions used for formatting and dynamically updating fields for drill management can be found in the file **src/main/resources/static/js/drill-management.js**
- e. Submissions on the drill management page (new drill or edit drill postmapping) is served by the controller in the file **src/main/java/com/rmrroot/tasktracker222/controllers/DrillController.java** in the method **editDrill()**
- f. Submissions on the drill management page (delete drill postmapping) is served by the controller in the file **src/main/java/com/rmrroot/tasktracker222/controllers/DrillController.java** in the method **deleteDrill()**
- g. Submissions of new drills or edits to existing drills to the database are handled in the **src/main/java/com/rmrroot/tasktracker222/services/DrillDaoImpl.java** file in the **save()** and **update()** methods respectively

-
- h. Deletions of drills in the database are handled in the
src/main/java/com/rmrroot/tasktracker222/services/DrillDaoImpl.java file in the
deleteById() method
 - i. Drill entities are defined in the file
src/main/java/com/rmrroot/tasktracker222/entities/Drill.java
 - j. Groups (i.e. rank, flight, workcenter, teams, locations) are defined in the file
src/main/java/com/rmrroot/tasktracker222/entities/Group.java

5.4 User Management

User List				User Editor	
Name	Rank	Workcenter	Flight	First Name	Amrin
Amrin Sandhar	AB	SCOS	SCOI	Last Name	Sandhar
Brian Frey	SrA	SCOP	CMD	Mil Email	amrin@us.af.mil
Bruce Wayne	MSgt	SCOX	SCP	Civ Email	amrinsandhar@gmail.com
John Smith	AB	SCOX	CMD	Personal Phone	9165555555
Noel Poulite	Amn	SCOX	SCO2	Office Phone	9167777777
Non Ad	Amn	SCOP	CMD	Rank	AB
Travis Fitzgerald	MSgt	SCOT	SCOI	Flight	SCOI
Vincent Vega	SSgt	SCOT	SCP	Workcenter	SCOS
Visoth Cheam	SrA	SCOS	SCO2	Teams	222ALL ✕ PTL ✕ Booster ✕ Top3 ✕
				Admin	True
				Submit	Delete

- The user management page is served by the controller in the file **src/main/java/com/rmrroot/tasktracker222/controllers/UserController.java** in the method **userManager()**
- The template used for the user manager page is the file **src/main/resources/templates/UserManagement.html**
- The stylesheets used for the user management page are **all the files** found in the **src/main/resources/static/user-management** directory
- The functions used for formatting and dynamically updating fields for user management can be found in the file **src/main/resources/static/js/user-management.js**
- Submissions on the user management page (edit user postmapping) is served by the controller in the file **src/main/java/com/rmrroot/tasktracker222/controllers/UserController.java** in the method **userEditSubmit()**
- Submissions on the user management page (delete user postmapping) is served by the controller in the file **src/main/java/com/rmrroot/tasktracker222/controllers/UserController.java** in the method **userEditDelete()**
- Submissions of user edits to the database are handled in the **src/main/java/com/rmrroot/tasktracker222/services/UsersDaoServiceImpl.java** file in the **update()** method

-
- h. Deletions of users in the database are handled in the
`src/main/java/com/rmrroot/tasktracker222/services/UsersDaoServiceImpl.java` file in
the **deleteById()** method
 - i. User entities are defined in the file
`src/main/java/com/rmrroot/tasktracker222/entities/User.java`
 - j. Groups (i.e. rank, flight, workcenter, teams, locations) are defined in the file
`src/main/java/com/rmrroot/tasktracker222/entities/Group.java`

5.5 User Registration

New User Registration

First Name*

Last Name*

Mil Email*

Civ Email*

Personal Phone

Office Phone

Rank*
 Choose rank

Flight*
 Choose flight

Workcenter*
 Choose workcenter

Teams*
 Choose teams

Register

- a. The user registration page is served by the controller in the file **src/main/java/com/rmrroot/tasktracker222/controllers/UserController.java** in the method **addUser()**
- b. The template used for the user registration page is the file **src/main/resources/templates/RegistrationForm.html**
- c. The stylesheets used for the user registration page are **all the files** found in the **src/main/resources/static/registration** directory
- d. The functions used for formatting and dynamically updating fields for user registration can be found in the file **src/main/resources/templates/RegistrationForm.html** in the inline `<script>` tag found at the bottom of the `<body>`
- e. Submissions on the user registration page (new user postmapping) is served by the controller in the file **src/main/java/com/rmrroot/tasktracker222/controllers/UserController.java** in the method **saveUser()**
- f. Submissions of user edits to the database are handled in the **src/main/java/com/rmrroot/tasktracker222/services/UsersDaoServiceImpl.java** file in the **save()** method
- g. User entities are defined in the file **src/main/java/com/rmrroot/tasktracker222/entities/User.java**
- h. Groups (i.e. rank, flight, workcenter, teams, locations) are defined in the file **src/main/java/com/rmrroot/tasktracker222/entities/Group.java**

5.6 Pending Approval

Your account is pending approval.

Your organization must approve your registration before you will be able to access this application.

- a. The pending approval page is served by the controller in the file
src/main/java/com/rmrroot/tasktracker222/controllers/UserController.java in the method **pendingApproval()**
- b. The template used for the user registration page is the file
src/main/resources/templates/PendingApproval.html

6.0 Database

6.1 ERDs

Note : There is no relational usage of the database. Only two, independent, tables are used.

a. Drills

drills	
drill_name	varchar(255)
date	date
start_time	time
location	varchar(255)
description	varchar(1000)
end_time	time
participants	text[]
color	varchar(255)
report_to_id	integer
 id	integer

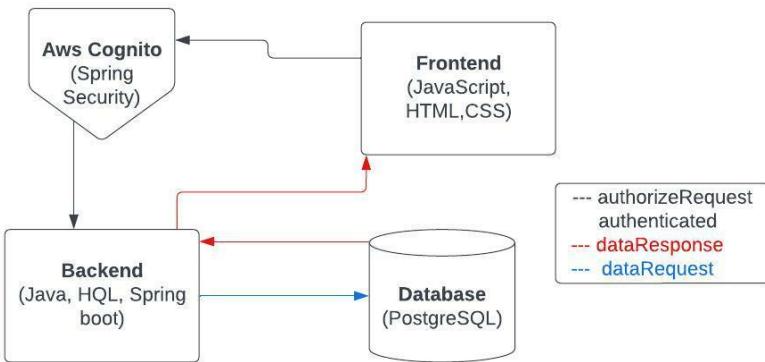
```
create table drills
(
    id          serial
    primary key,
    drill_name  varchar(255),
    date        date,
    start_time  time,
    location    varchar(255),
    description  varchar(1000),
    end_time    time,
    participants text[],
    color       varchar(255),
    report_to_id integer
);
```

b. Users

users	
flight	varchar(255)
rank	varchar(255)
work_center	varchar(255)
first_name	varchar(255)
last_name	varchar(255)
mil_email	varchar(255)
civ_email	varchar(255)
username	varchar(255)
email	varchar(255)
office_number	varchar(255)
phone_number	varchar(255)
admin	boolean
teams	character varying[]
approved	boolean
 id	integer

```
create table users
(
    id          serial
    primary key,
    flight      varchar(255),
    rank        varchar(255),
    work_center varchar(255),
    first_name  varchar(255),
    last_name   varchar(255),
    mil_email   varchar(255),
    civ_email   varchar(255),
    username    varchar(255),
    email       varchar(255),
    office_number varchar(255),
    phone_number varchar(255),
    admin       boolean,
    teams       character varying[],
    approved    boolean
);
```

6.2 Other diagrams showing data flow



a.