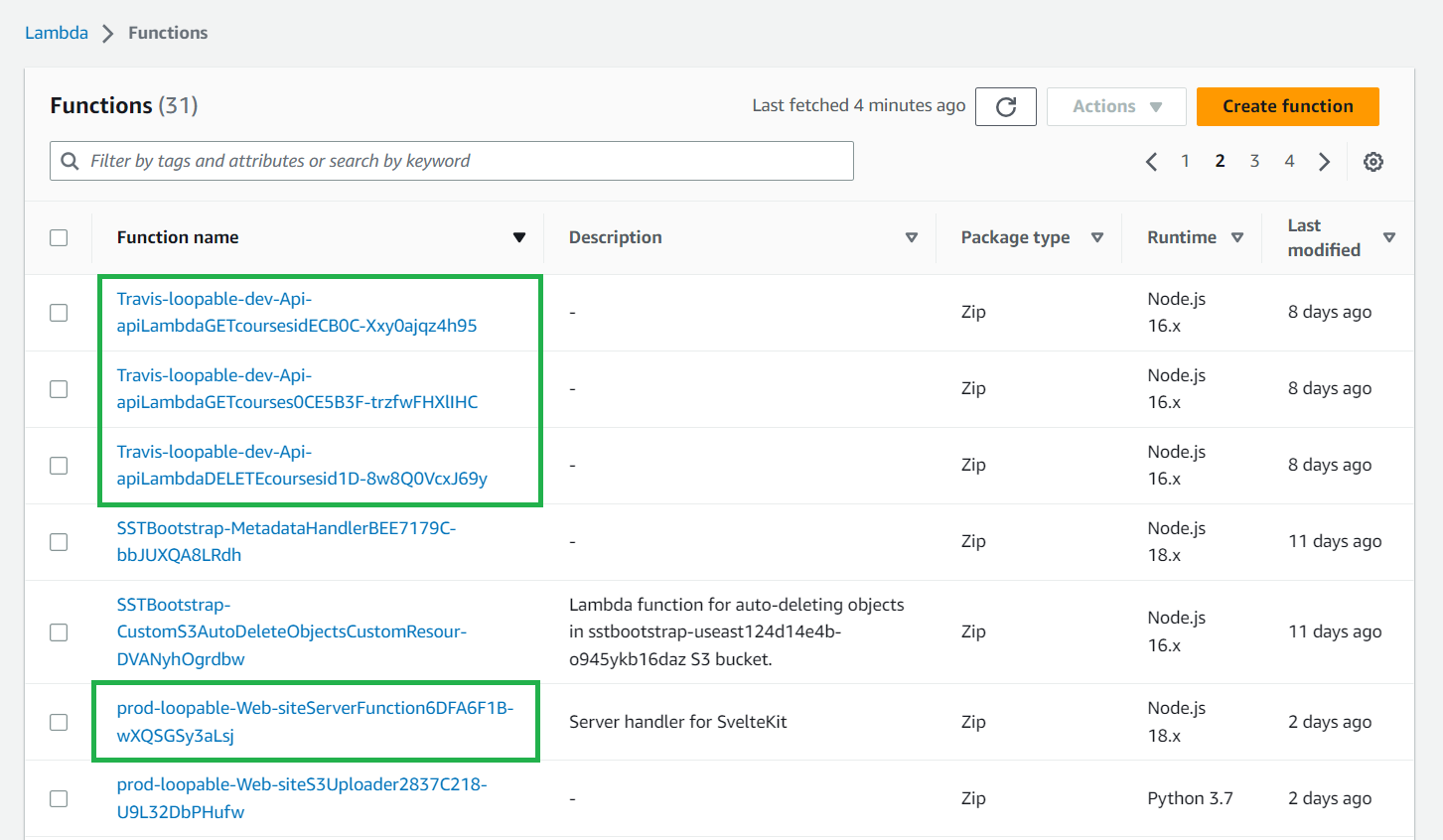
1. **Overview of the architecture**
   1. **List of the services used**

* Cognito

Cognito is used for User Authentication and Lambda functions are authorized with Cognito JWT auth.

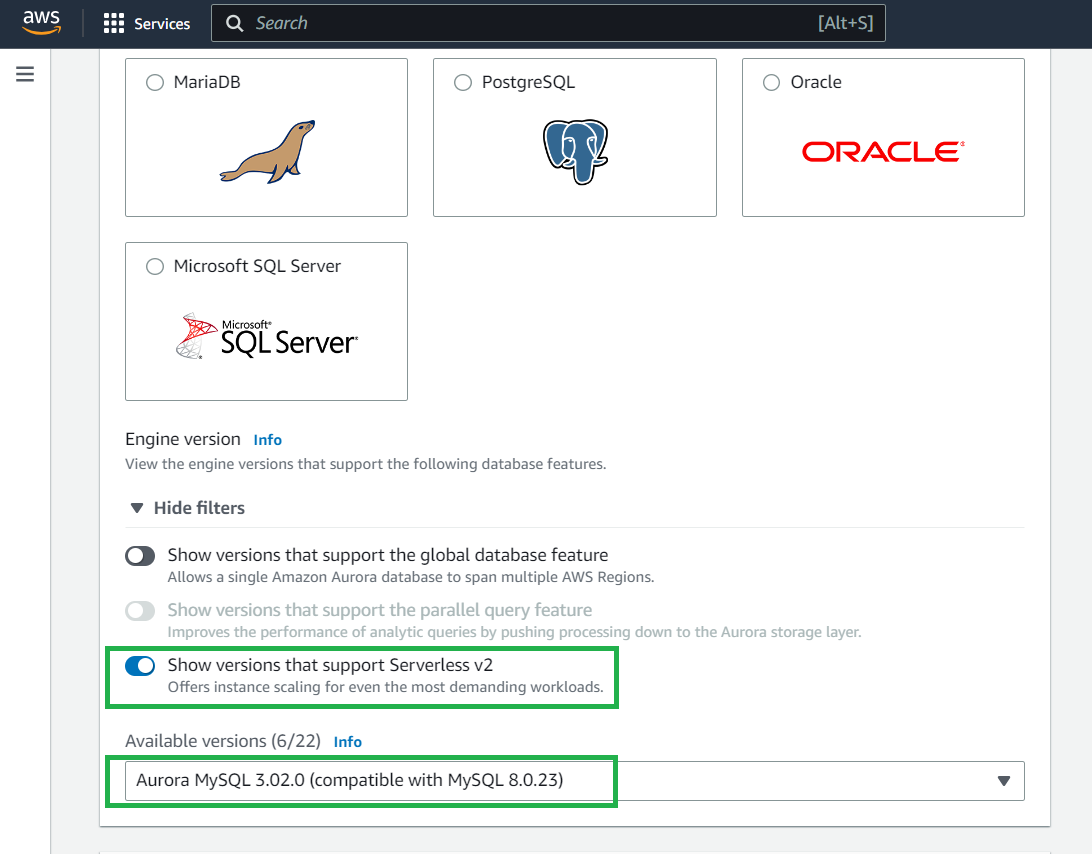
* API Gateway
* Lambda

Svelte frontend and backend RESTful APIs are implemented and served by Lambda,  


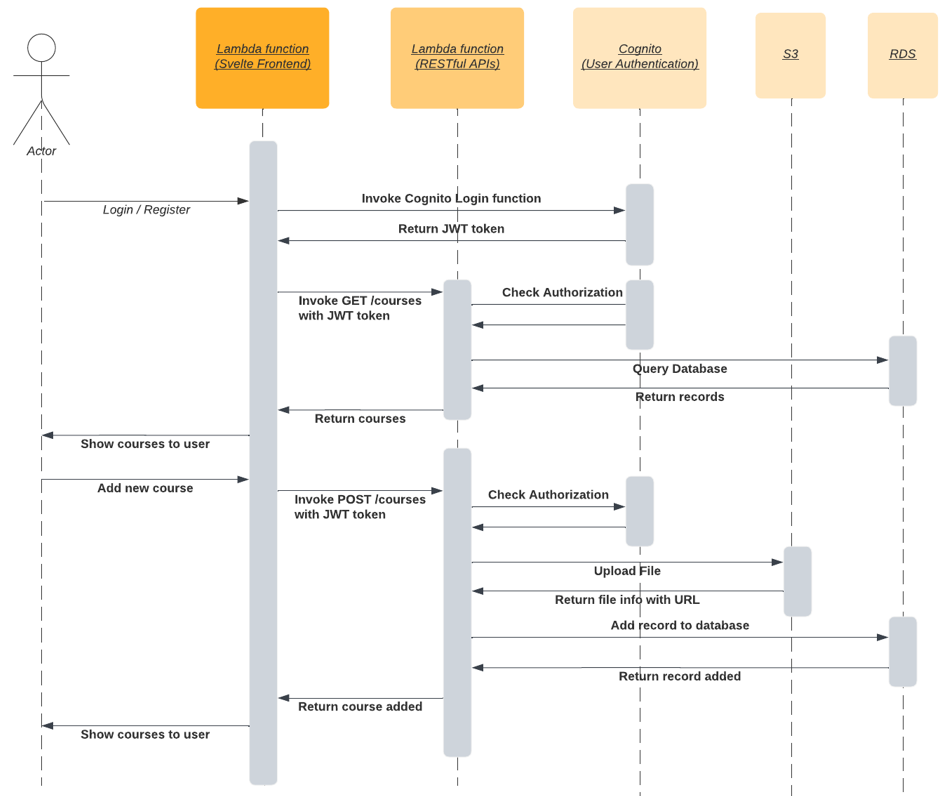
In the above image, the top 3 functions are for RESTful APIs. And the below one is the function to serve Svelte website. We can find all these functions in the AWS Lambda/functions page.

* RDS

AWS RDS (Aurora MySQL) is used as a database.

For development, we use Serverless v1 (mysql 5.7). We can change this to Serverless v2 on production. (stacks/DB.ts: line 6)  
Valid mysql versions can be found in the “RDS/Create Database” page.  


* 1. **List of the tools used**
* SST v2.16.0 (Serverless Stack: Serverless application development framework) We must be careful with SST version, because some unstable versions are making troubles in development.
* Svelte v3.54.0 (We can use any version of Svelte)
* Yarn (node package manager)
* AWS CLI
* Node v18
  1. **Sequence diagram – including how the application is distributed on Lambdas, i.e., the main website is on Lambda #1, the courses API is on Lambda #2, etc.**

****

1. **Development process**
   1. **How to start a dev environment locally**

$ yarn sst dev

// All necessary AWS stacks will be created for development.

$ yarn dev

// Svelte frontend will run and served locally and it will be bound to AWS stacks created.

* 1. **How to deploy a development environment**

$ yarn sst deploy –stage dev

* 1. **How to deploy a staging environment**

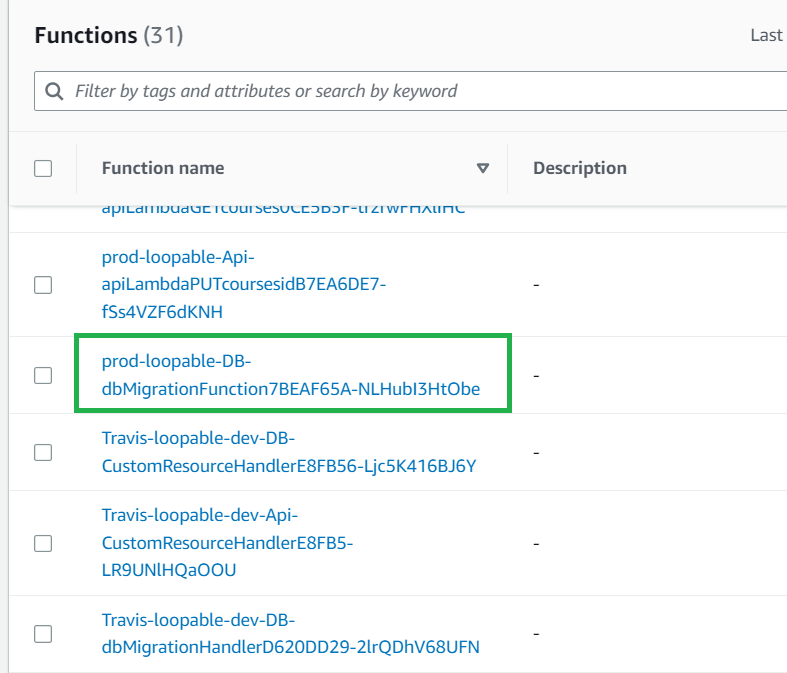
$ yarn sst deploy –stage staging

* 1. **How to deploy the production environment**

$ yarn sst deploy –stage prod

* 1. **How to deploy a code change to each environment once they are set up**

If we specify the stage option in the deploy command, changes will be made on the corresponding AWS instances. For example, if you specify stage as “prod” and make changes on db migration script, SST will re-deploy only the highlighted instance in the below image. As you can see here, instance name will be prefixed with stage name like “prod-loopable-DB…”.

****When we run deploy commands, all resources will be deployed incrementally, which means only new changes will be reflected.

* 1. **How to update the database schema**
* In order to make any changes on database schema, we need to define the migration scripts in the “**services/migrations**” folder.
* When we deploy an application, the entire migration scripts will be executed in alphabetical order. Each migration script represents each table in the database and has 2 functions: up (CREATE TABLE) and down (DROP TABLE). Whenever we run migration scripts, it will drop the existing table and create it again.
  1. **How to manage resources with SST**

We can manage and debug resources on the SST console. SST console can be accessed via browser after you run the following command on command prompt.

1. **Clean up**
   1. **How to remove each environment**

$ yarn sst remove –stage <stage>

* 1. **Things I need to be aware of, e.g., env variables, etc.**
* **AWS CLI** must be installed for local development.
* **AWS\_ACCESS\_KEY\_ID**, **AWS\_SECRET\_ACCESS\_KEY** must be set in “**.env**” or “**~/.aws/configure**” file.
* It can also be done in command prompt.

$ aws configure

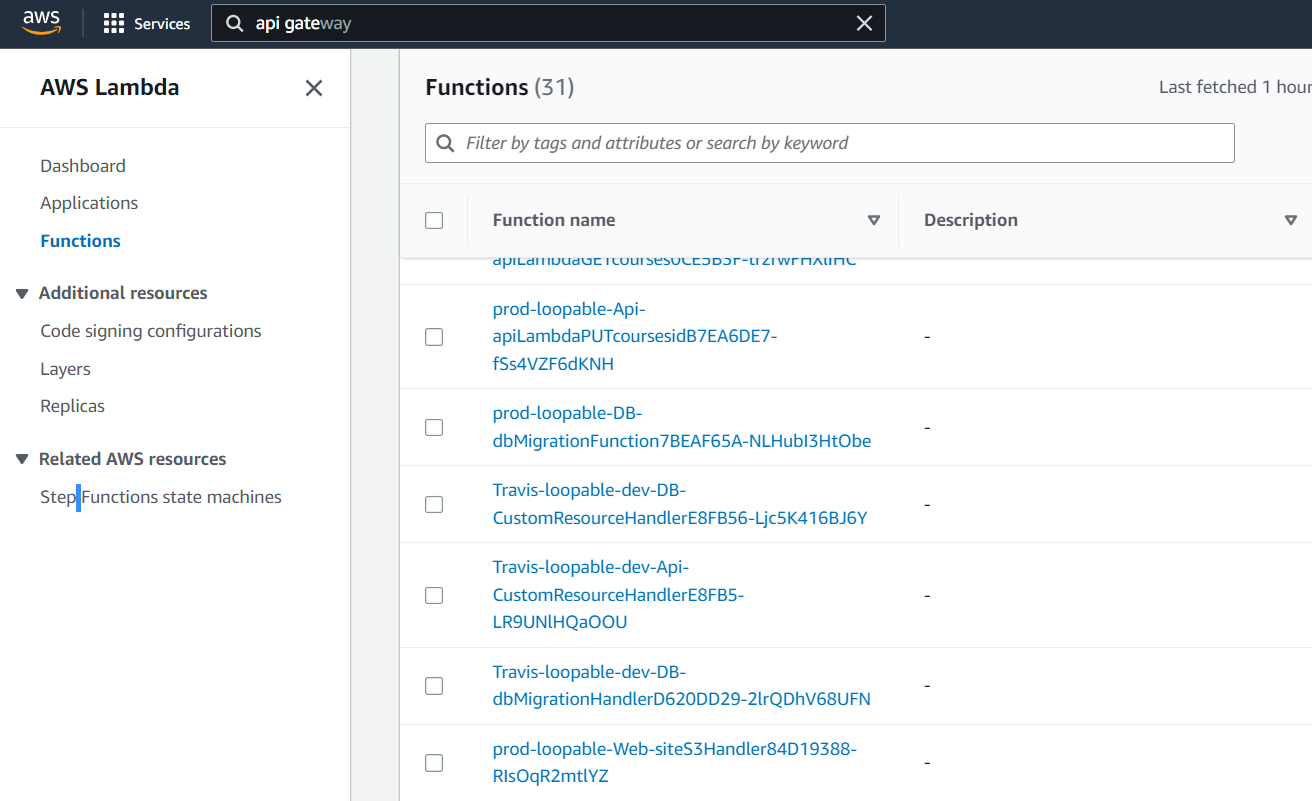
* 1. **Troubleshooting issues**
* We can manage and debug resources on the **SST console**. SST console can be accessed via browser after you run the following command on command prompt.

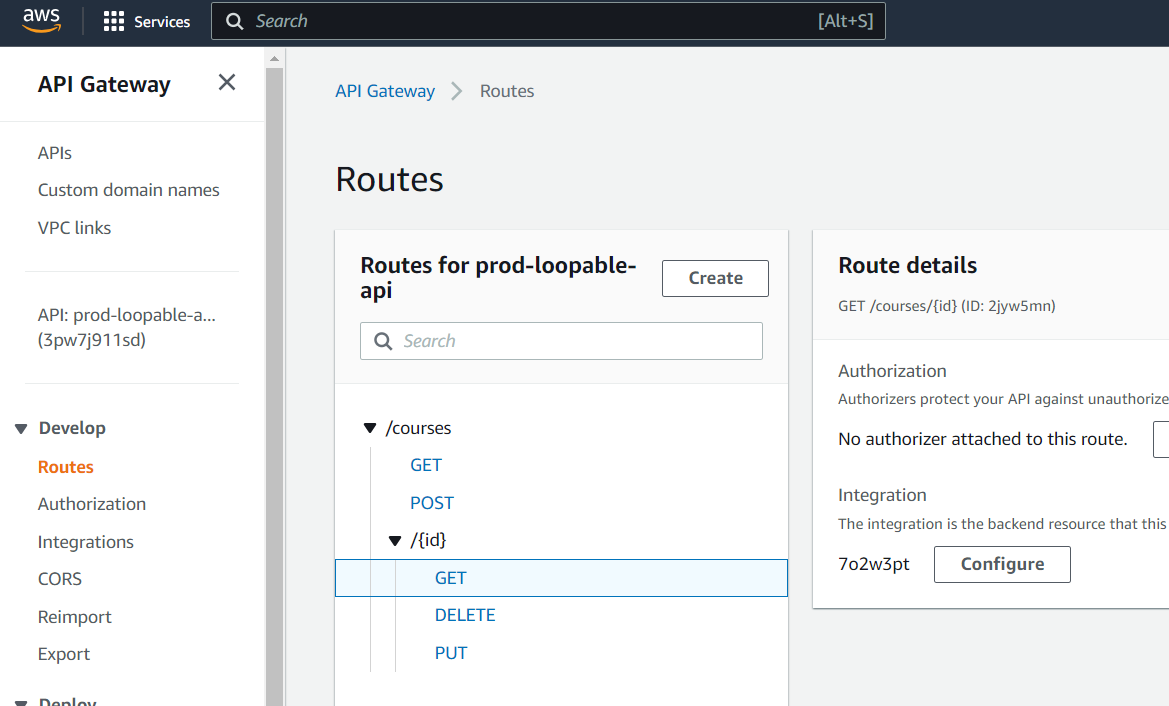
$ yarn sst console –stage=<stage>

* While we are developing locally, all errors and warnings will be listed on command prompt or SST console.
  1. **How to ensure I don’t accidentally increase my bill with Aurora.**

Since we are using RDS Serverless v1 (mysql 5.7), there won’t be any trouble with billing.

* 1. **Services from “yarn sst dev” not appearing in the console (where to find them?)**
* All resources created can be found on **AWS/CloudFormation** page.
* Whether the stage is dev or prod, once we deploy an application ($ yarn sst deploy), all instances can be found and managed on the AWS console (Lambda, API Gateway, RDS and Cognito pages).





* But in the dev environment ($ yarn sst dev), instances can be found only on the CloudFormation page.

