# How to setup AWS Environment

## Redis

Create a Redis in eu-central-1 zone. If you will create it with default port (6379) then ensure that inbound and outbound rules of the security group are correct.

## S3

Create a s3 bucket in Frankfurt Region. The name of bucket is linnworksexcel and make it public from permissions tab.

(I know that it is not good idea that making it public. But I couldn’t connect s3 from queue service. So, I need to examine it how my app can connect)

## VPC

If you don’t have default VPC. Please create it first and you will need at least 2 subnets for this VPC.

You also need route table and you should bind s3 endpoint from Routes tab. If you don’t have s3 endpoint, please endpoint for s3.

Then create a internet gateways in order to connect RDS and bind this internet gateway to route table.

## RDS

Create Microsoft Sql Server Express Edition and make it public (if you want to connect from your local environment).

Ensure that VPC of database is same with Lambda’s.

# How to publish application on AWS

## Queue Micro Service

There is a json file under LinnWorks.Queue.MicroService project which is aws-beanstalk-tools-defaults.json. You can use this file, but you should update profile name.

Ensure that VPC of this Beanstalk is same with Redis’s.

This is a web application and you should give URL of the app to Front End App (I explain below)

Before publish please update Redis ConnectionString under RedisConnectionFactory class. You will the see the string when you open it.

(I know that it is not good idea to embed this string into code)

## Lambda

There is a json file under LinnWorks.Processor.MicroService project which is aws-lambda-tools-defaults.json. You can use this file, but you should update profile, function-subnets and function security group.

Ensure that VPC of this Lambda is same with Redis’s.

After that you should add s3 trigger to this lambda and ensure that you select correct bucket name. Then please don’t forget to save the changes and lambda will be ready.

# How to Run Apps

We have 4 apps. If you publish micro services on AWS. Then 2 apps left which are FrontEnd and Api.

## Front End

Under SolutionItems folder you can find Run.Web.cmd file. You can execute this file in order to run app.

But first you should update some endpoints.

### Queue Service Endpoint

You must update queueApi.js file which is under LinnWorks.Task\src\FrontEnd\LinnWorks.Task.FrontEnd\ClientApp\src\api

### Api Endpoint

If the Api app runs on your local environment. You don’t need to update anything. But if you want to update Api endpoint, you may update Api.js file which under LinnWorks.Task\src\FrontEnd\LinnWorks.Task.FrontEnd\ClientApp\src\api

# Api

Under SolutionItems folder you can find Run.WebAPI.cmd file. You can execute this file in order to run app. Before run it, you may change connection string of database and you can find it in ApplicationDbContext class.

Database and related tables will be created when you start to run Api app.

(Lambda and this app use same database connection string. So I suggest to update connection string before you publish lambda)

(I tried to use SecretsManager in order to get connection string but I couldn’t connect SecretsManager then I removed it. But you can review SecretsManager project)

# How to Upload File

After you setup AWS and run applications. You can navigate <http://localhost:5000/> and you will see the Import File and Fetch Data under NavBar. You can just choose correct formatted csv and upload it.

# How to Update Imported Data

Under Fetch Data page you will see filter section, 2 buttons and 1 table.

If you have some data on table then click any cell and it will turn into editable field. After you edit any field then click Save Changes button.

# What can be improved

1. No Authorization
2. Docker. The apps are without docker support right now. Because my computer didn’t allow me to do that.
3. Better User Interface
4. Logging (no logging right now)
5. Exception Handling (I didn’t do anything for exception handling)
6. I didn’t try to cover all branches. I just want to show how I write unit test. 1 unit test is broken which lambda function unit test. Because I don’t really know how I should test it. Because function connects Redis, S3 and RDS. I need to mock related class. But I can’t do it. Maybe I need to change function or find some way, or you suggest me how to do it 😊
7. Architecture of lambda function can be improved. It might be good idea to create more than one function. One function can connect Redis, s3 and parse file and give the object to another micro service (function). The other micro service can just insert data.
8. All ConnectionStrings may be in Secrets Manager or somewhere else instead of embedded into code.

I suppose I didn’t forget anything about how to setup/run. But you can’t run app please contact with me 😊

Best Regards,