Configuring Properties

Introduction Spring boot profiles

Now there may be requirement that on various environments we can have different DB configuration this can be achieved by profiling concept provided by Springboot

We can set the current active profile in application.properties with the below property

Spring.profiles.active =prod

Now create the application properties file with appending prod or qa

application\_prod.properties

application\_qa.properties

depending upon the value specified in profile it will automatically read the properties from that particular file

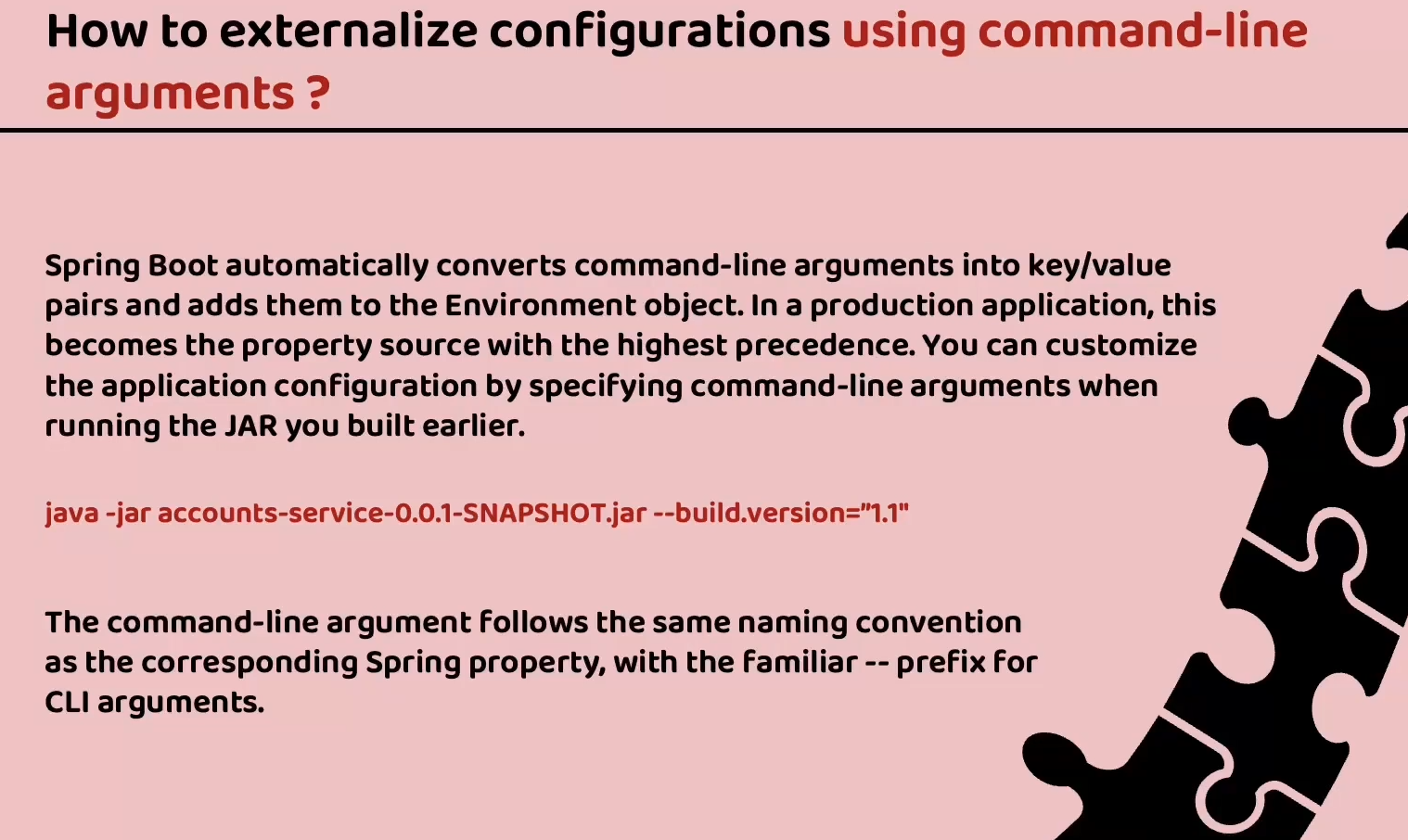
say in our case application\_prod properties will be loaded.

But as microservices grow maintaining properties for each is cumbersome process so will externalize these properties

Now if we want to change the active profile then we have to change the value of spring profile active in all the service related property file . But this is again a cumbersome process so spring boot has provided many different ways to set the properties at run time

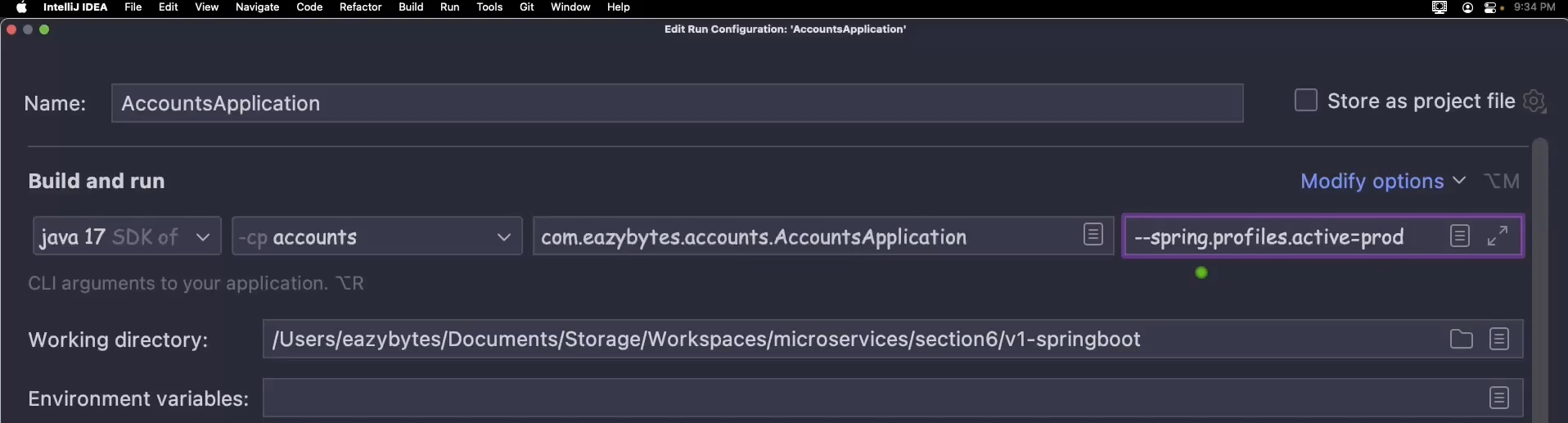
Using command line

Command line has highest property over the property defined in application.properties



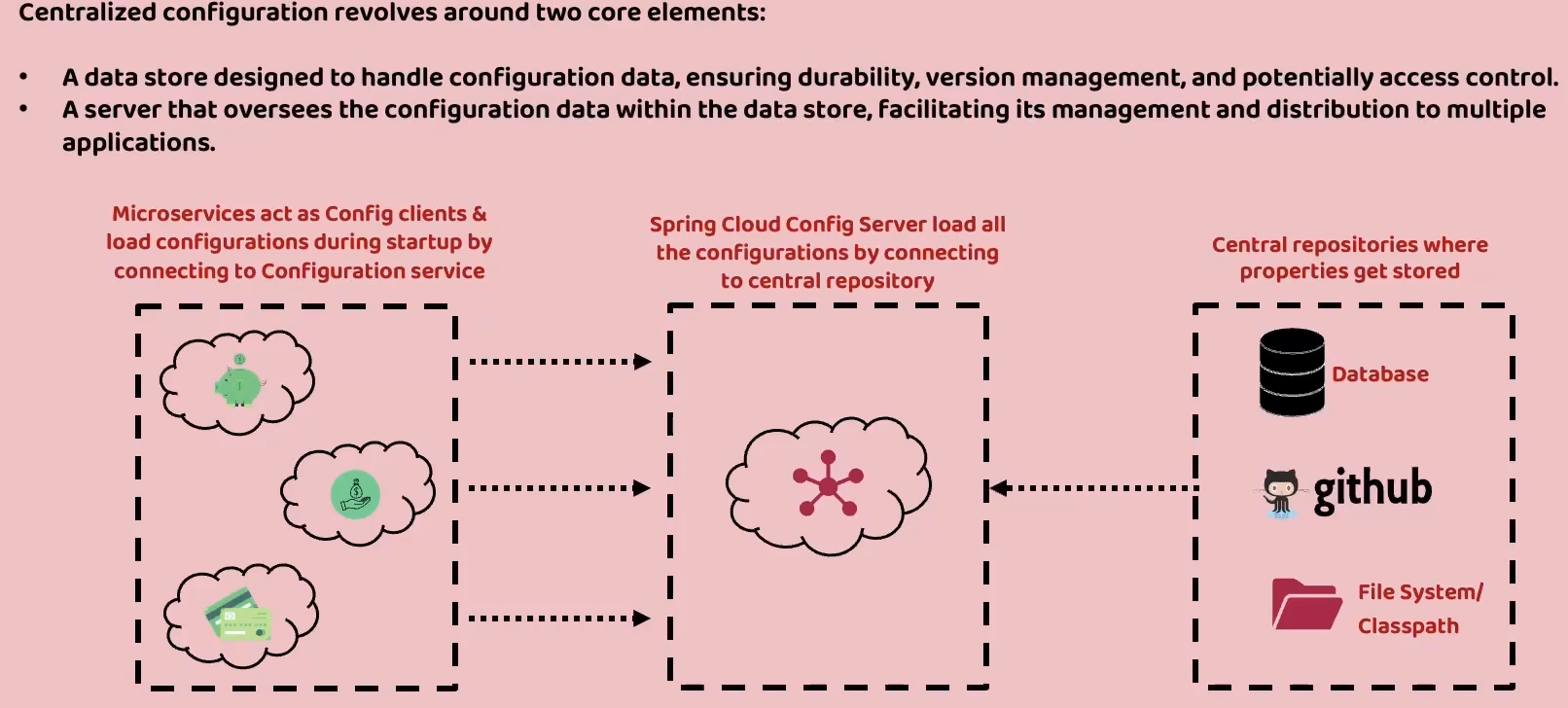
We can also set properties in environment variable . Now to set properties through command line we can set it in the application run configurations

In intellij we can add below to run command



**Introduction to Spring cloud config**

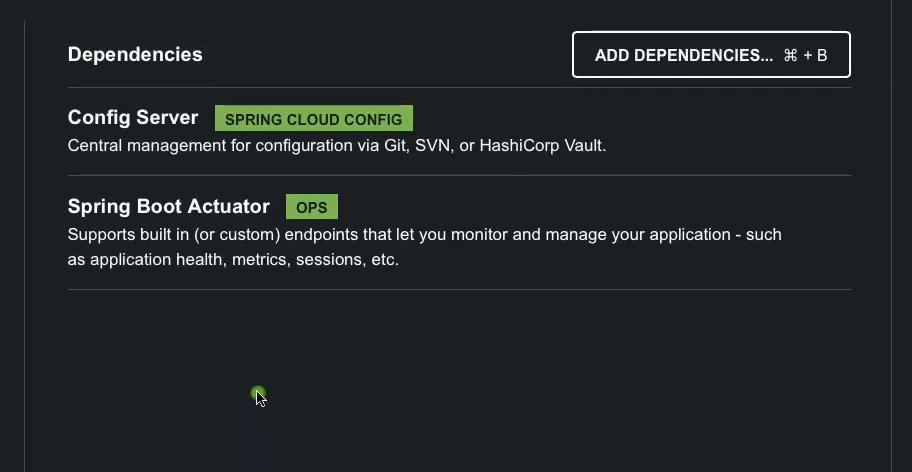
Spring cloud config is a project in the spring ecosystem which helps to create centralized config server. Contains all the properties related to microservices centrally. Centralized repositories can be in file system or github or DB



Lets start creating config server create a spring boot application

We should add the dependency Spring cloud config – Config server

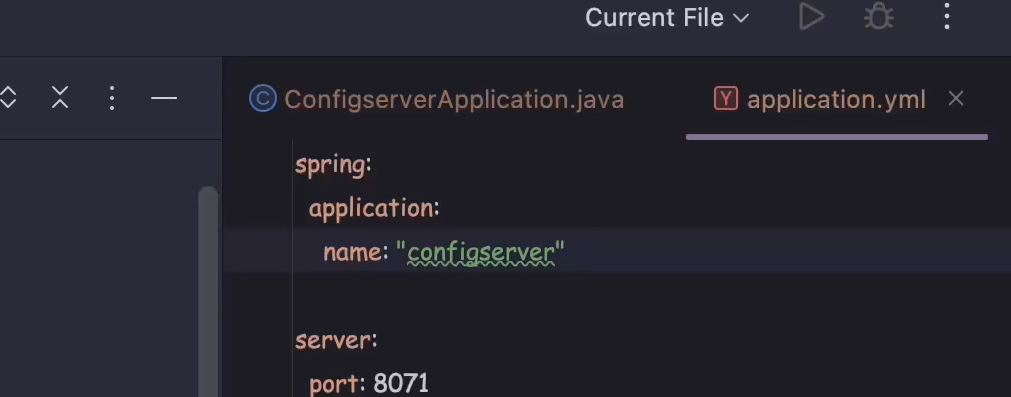
Also we need to add actuator which helps in readiness and liveness which will see during further development



Now once project is created then in the main application file we need to add annotation @EnableConfigServer

Then in the application properties set server port and application name

server.port=8071

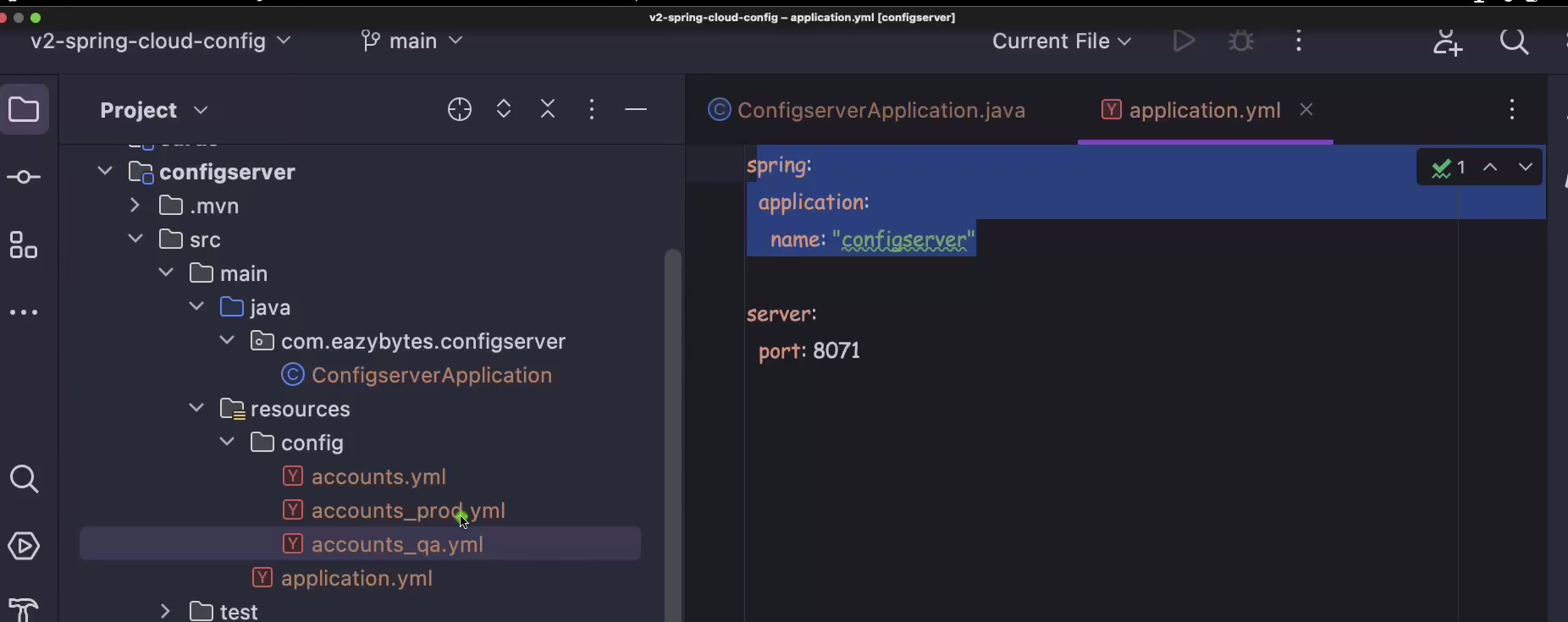


There are various approaches to store the properties in the config server

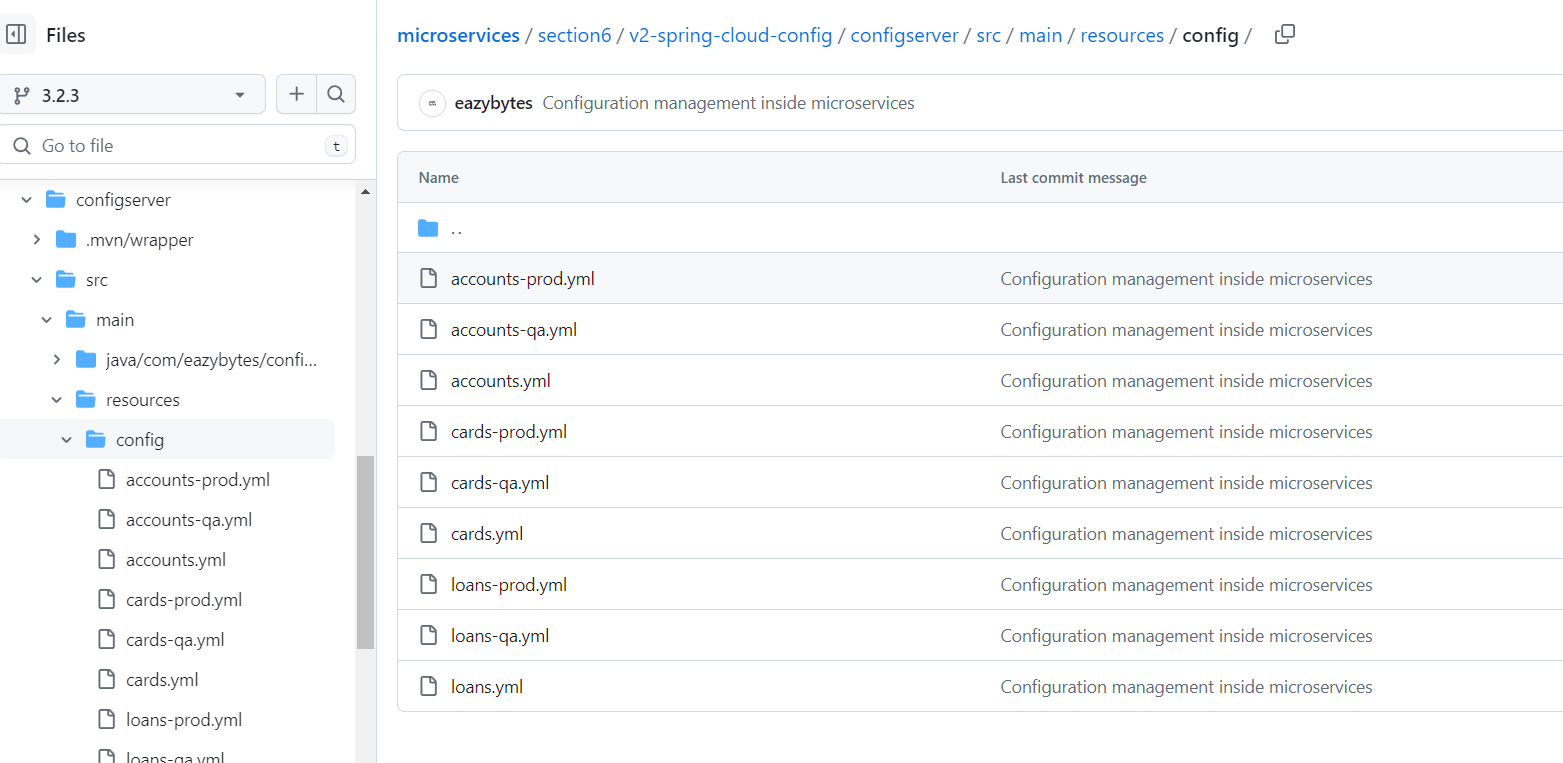
1. Storing in the class path 2) Storing in the file location 3) Storing on to the central repository like git hub

**Storing in the class path**

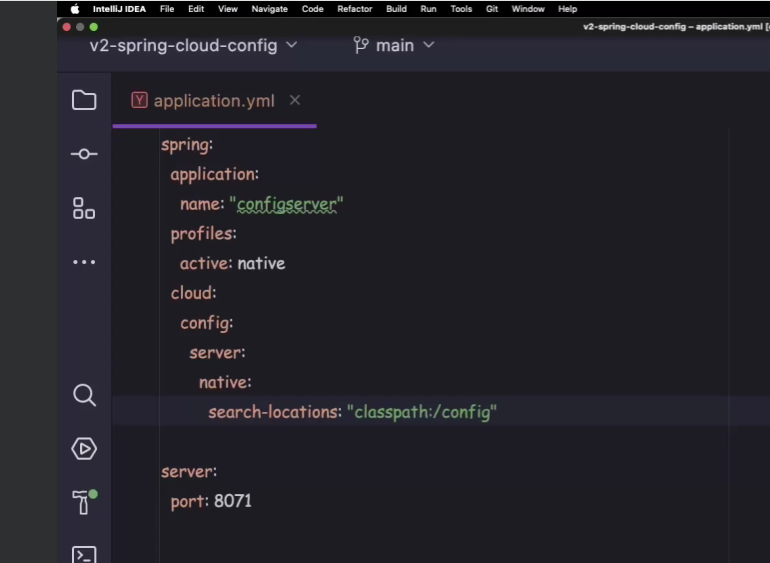
So first will create folder in the resources with name config and copy all the properties file from all the services into it. But we need to rename them to the service name as there will be confusion since all the services will have same properties file names. The service name should be same name as specified in property file service.application.name = accounts



Please note we should use hyphen (-) instead of underscore ( \_ ) as below



Once all the properties files are added in the config server, then we need to add few properties in the application.properties file of config server



We need to set spring.profiles.active=native . native bcoz all the properties are stored in the same config server in resources folder. We should also set class path so that config server can search the locations

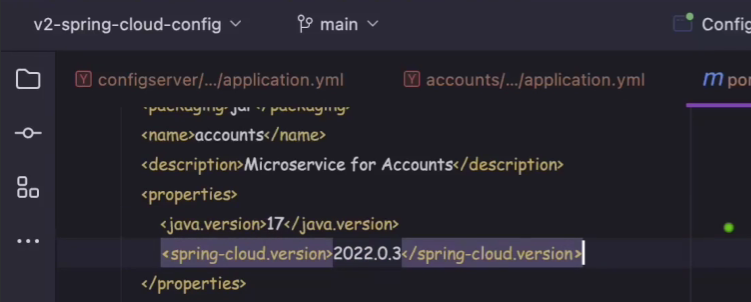
Spring.cloud.config.server.native.search-locations:”classpath:/config”

Config is the folder in resource folder

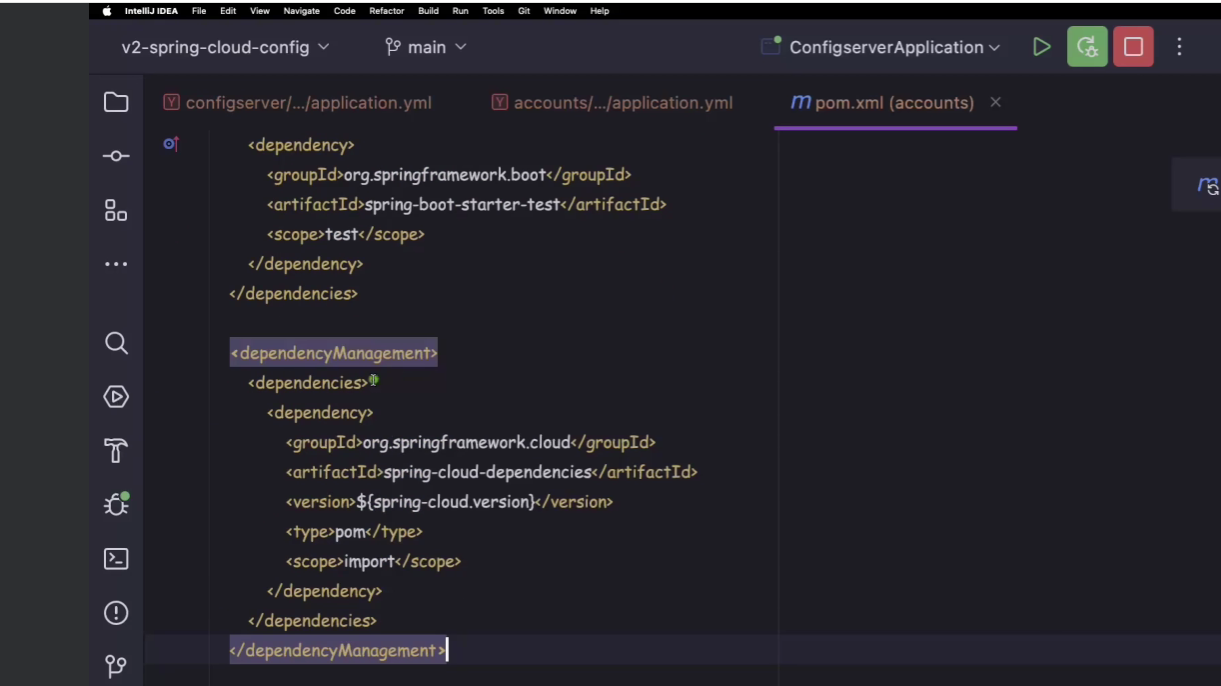
Now lets making changes in microservices so that it can read the properties from the config server

Delete all the prod and qa related property files from services. Then we need to add a dependency spring cloud config – config client.

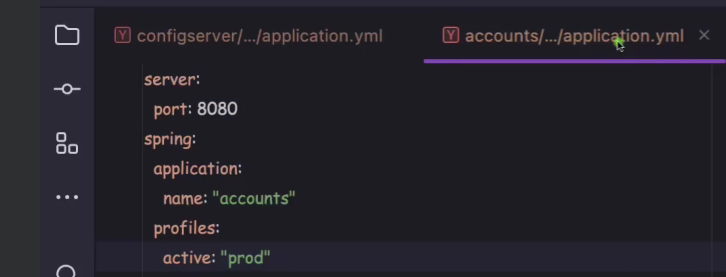
We need to also mention spring cloud version config properties as below



We also need to add dependency management related properties



We need to add property spring.config.import to specify the location of the config server and also added application name and spring.profiles.active

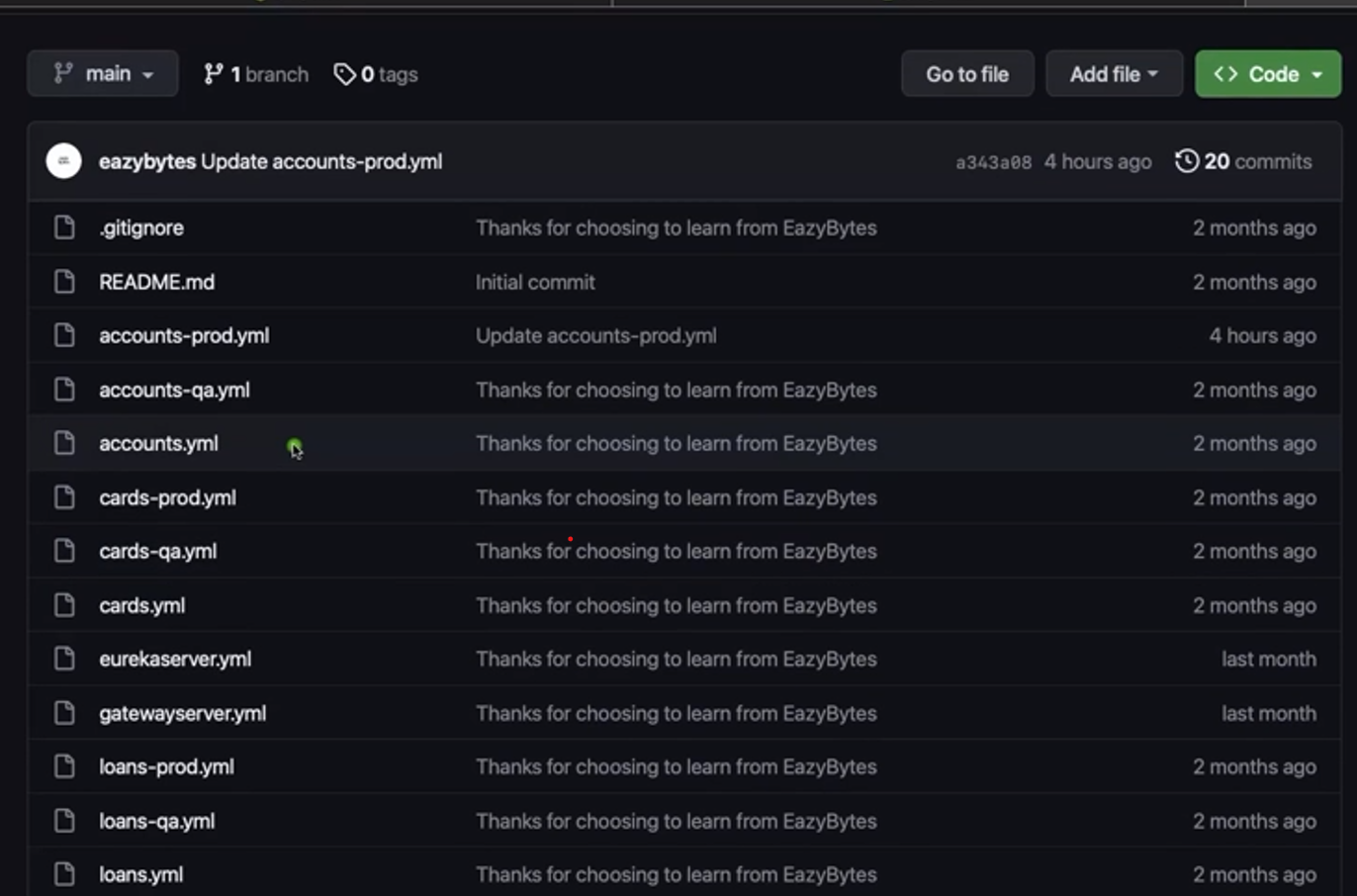




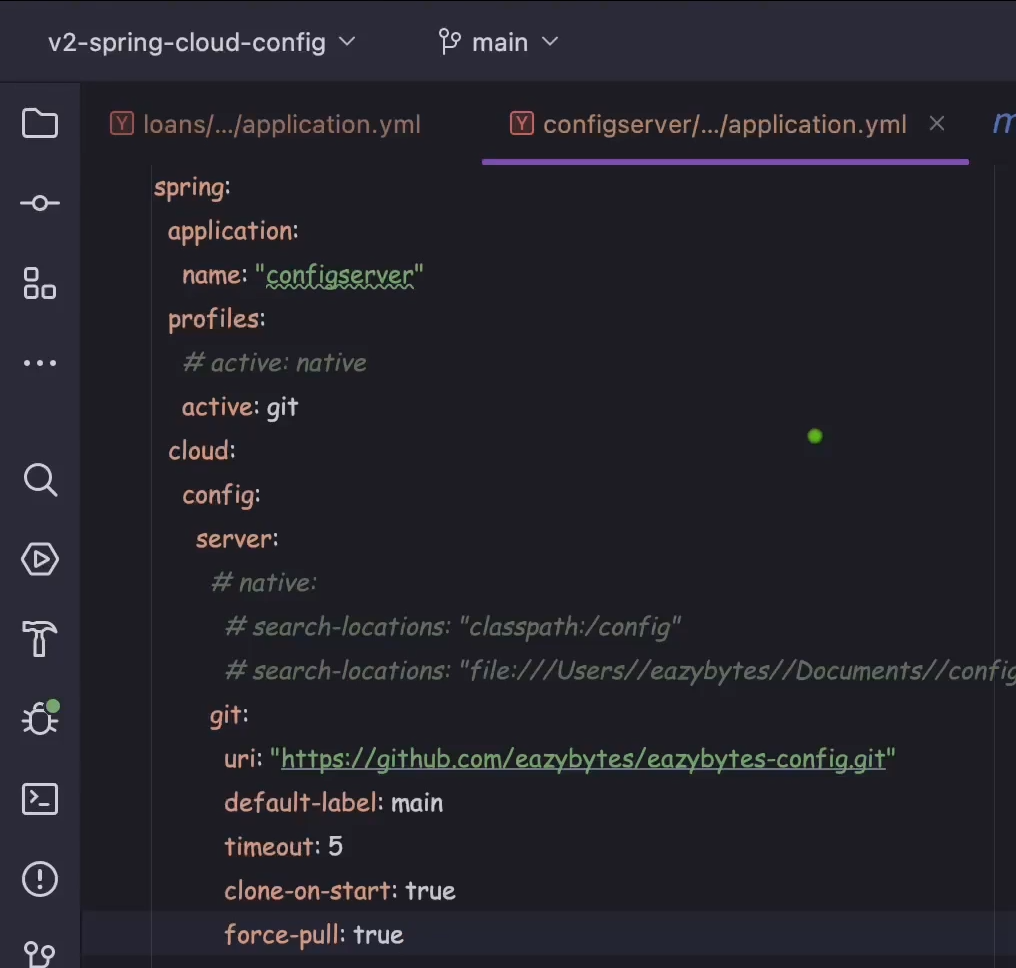
First we need to start config server then followed by all other microservices. After starting then prod related properties will be loaded in the services

**Reading configurations from Git hub repository**

Create a repo on git hub with all the property files of services added as below

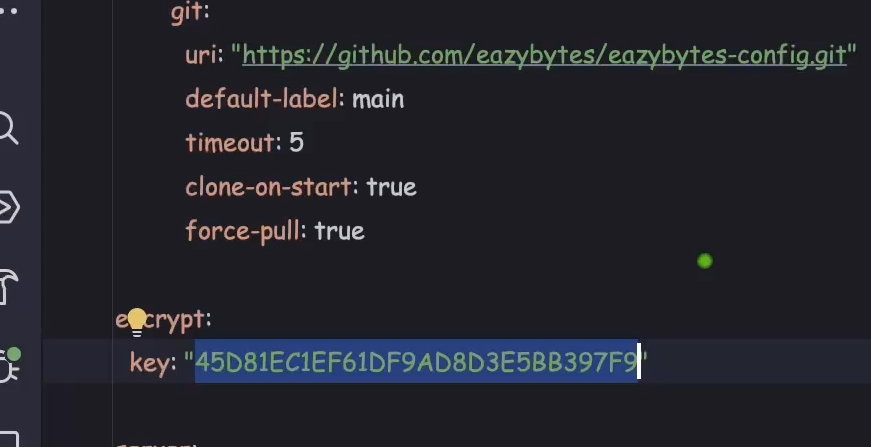


Now we have to change the properties of the config server if we want access property from git hub then profile should be git and then set the git hub repository details and default label is actually the branch name. Also at the startup only we have to clone all the property files so that it doesnot cause any issue later. Force pull should be added so that it ignores all the local changes and after restart it will pull the requests from git hub



Encrytption & Decrypting the properties

We can also encrypt the properties we can add the below properties key can be any value

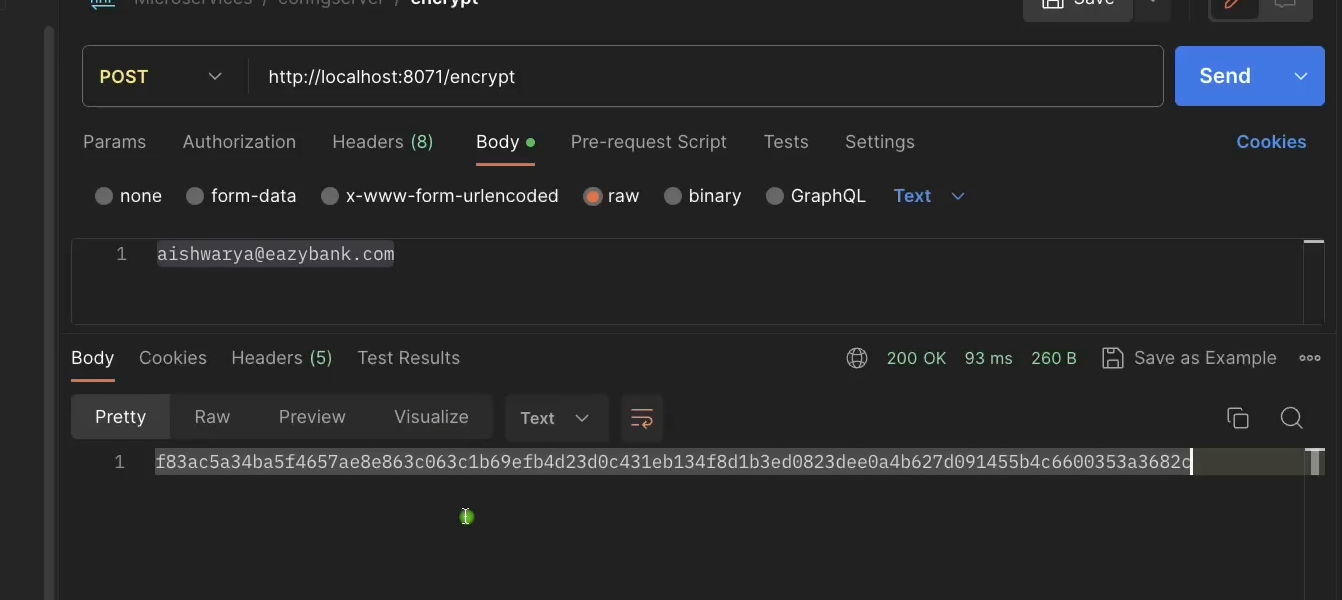


Once you add the encrypt key then our config server is going to expose certain methods through which we can encrypt our data . Now as you can see below that we are accessing encrypt end point to which we are passing some value and it returns the encrypted data

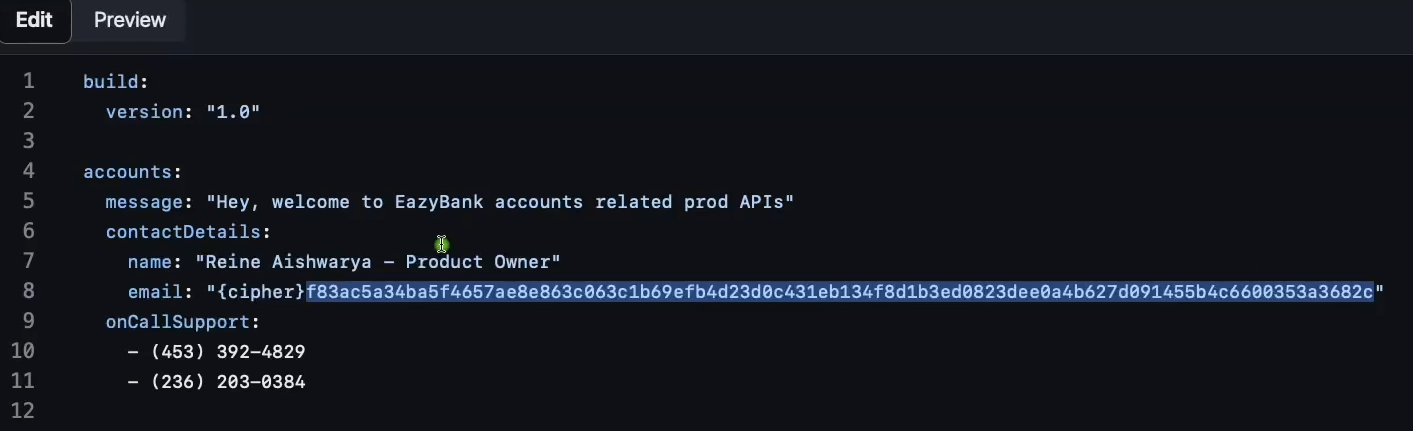
I want to encrypt the email data



To encrypt this data call config server encrypt method its an in built method



Now once the encryption data is generated then we can set this data in property file



We also need to append {cipher} to the encrypted data bcoz spring boot doesnot understand that whether the data is normal data or encrypted data so with cipher it assumes encrypted data and it tries to decrypt when properties are required