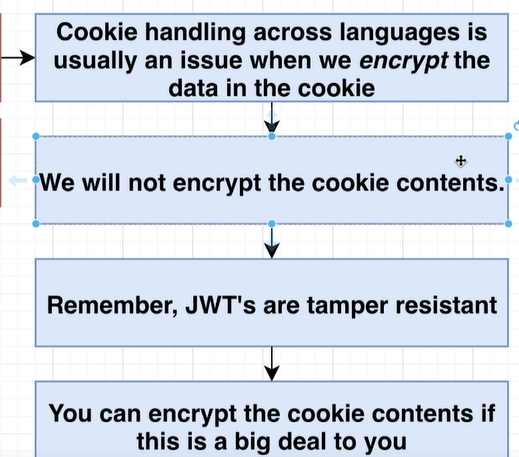
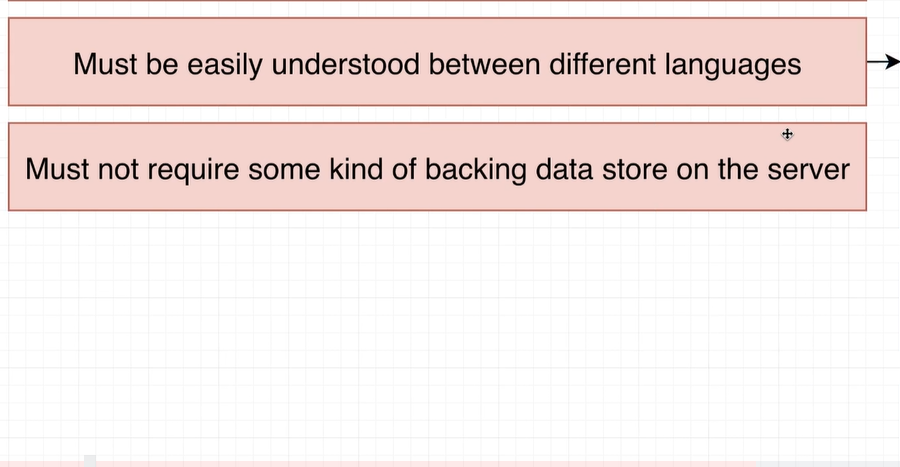
1)cookie-session

<https://www.npmjs.com/package/cookie-session>

we will not encrypt the contents of cookie .



2)wire cookie-service as a middleware to express application

npm install cookie-session @types/cookie-session

express should be aware that it’s behind a proxy of engress-eninx

const app = express();

app.set('trust proxy', true);

app.use(json());

app.use(

    cookieSession({

        signed: false,

        secure: true

    })

);

3)generate JWT

<https://www.npmjs.com/package/jsonwebtoken>

right after we save a user to database we have to generate a JWT and store it to req.session

  //generate JWT

    const userJwt = jwt.sign({

      id: user.id,

      email: user.email

    }, 'asdf');

    //store it on session object

    req.session = {

      jwt: userJwt

    };

    res.status(201).send(user);

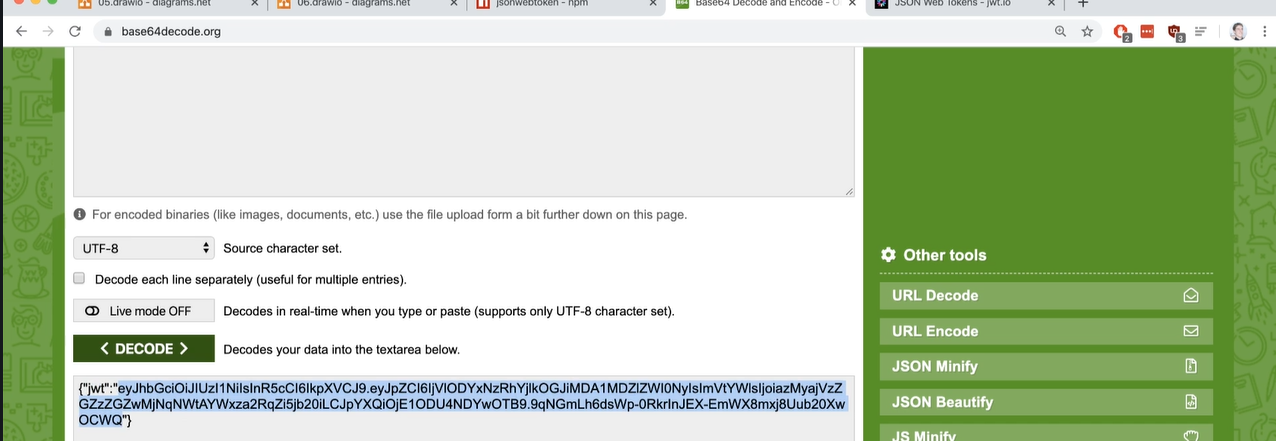
  }

Note: <https://memorymap.dev/api/users/signup>

Make sure you use https: while testing as postman will default to http

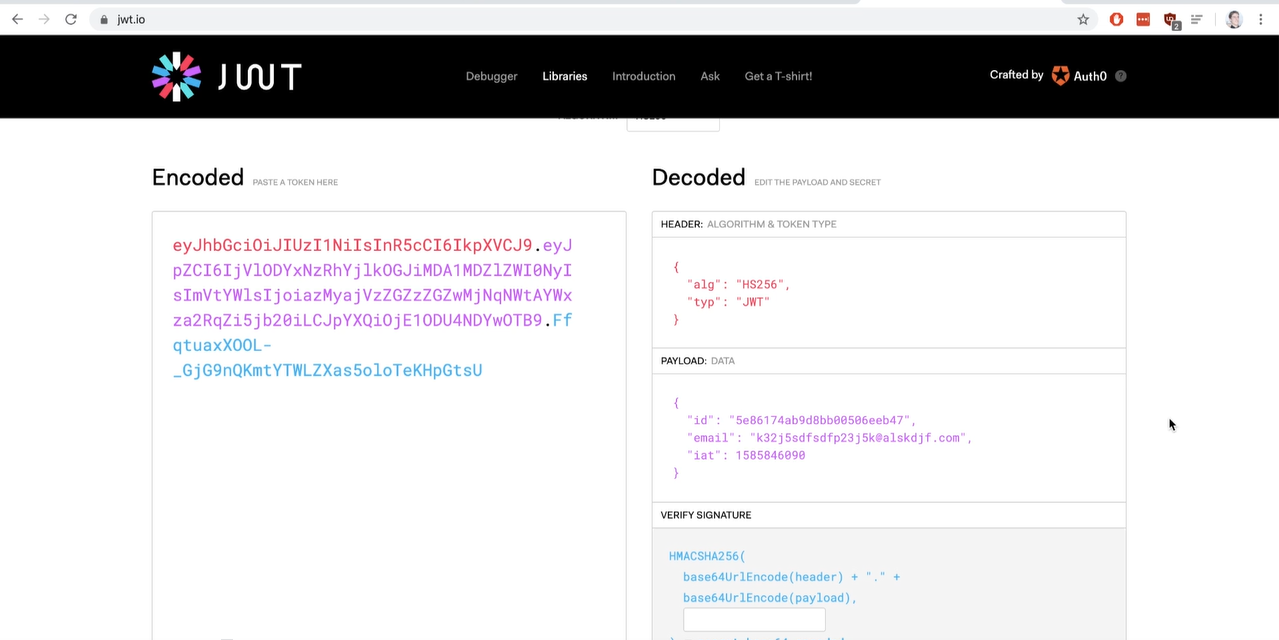
4) 

The jwt is base 64 encoded you can check it by converting it to UTF-8



Goto jwt.io

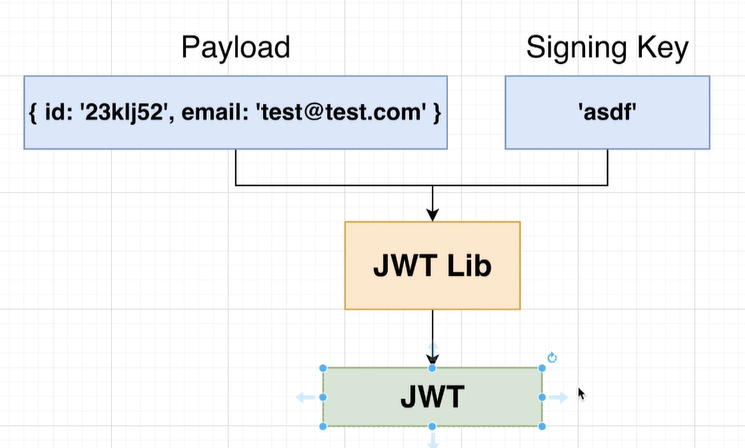
And pasted that jwt with signing key(asfd) to verify the token

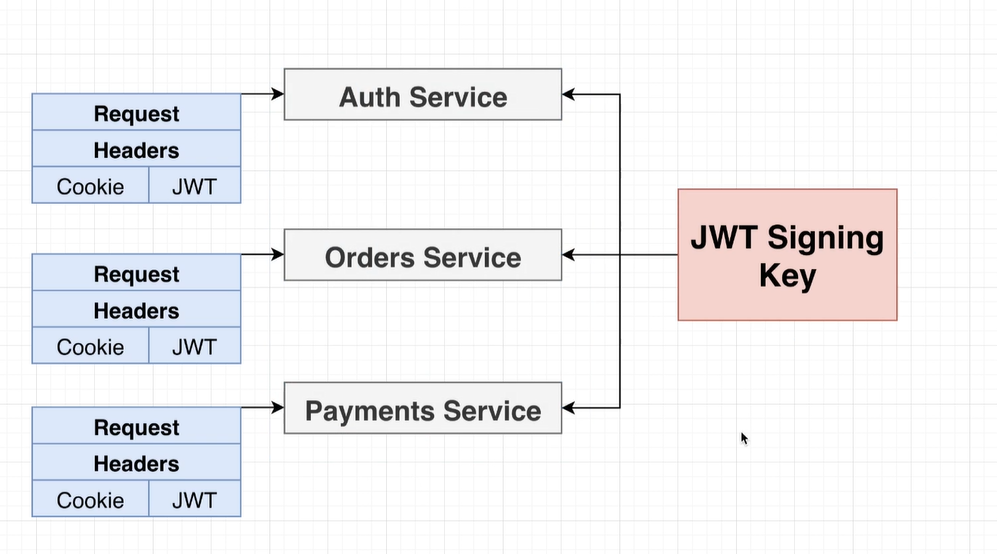


With jwt anyonce can see the information inside of it.

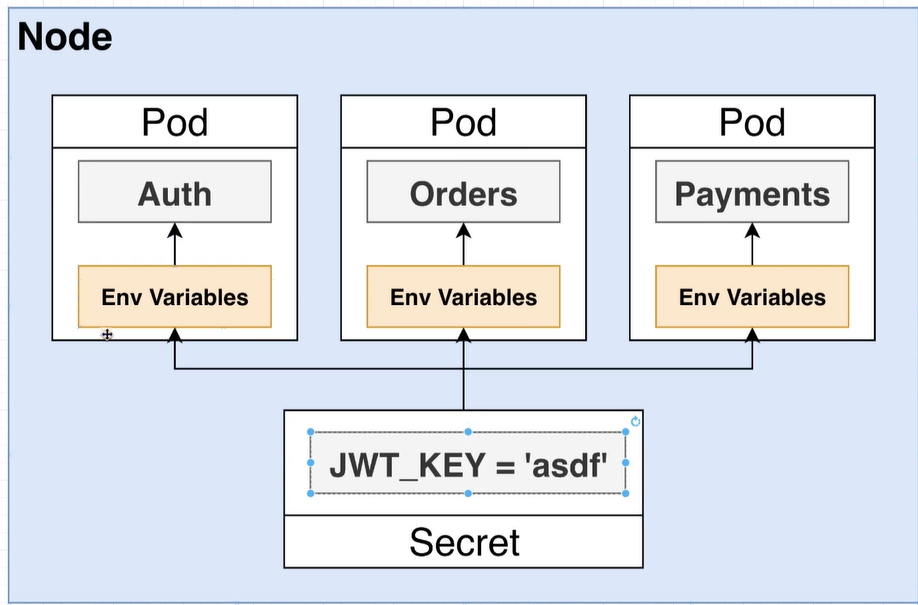


5)Securing Signing Key as it will be shared by other servies so we need to secure it and share it.



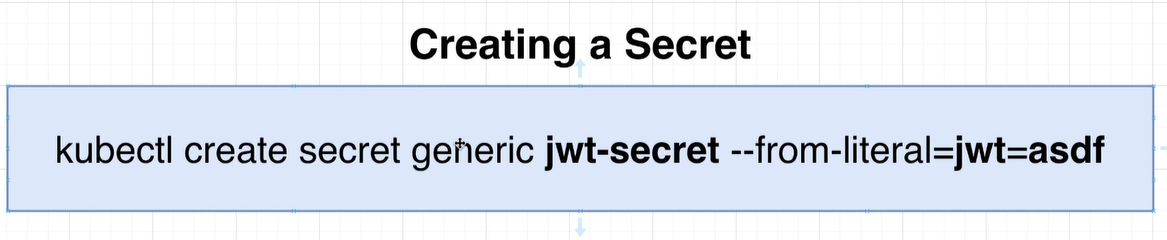


Solution: Figuring out how to store some information and share it with all the services in kubernites



Auth is running inside a container and inside that container there are env variable

We have to assign the key to env variable inside the container.

7) https://www.udemy.com/course/microservices-with-node-js-and-react/learn/lecture/19119820#questions

The reason that we are using this imperative approach right here is because we don't really want to have a config file that lists out the value of our secrets.

Downside of this is we have remember and run all the secret when you spin up a new cluster.

Do not do this for production environment.

* There are different kinds of secrets we can create in kubernities this is an example of a different kind of secret. That is a different kind of secret.We are creating a generic secret which means this is just a all purpose kind of secret piece of information
* Jwt-secret(is the name of the secret).
* This name is very similar to the names we assign to pods or deployments.This is mostly used for logging purposes but in the case of a secret we also use this to refer to inside of a pod config file you'll see an example that very shortly over here on the right hand side we can assign many different properties or many different key value pairs into a secret. In this case we are saying one key value pair of Jason Webb token or JWT equals a asdf.

**6)Creating and accessing secret**

kubectl create secret generic jwt-secret --from-literal=JWT\_KEY=asdf

    spec:

      containers:

        - name: auth

          image: vinkum655/auth

          env:

            - name: JWT\_KEY //name of env variable inside container

              valueFrom: //value for that variable

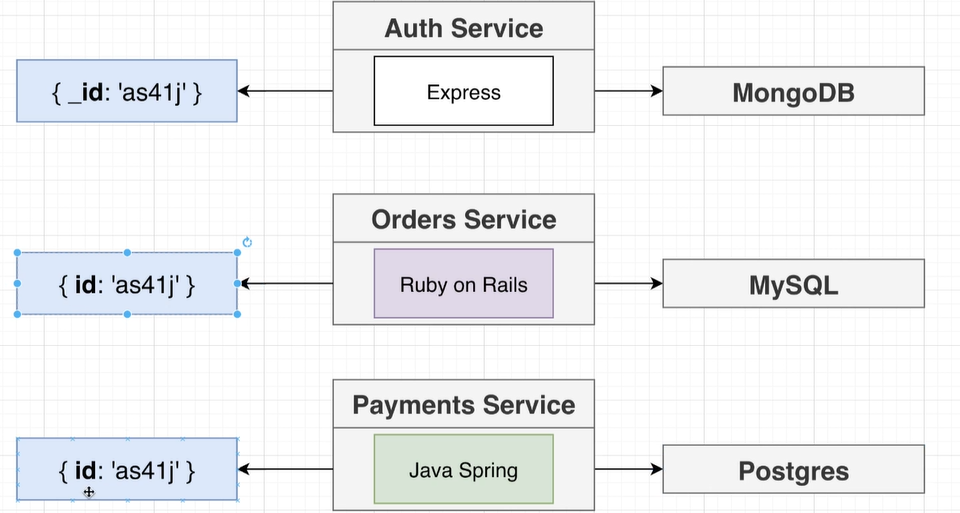
                secretKeyRef:

                  name: jwt-secret

                  key: JWT\_KEY

Accessing env varible inside a pod(change required in to access JS code).

7) we should have consistant response from the db to frontend (eg \_id)

[https://www.udemy.com/course/microservices-with-node-js-and-react/learn/lecture/19119842#questions](https://www.udemy.com/course/microservices-with-node-js-and-react/learn/lecture/19119842#questions#)

it turns our we can change the way js convert object into json

eg const person={name:”vinay”}

JSON.stringify(person)

Const personTow={name:’alex’, toJSON(){return “whaterver you want the json tobe”};}

<https://www.udemy.com/course/microservices-with-node-js-and-react/learn/lecture/19119842#questions>

{

    toJSON: {

        transform(doc, ret) {

            ret.id = ret.\_id;

            delete ret.\_id;

            delete ret.password;

            delete ret.\_\_v;

        }

    }

}); //this could happen at view level