Individual Project

41492 – Engenharia de Software, Nuno Sá Couto e Rafael Direito
October 19th 2024

Progress Tracking Questions

(Please raise your hands)

- Have you already thought about the functionality your To Do List Web Application will offer?
- Have you written any user story that maps the various features you aim to offer through your app?
- Have you already started implementing the UI / API?
- Have you already integrated Cognito in your solution?
- Have you already decided which AWS services you will use?
- Have you already deployed parts of your solution in AWS?

Let’s start from the ground zero...

- Configure a VPC with
    - 2 Azs
    - 1 Public Network per AZ
    - 1 Private Network per Az
- Deploy an EC2 Instance to host a Website/API (you may use the User Data script provided in last class)
    - For now, this VM can be public (just during dev.)
- Create a Target group to point to the EC2 Instance
- Create an ALB that forwards the requests to this Target Group
- Access the Website/API through the ALB

Let’s start from the ground zero…

- If you are able to do this, then you can:
  - Host your website in a EC2 Instance (there may be better approaches)
  - Host your API in a EC2 Instance (there may be better approaches)
  - Use ALBs to expose the API and the UI
  - Integrate the API GW with the API ALB (there are other approaches)
  - Integrate CloudFront with the Web ALB (there are other approaches)

This deployment scenario is just a very simple example
There are N other approaches to achieve the same
You should use the services that most suit you

Let’s start from the ground zero…

- Services you may use to host your Web UI
  - Amazon S3 (Static Website)
  - Amazon EC2 (Full Stack Apps + Static Websites)
  - AWS ECS (Full Stack Apps + Static Websites)
  - AWS Elastic Beanstalk (Full Stack Apps)
  - Amazon Lightsail (Small Websites and Applications)
  - AWS Amplify (Full Stack Web Apps – e.g. React, Vue, Angular…)
  - AWS CloudFront (offers an HTTPS endpoint and can forward traffic to the previous Services)

“Endless” Approaches

You should check which services you can use in AWS Academy

Important aspects to consider

Cookies and Headers

- If your solution relies on cookie exchanges or on specific headers you must be very careful when configuring CloudFront, ALBs, API GWs, etc. For security reasons, AWS's default behavior is to block cookies and ask you to whitelist the headers that should be sent/received

Cognito Redirects

- For Cognito to work, the redirects must be achieved with TLS/SSL. This means your solution must be offered through HTTPS. Please address this ASAP, as it may be complex to deploy an HTTPS protected solution in AWS.

Logs:

- This must be one of the first thing for you to configure. You should start by thinking of how you will collect the logs. CloudWatch is very easy to use. Logs are extremely important for you to debug your deployments. Example: If an ECS Task fails, how will you know why it failed?

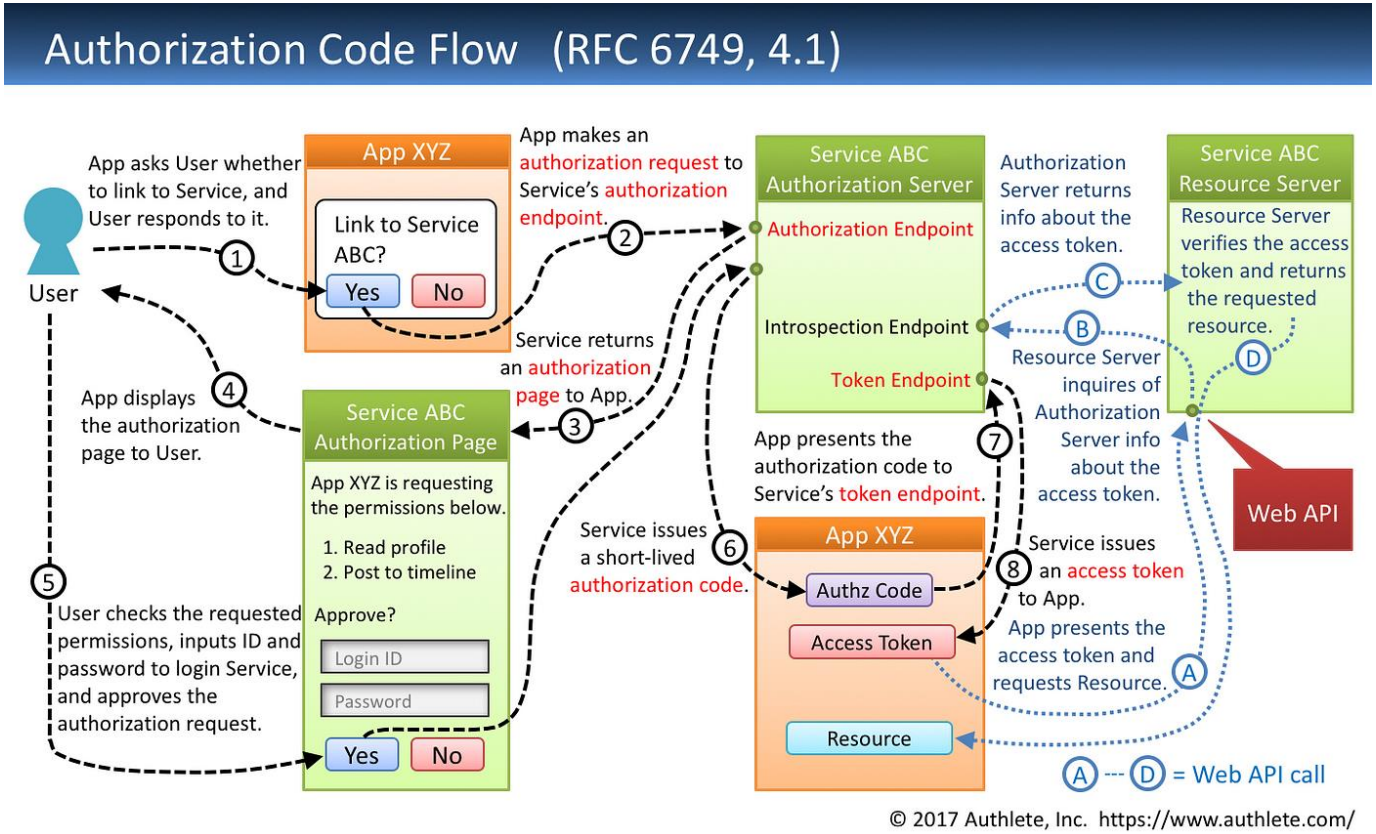
Important aspects to consider

Automation:

- In the individual project no automation is required. You can set up everything by hand in your Learners Lab environment. Don’t overcomplicate things! (It is better to have a manually configured working application, than an automated mess)

Important aspects to consider

OAuth2:



Client Secret cannot be exposed to the client

One way to achieve this is to use the REST API as a Proxy

Demo

HTTPS:

- How to achieve “HTTPS” on a ALB?

Certificates and Instructions will be provided on-demand.
You must ask for these artifacts via e-mail (rafael.neves.direito@ua.pt)

Questions about the Individual Project?