Cloud CI/CD Pipelines
How to easily deploy in the cloud

Sereno Balestra
AWS DevOps

Session Agenda

In this session we'll go through the following points:

- Just a few technical words definitions, like
  DevOps, CI/CD, Build, Deploy, Pipeline …
- Managed builds in the AWS Cloud: CodeBuild
- CodeDeploy and deployment strategies
- CodePipeline
- Advantages of using cloud CI/CD services
- DEMO: simple application end-to-end build and deployment

Some quick definitions

- **DevOps** => is an IT working methodology. It combines the work of software development and infrastructure operations to improve the development lifecycle.

- **Build** => is the act of transforming the source code into a final package (artifact) ready to be installed, of course after doing code checks and tests.

- **Deploy** => release the application into its final environment, e.g. update server with new app version

- **CI/CD** => core principle of DevOps. (Continuous Integration / Continuous deployment). The code is automatically tested,packaged and then released

- **Pipeline** => ordered set of steps, each containing different tasks with commands, to accomplish a goal, e.g. a pipeline to test&build the software and then release it

AWS CI/CD Services

CodeBuild

CodeBuild is an AWS fully managed service which lets you run your build steps in the cloud.

Build steps will be run on VMs or containers and they are billed per running minute.

The steps can be defined in a simple YAML file
The build outputs will be stored in S3

Build logs can be watched in real time from UI.

It can be used to run whichever command/process, you can provide your own container image.

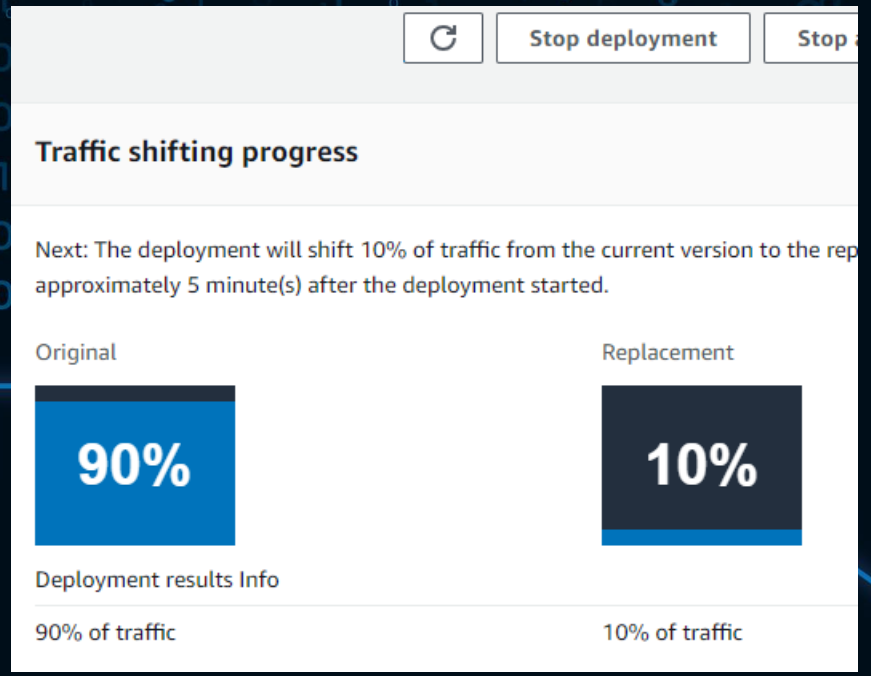
```yaml
version: 0.2

phases:
  pre_build:
    commands:
      - echo Installing source NPM dependencies...
      - cd src
      - npm install
  build:
    commands:
      - echo Build started on `date`
      - echo "Linting code, Running unit & integration tests"
      - echo "Running dependency scan, Running code quality scan"
      #this is just a demo, so none of the above steps is actually done here.
      #Please ALWAYS consider to add those to your CI pipeline!
  post_build:
    commands:
      - echo Build completed on `date`
artifacts:
  files:
    - main.js
    - index.html
    - node_modules/**/*
  base-directory: 'src'
```

CodeDeploy and deployment strategies

CodeDeploy is an AWS service that takes care of releasing application onto different systems.
It integrates directly with other AWS services (Lambda,EC2…), but it can also manage non-AWS systems.
CodeDeploy can be configured to automatically roll back the deployment based on CloudWatch alerts triggers.

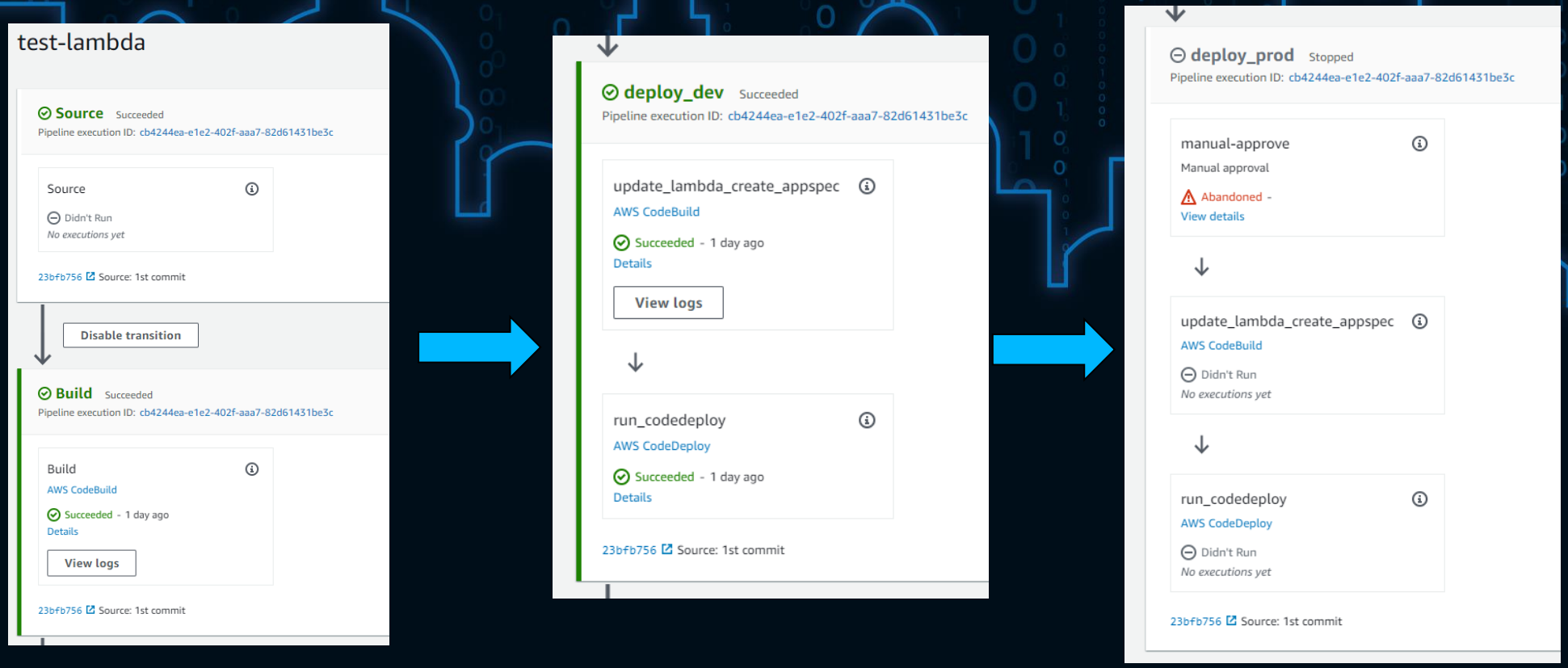
There are different deployment strategies that can vary according to the target platform, for AWS Lambda we've got the following:

- Normal deploy => all the traffic is switched to the new version right away
- Canary strategy => only a small share of traffic is switched to the new version for some time
- Linear strategy => traffic is switched to the new version in steps e.g. 10% every 10minutes



CodePipeline

CodePipeline is a fully managed AWS service that lets you create pipelines in AWS.
Pipelines are composed by stages, each stage can have one or more actions groups and within an action group you can have multiple parallel tasks.
Actions can leverage many AWS services (CodeBuild,CodeDeploy,Lambda) and can also integrate with other systems.



Advantages of Cloud CI/CD services adoption

- No servers management / manual installs
- You only pay for what you use
- Direct integration with many cloud services
- Secure least-privilege pipelines with cloud IAM permissions
- Easy to configure
- Built-in monitoring metrics/logging
- Simple notifications for events
- Out of the box GIT integrations

LAB DEMO
Let’s deploy a simple app with cloud CI/CD

AWS Services

For our demo we'll use the following AWS services:

- API Gateway => service to expose REST/http APIs to the internet

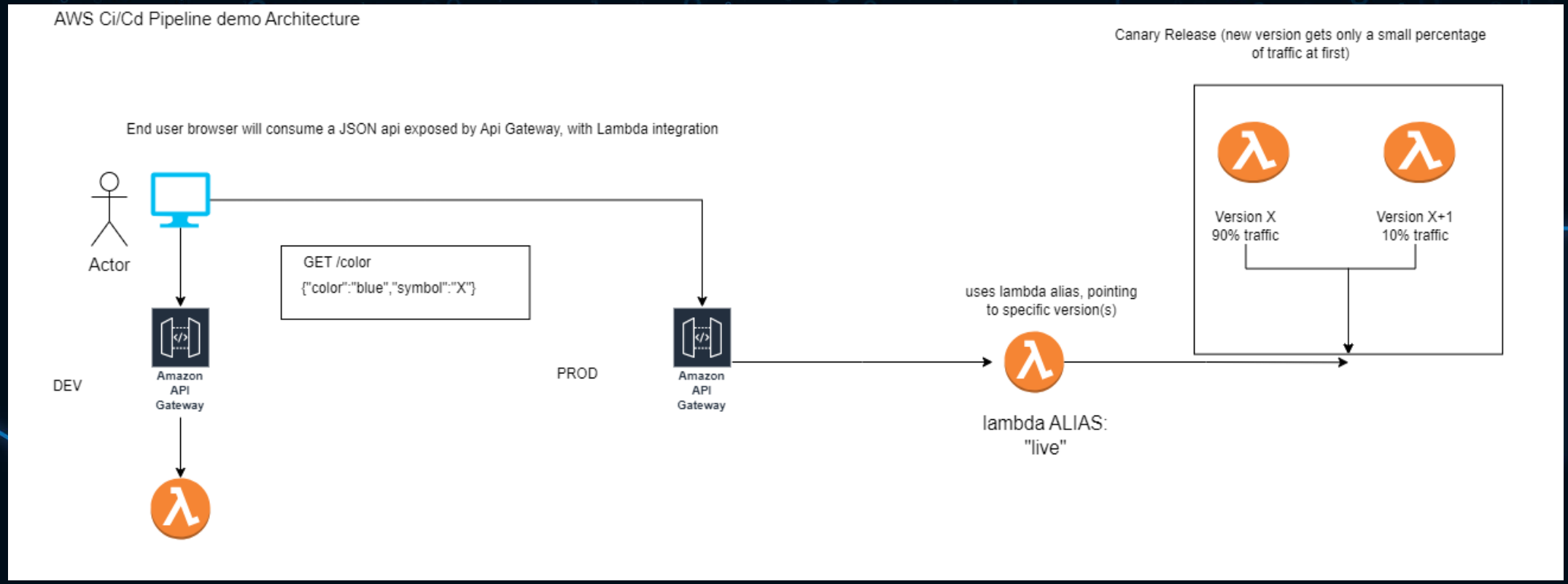
- Lambda Function => AWS serverless functions, you simply upload your code and AWS takes care of everything

- S3 => fully managed object storage

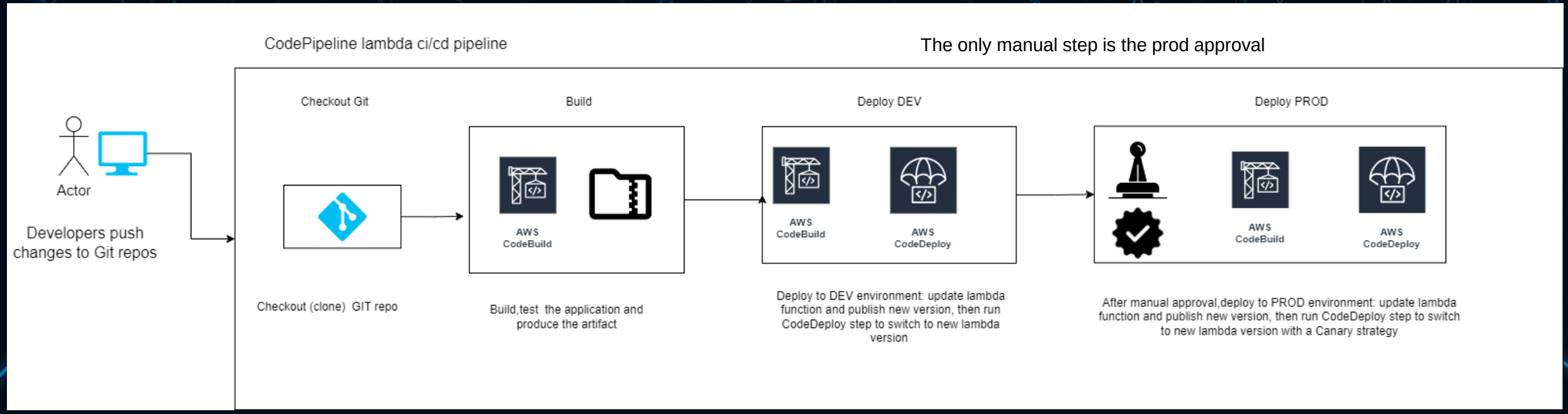
- CodePipeline, CodeBuild, CodeDeploy => services to fulfill CI/CD by running pipelines in the cloud

- CloudWatch Metrics and Logs => to monitor all cloud components (CI/CD pipelines,lambda functions..)

Lab Demo: Application Architecture



Lab Demo: CI/CD Pipeline

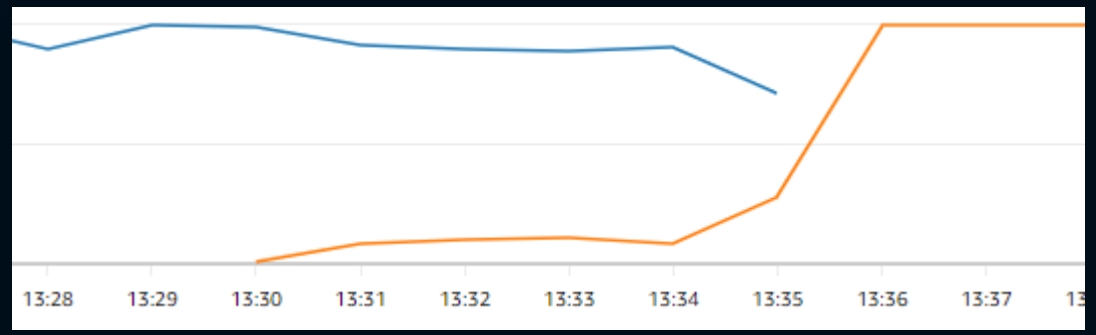


Monitoring

CodeBuild ships many metrics that let us keep track of the build statuses and duration of the different steps.

Each CodePipeline element (whole pipeline,stages or actions) send events that can be used to trigger notifications.

The applications can be checked during deployment by looking at the specific service metrics, in our case for Lambda we can check (among many other) the invocations,errors and duration.



Lambda invocations during deployment, we can see the 2 versions

Session Recap

Today we've explored the AWS Cloud CI/CD world, with a simple application built and deployed with CodePipeline to Lambda for 2 environments: DEV and PROD, in a fully automated way and with no downtime.

In PROD environment we applied the Canary release strategy, to send only a small amount of traffic to test the new version and shown this grafically from the UI.

We've gone through the services used, what they do and how they interact.
Finally, we've learned how to monitor our pipelines & applications.

Q&A time

(For any further question, feel free to use my contact details)

THANKS!

sbalestra@sorint.it
+39 340 953 82 37