



Getting Started with CI/CD on **AWS**

AWS Cloud Support Team
2021

Housekeeping

:+1: 👍 :-1: 👎 :raising_hand: 🙋



© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.



Day 2

Day 2 - Agenda

- What have we learnt?
- Orchestration
- Infrastructure as Code
- Artifact Repositories

Recap from yesterday

Review

Agenda

Questions

1. What is Continuous Integration?

Review

Agenda

Questions

1. What is Continuous Integration?
2. What is Continuous Deployment?

Review

Agenda

Questions

1. What is Continuous Integration?
2. What is Continuous Deployment?
3. What is Continuous Delivery?

Review

Agenda

Questions

1. What is Continuous Integration?
2. What is Continuous Deployment?
3. What is Continuous Delivery?
4. What are the advantages of these principles?

Are you ready?

Orchestration

So now we know the individual components.
How are we able to put it all together?

Orchestration

So now we know the individual components.

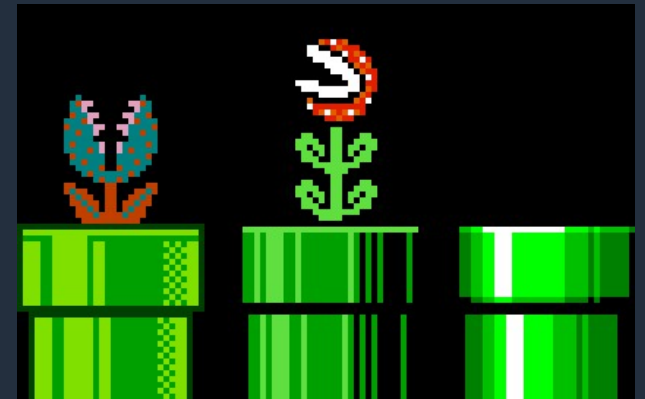
How are we able to put it all together?

The process of delivering and deploying software used to take days or even weeks to accomplish, rather than hours or minutes. Continuous delivery requires a greater emphasis on orchestration, using automation processes for the delivery of software updates.

Pipelines allow the automation of the different stages in the software lifecycle namely the build, test, and deploy phases. This can then be repeated in the release process every time there is a code change, based on the release model you define. This enables you to rapidly and reliably deliver features and updates.

Having a Pipeline in place allows:

- Automation
- Efficiency
- Improved Cross Department Communication
- Cost Savings
- Shorter Time to Market



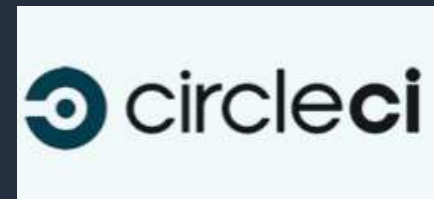
Orchestration

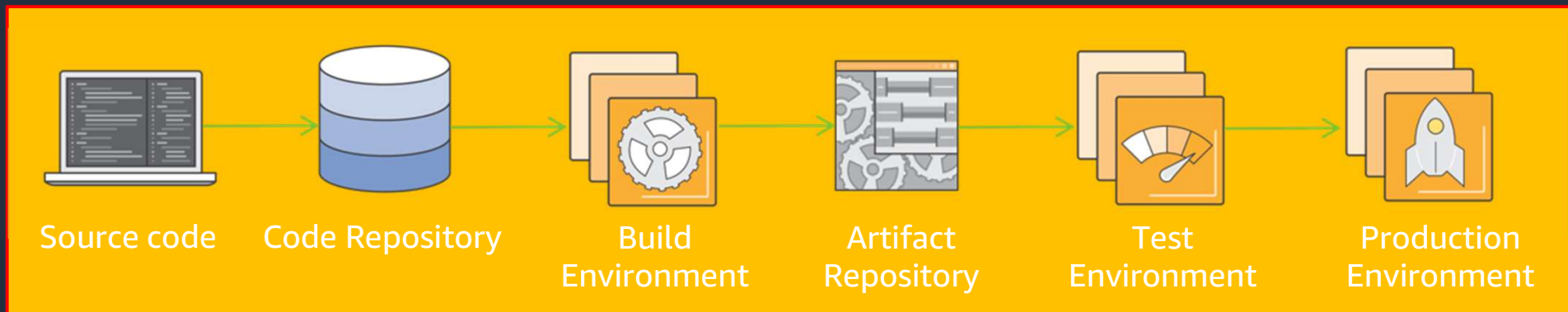
AWS CodePipeline



- Automate your release processes
- Establish a consistent release process
- Speed up delivery while improving quality
- Incorporates with your existing source, build, and deployment tools IDs.

Third-party Pipeline Tools





Lab 4 CodePipeline:

We will now create a continuous delivery pipeline using AWS CodePipeline

Outcomes:

- Fetch source code from GitHub
- Build an application
- Deploy the application to our AWS ElasticBeanstalk QA environment
- Deploy the application to our AWS ElasticBeanstalk Production environment after manual approval

https://github.com/pedreviljoen/aws-vls-cpt-ci_cd_workshop/blob/main/labs/04-pipeline/README.md

Before we move on, any questions?

Let's take a break while it runs
(we will be back in 10 minutes)

Infrastructure as Code

Infrastructure as Code Defined:

Infrastructure as code is the process of managing and provisioning computer data centers through machine-readable definition files, rather than physical hardware configuration or interactive configuration tools.

A simpler version:

Infrastructure as Code means to manage your IT infrastructure using configuration files.

Infrastructure as Code

Why would you want to do that?

Obvious example is we have been doing with you is allows repeatability, your code becomes the single source of truth.

Benefits:

- Speed
- Consistency
- Accountability
- More Efficiency During the Whole Software Development Cycle

Introduction to CI/CD

AWS CloudFormation



- Automate best practices
- Scale your infrastructure worldwide
- Integrate with other AWS services
- Manage third-party and private resources

Third-party Infrastructure as Code Tools



Lab 5 CloudFormation

This lab will provide a sample of Infrastructure as Code

Outcomes:

- Gain an Understanding of how AWS CloudFormation is able to launch resources in a repeatable manner.
- We'll also be using the created Resources Later.

https://github.com/pedreviljoen/aws-vls-cpt-ci_cd_workshop/tree/main/labs/05-infra-as-code

```
{ "AWSTemplateFormatVersion" : "2010-09-09",
  "Description" : "AWS CloudFormation Sample Template",
  "Parameters" : {
    "ReadCapacityUnits" : { .. },
    "WriteCapacityUnits" : { .. }
  },
  "Resources" : {
    "myDynamoDBTable" : {
      "Type" : "AWS::DynamoDB::Table",
      "Properties" : { .. }
    }
  },
  "Outputs" : {
    "TableName" : {
      "Value" : {"Ref" : "myDynamoDBTable"},
      "Description" : "Table name of the newly created DynamoDB table" }
```

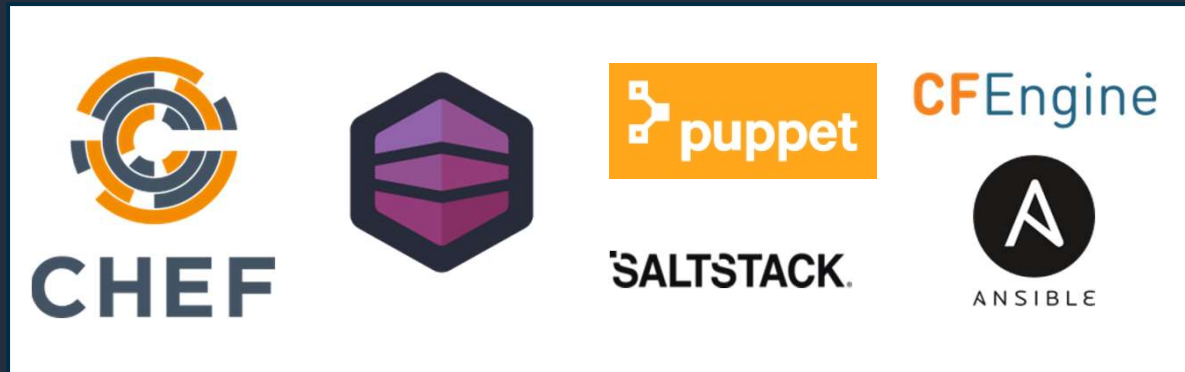
Configuration Management

Infrastructure as code can also be seen as a management tool that ensures that all systems in an environment are configured in the same way and have the same patches applied to them

This can include OS Patches or even application patches.

These tools fall under Configuration Management and can be seen as an extension of Infrastructure as code.

Config management tools:



Deployment Services

AWS CodeDeploy

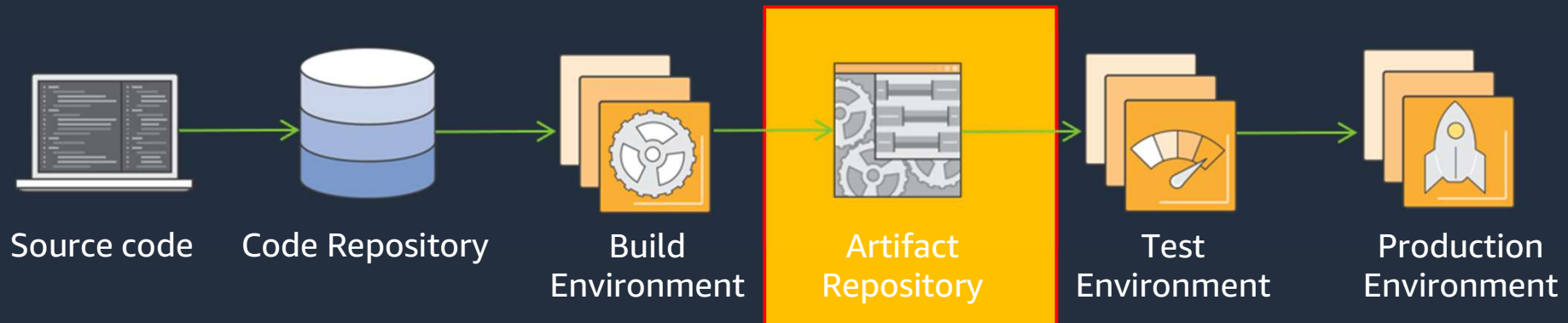


- Automated Deployments
- Centralized Control
- Minimized Downtime
- Easy to Adopt

AWS CodeDeploy vs AWS Elastic Beanstalk

- CodeDeploy allows greater customization in automation.
- Elastic Beanstalk has preconfigured environments.
- CodeDeploy allows On Prem deployments
- Elastic Beanstalk has a custom CLI allowing easier deployments from terminal and IDE integrations.

Before we move on, any questions?



AWS CodeArtifact



Benefits:

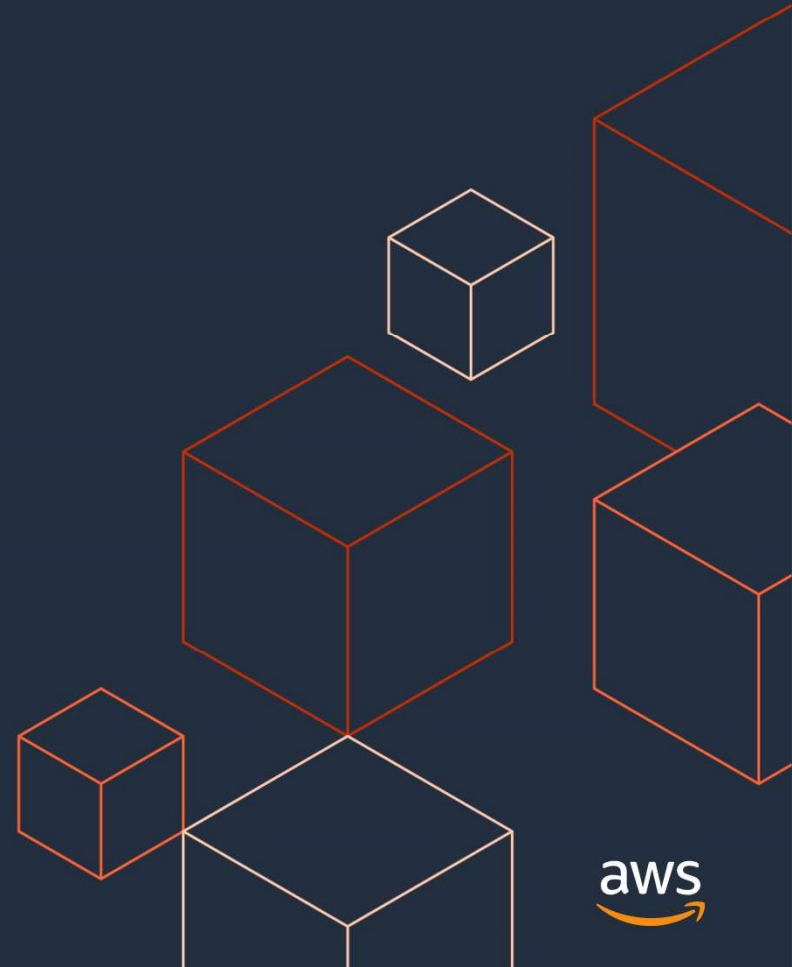
- Dependency management
- Efficient builds
- Release stability
- Audit

Artifact management tools, also known as artifact repositories, are used to store, organize and distribute artifacts (that is, binary files plus their metadata) in a single centralized location. This reduces the amount of time spent downloading dependencies from a public place. Artifact management tools also prevent inconsistencies by allowing development teams to find the right version of an artifact easily.

Before we move on, any questions?

End Of Day 2

Q&A



Thank you!