

中村勉強会  
～AWSインフラ編～  
「Codeサービス系 #1」

## この勉強会の意図

- ・ 施設予約システムのインフラ全般を担当
- ・ 特に悩んだのがコンテナサービスのCI/CDとインフラIaC化
- ・ 約半年やってきたことの整理と生まれた知見の共有  
→インフラ構築・運用属人化を防ぐ
- ・ 今後の課題整理

あとついでに目標の達成!!!

# 本日のお題目

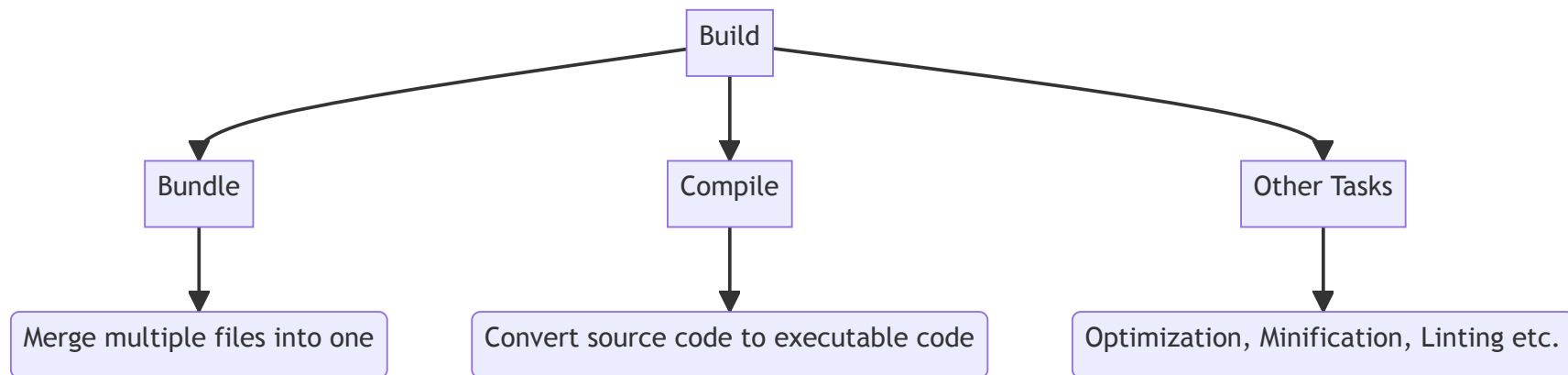
1. CI/CDの重要性
2. 実現手段と選定理由
3. CI/CDに関わるCode〇〇サービス

# 1.CI/CDの重要性

## CI/CDとは？

### ①CI(Continuous Integration)

→ コードの変更を継続的に統合する過程(ビルド、テスト)

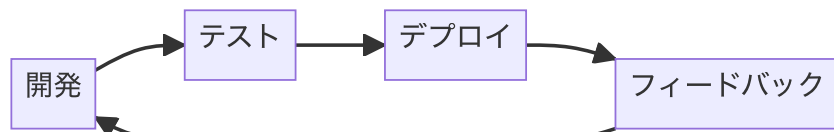


### ②CD(Continuous Deployment/Delivery)

→ 統合されたコードを実環境に継続的に反映させる過程(デプロイ)

# 1.CI/CDの重要性

## 高速なフィードバックループの実現



- ①アプリケーションの品質安定
- ②手動運用時に起こりうるヒューマンエラーの削減

だけではなく...

# 1.CI/CDの重要性

もう一つの幸せスパイラルを生む



## 2.実現手段と選定理由

実はCI/CDツールはたくさんある

→CircleCI,Jenkins,GitLab,GitHub Actions, Azure,GCP系サービス

結論:AWSのサービス群を選択

### 1.信頼と実績

- ・すでにプロトが用意してあって、最低限のCI構築はできていた
- ・CodeDeployだけ一応使ったことがあった

### 2.応用可能性/親和性

- ・その他インフラリソースとの親和性
- ・IaCサービス(AWS Cloud Development Kit)との親和性

### 3.学習コスト

- ・学習コストのことだけ考えれば、基本的にサードパーティ製のツールはない方がいい

## 3.CI/CDに関わるCode〇〇サービス

1. CodePipeline...パイプライン(流れ)を定義
  2. CodeCommit/Artifact...GitHubのAWS版、ソースストレージ
  3. CodeBuild...ビルドのための環境を素早く用意、ビルドプロセスの構築
  4. CodeDeploy...デプロイ
- 

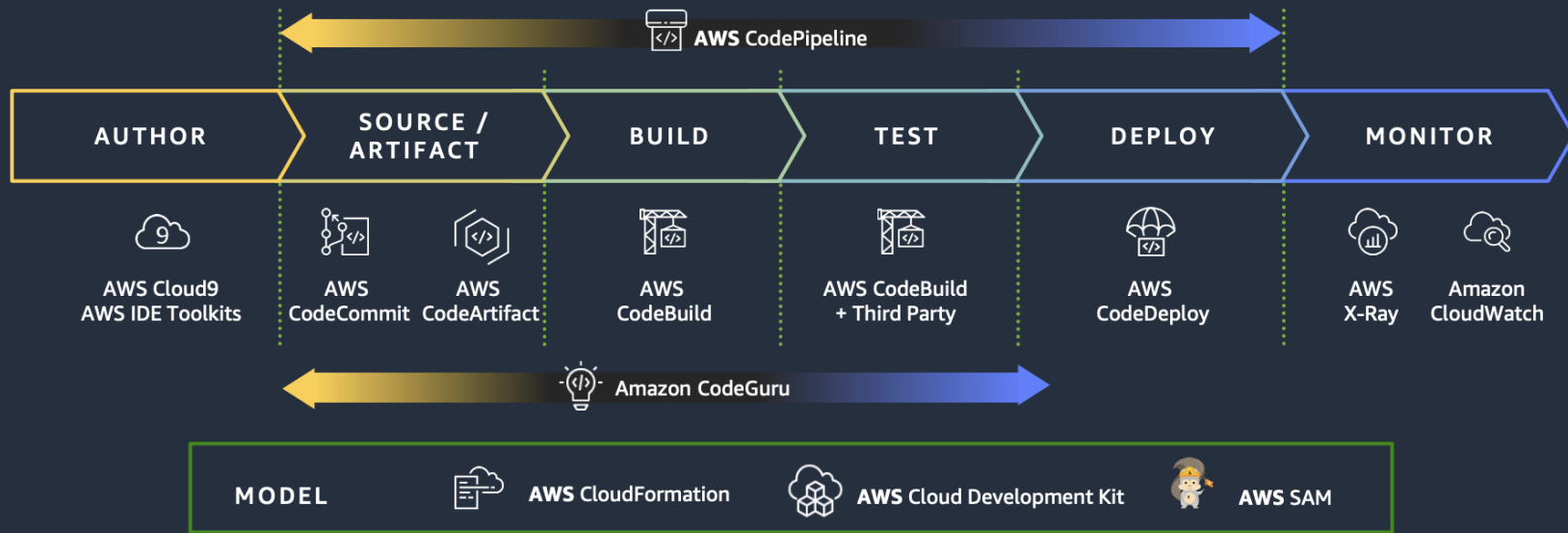
よりマネージドなCI/CDサービス

5. CodeGuru...リッチでニッチな使い道(機械学習使ったコードレビュー)
6. CodeStar...CI/CDめっちゃマネージド(テンプレートから選べる)
7. CodeCatalyst...メンバーオンボーディング、IDE連携、CodeStar+CDK構築まで一気通貫(マネージドの鬼)、コツを掴めば、爆速開発環境、インフラ、パイプライン構築可能かも



# 3.CI/CDに関わるCode〇〇サービス

改めて整理する(していただく)と



## まとめ

- ・ AWSのCodeなんちゃらサービスはたくさんある
- ・ CI/CDというか、運用構築は早めに取り組んだ方がいい
- ・ Codeなんちゃらは整理すれば意外と分かりやすい
- ・ 次回はCodePipeline、CodeBuildをやりたい