



# 適材適所で選ぶ CI/CD パイプライン構築サービス&ツール

Masahiro Matsumoto  
Senior Solutions Architect  
Amazon Web Service Japan G.K.

# 自己紹介

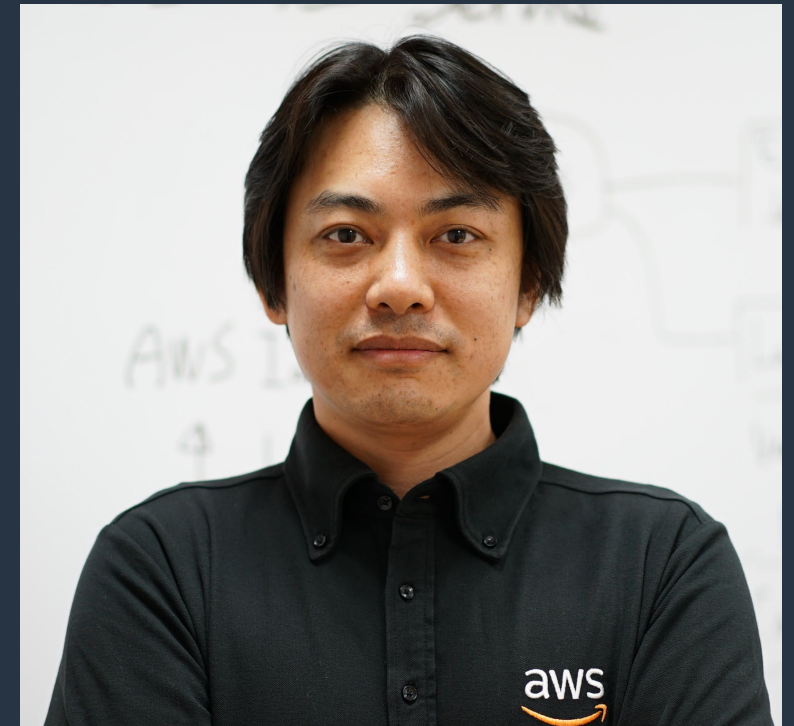
松本 雅博（まつもと まさひろ）

技術統括本部 西日本ソリューション部  
ソリューションアーキテクト

関西を中心に、西日本のお客様をご支援

好きなサービス

- Code シリーズ
- AWS CloudFormation, AWS Cloud Development Kit



# Agenda

- Day 1 の振り返り
- CI/CD パイプライン構築に利用できるサービス
- よく頂くご相談
- まとめ

# Day 1 の振り返り

# DevAx::connect シーズン 3 「rethink CI/CD」

開発者のための開発者による Web セミナーシリーズ

 #devaxconnect

← rethink CI/CD →

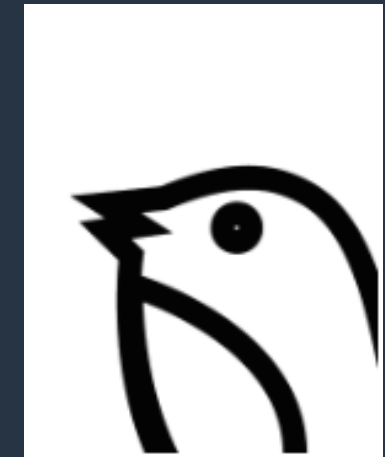
4/7  
CI/CD再入門

4/14  
CI/CDツール

4/14  
CI/CD可視化

4/21  
トランクベース

4/28  
複数ブランチ運用



令和も早や 4 年。私たちは「CI/CD」をできているのか

# CI/CDとは

- 継続的インテグレーション (CI)
  - 継続的にコードをチェックインして開発者の最新のコードがマージされた状態にしておくこと
  - 安全にマージを行うために自動ビルドや自動テストを構築すること
- 継続的デリバリー/デプロイ (CD)
  - デプロイメントパイプラインを構築すること
  - パイプラインの可視化をすること

# リリースプロセスのステージ



- ソースコードのチェックイン
- コードのピアレビュー

- コードのコンパイル
- ユニットテスト
- スタイルチェッカー
- コンテナイメージ、関数デプロイ
- パッケージの作成

- 周辺システムとの統合テスト
- 負荷テスト
- UIテスト
- セキュリティテスト

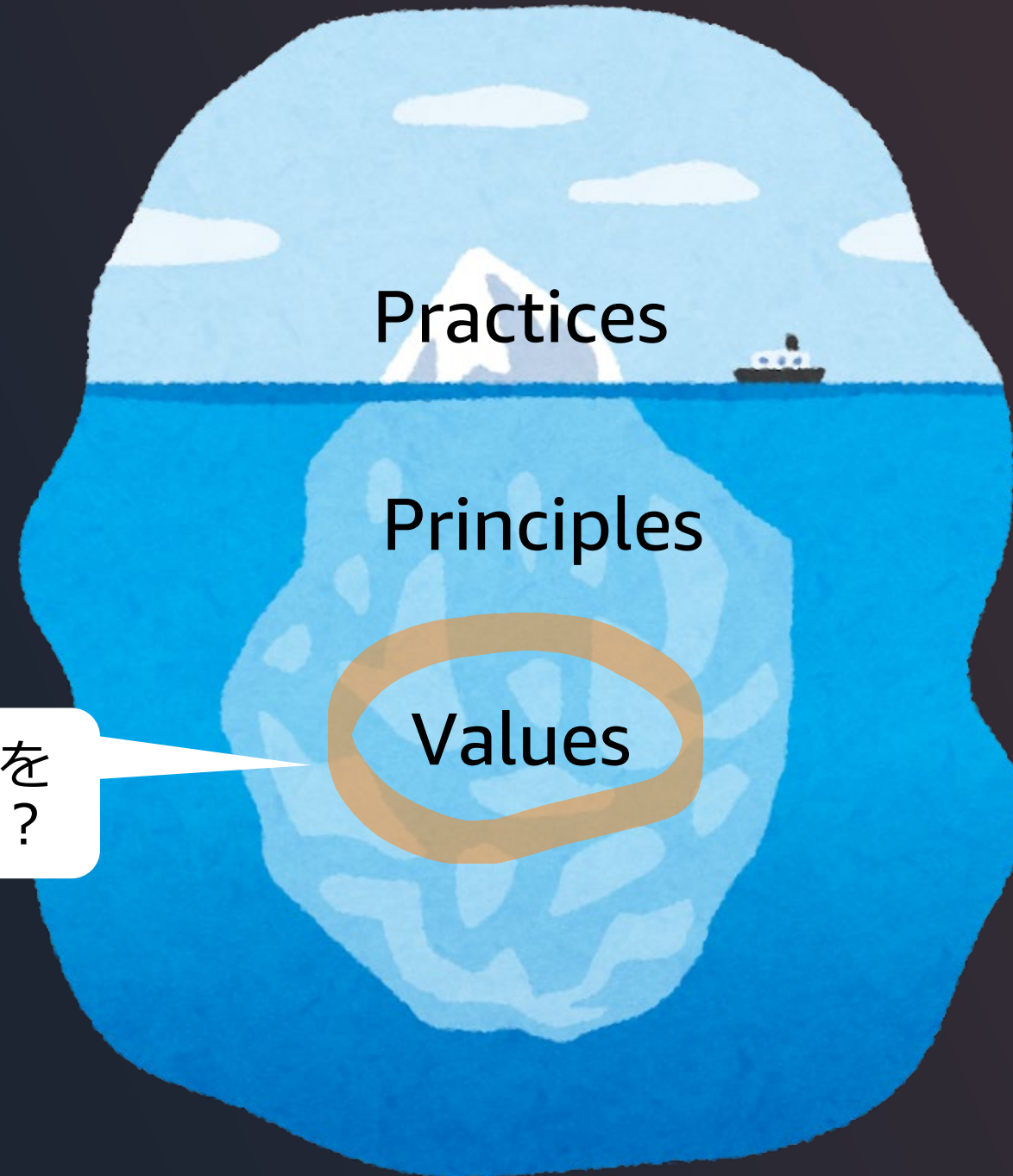
- 本番環境へのデプロイ
- エラーを素早く検知するための本番環境のモニタリング

# リリースプロセスのステージ





# 価値・原則・プラクティス



あなたの組織はなにを  
価値としていますか？

# CI/CDの導入

- CI準備編

1. コードのバージョン管理はできていますか？
2. ユニットテスト書いていますか？
3. ビルドは自動化出来ていますか？

- CI接続編

4. ビルドとテストを自動化する環境はありますか？
5. Webhookは使っていますか？
6. チェックインのルールを決めていますか？

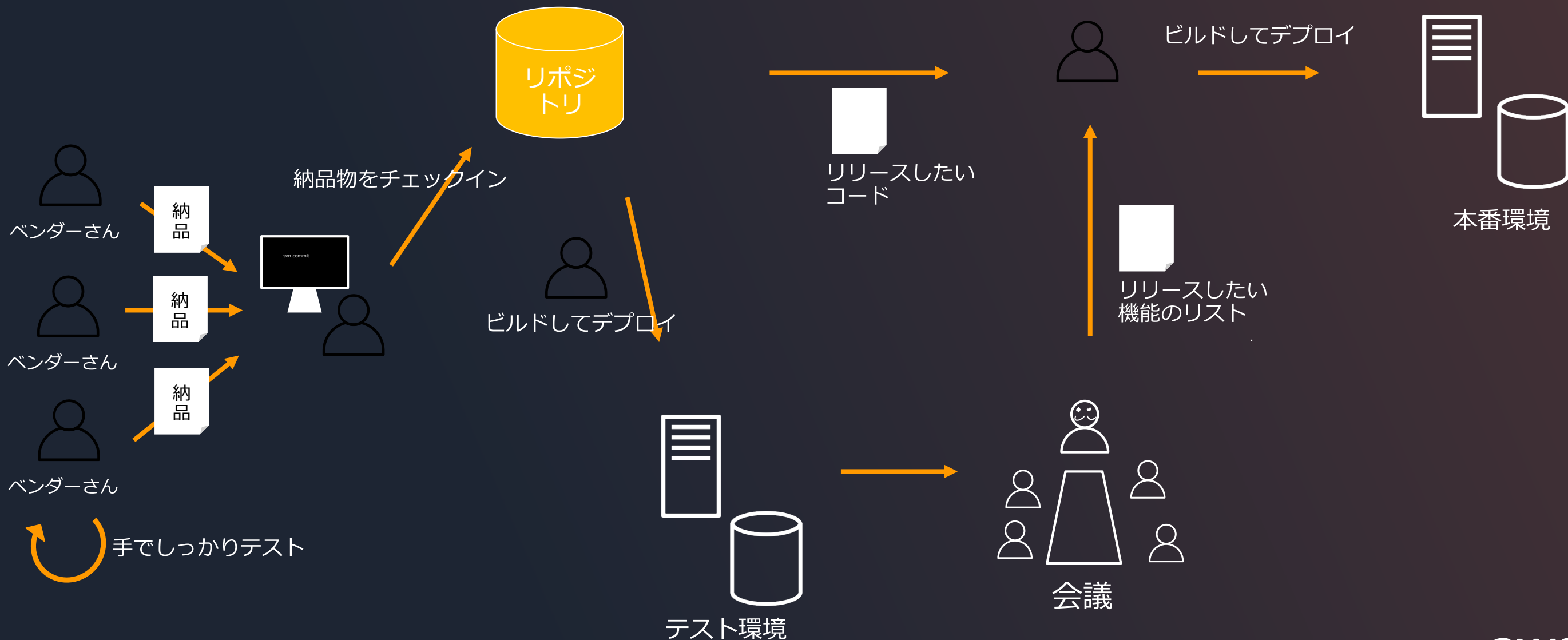
- CDステージ構築編

7. CIできていますか？
8. テストを人に頼っていませんか？
9. デプロイは自動化していますか？

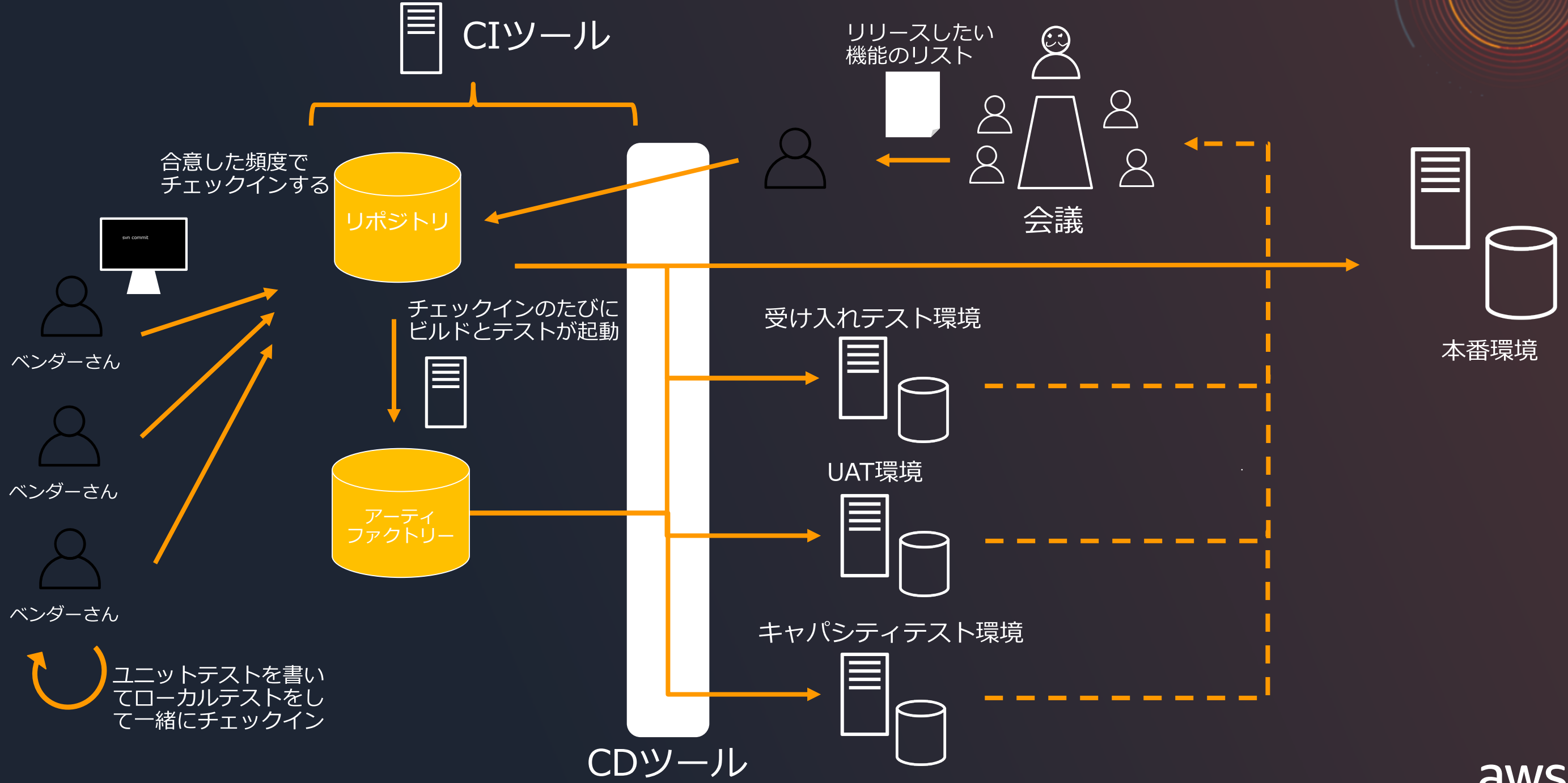
- CDパイプライン構築編

10. 本番環境のデプロイは安全に素早く出来ていますか？
11. パイプラインの状況が見えていますか？

# CI/CD導入前



# CI/CD導入後



# CI/CD パイプライン構築に 利用できるサービス

# CI/CDの導入

- CI準備編

1. コードのバージョン管理はできていますか？
2. ユニットテスト書いていますか？
3. ビルドは自動化出来ていますか？

- CI接続編

4. ビルドとテストを自動化する環境はありますか？
5. Webhookは使っていますか？
6. チェックインのルールを決めていますか？

- CDステージ構築編

7. CIできていますか？
8. テストを人に頼っていませんか？
9. デプロイは自動化していますか？

- CDパイプライン構築編

10. 本番環境のデプロイは安全に素早く出来ていますか？
11. パイプラインの状況が見えていますか？

# CI/CDの導入

- CI準備編

1. コードのバージョン管理はできていますか？
2. ユニットテスト書いていますか？
3. ビルドは自動化出来ていますか？

- CI接続編

4. ビルドとテストを自動化する環境はありますか？
5. Webhookは使っていますか？
6. チェックインのルールを決めていますか？

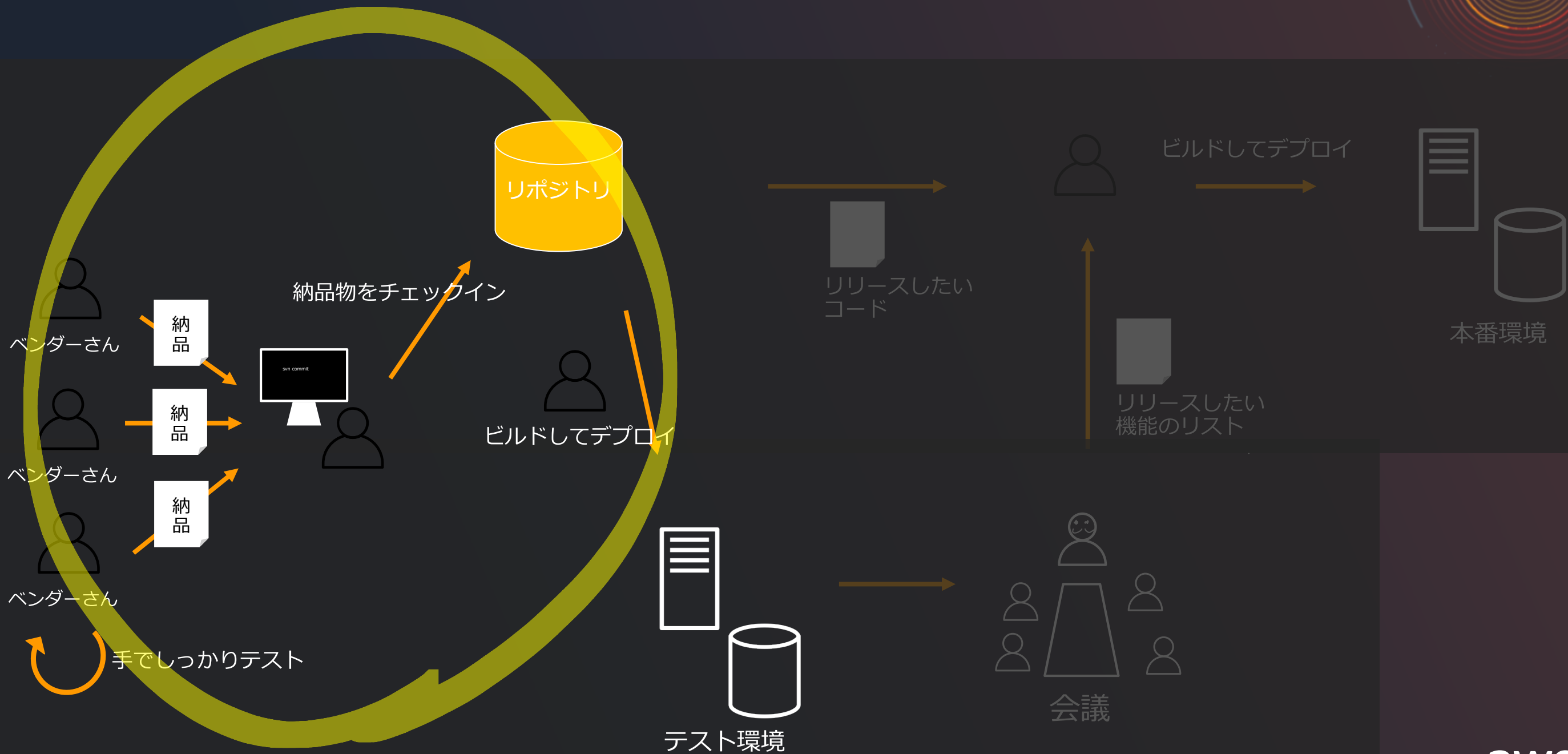
- CDステージ構築編

7. CIできていますか？
8. テストを人に頼っていませんか？
9. デプロイは自動化していますか？

- CDパイプライン構築編

10. 本番環境のデプロイは安全に素早く出来ていますか？
11. パイプラインの状況が見えていますか？

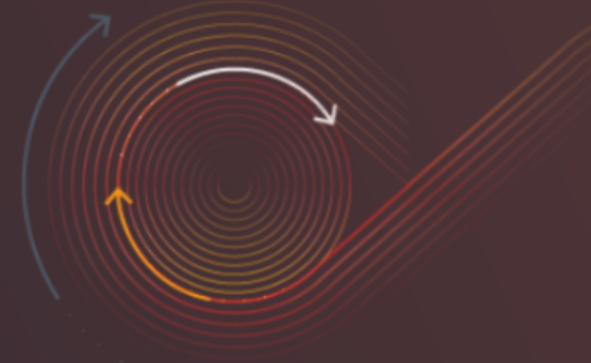
# CI準備編



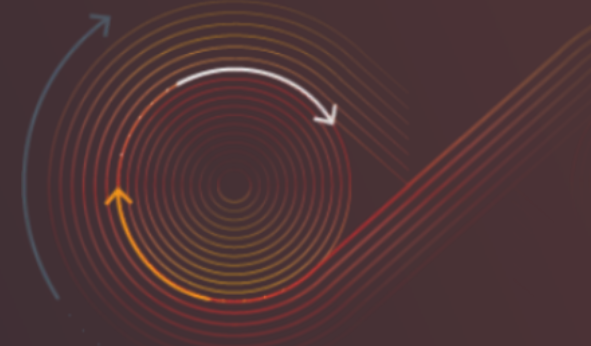


# コードのバージョン管理

- ファイルの変更履歴管理
- チームでの共同作業



# コードのバージョン管理



- ファイルの変更履歴管理
- チームでの共同作業



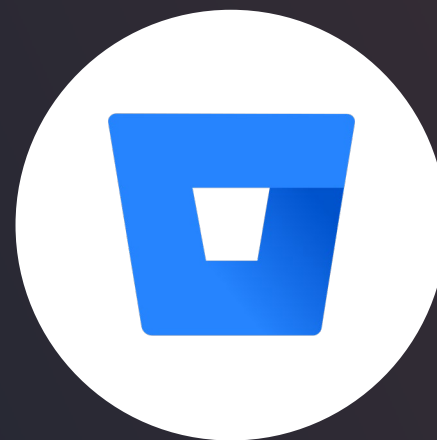
AWS CodeCommit



Git



Backlog



Bitbucket

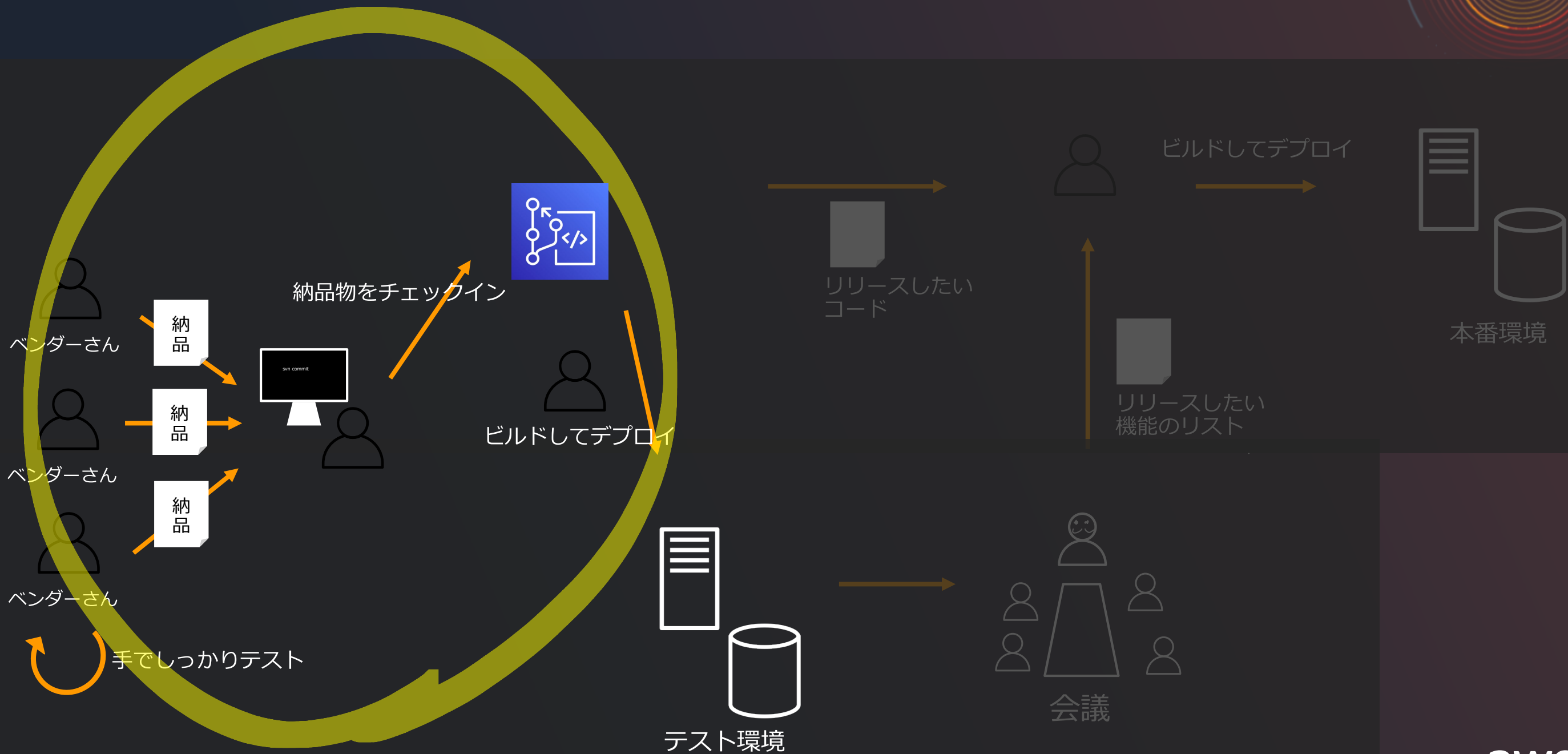


GitHub



GitLab

# CI準備編



# CI/CDの導入

- CI準備編

1. コードのバージョン管理はできていますか？
2. ユニットテスト書いていますか？
3. ビルドは自動化出来ていますか？

- CI接続編

4. ビルドとテストを自動化する環境はありますか？
5. Webhookは使っていますか？
6. チェックインのルールを決めていますか？

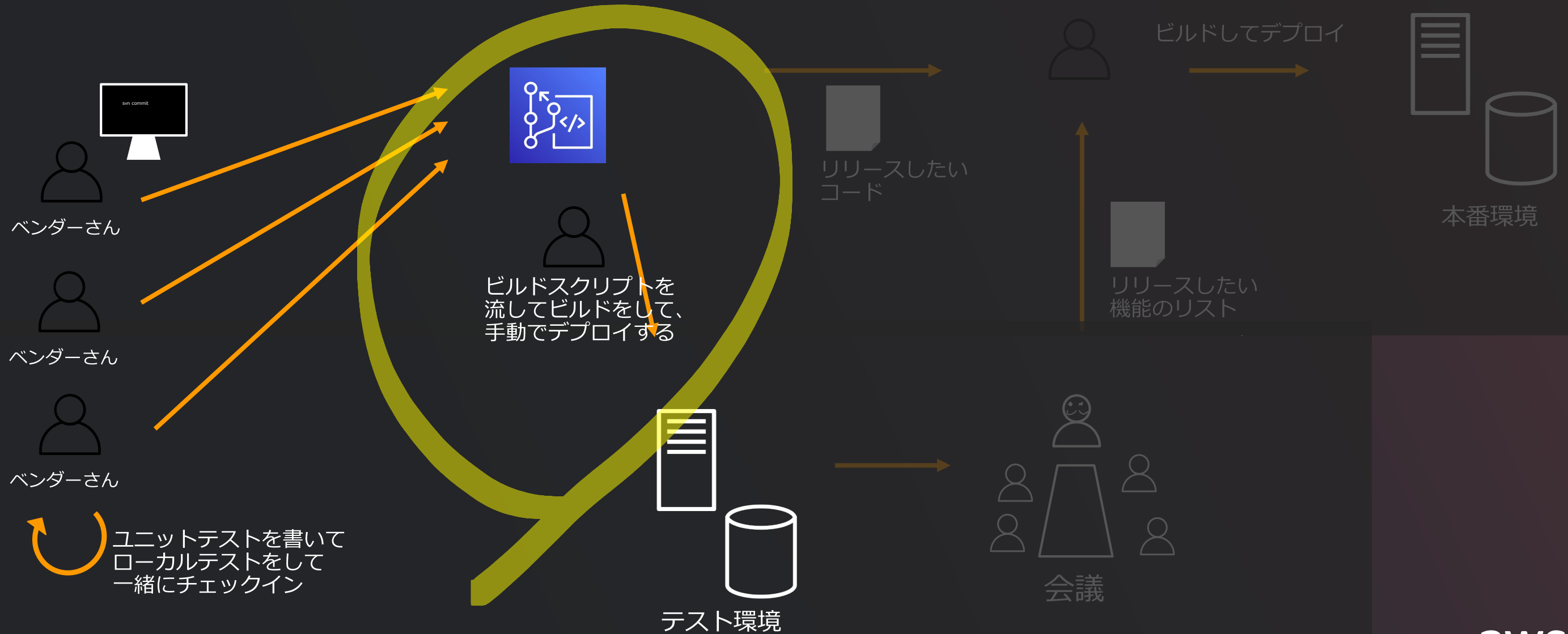
- CDステージ構築編

7. CIできていますか？
8. テストを人に頼っていませんか？
9. デプロイは自動化していますか？

- CDパイプライン構築編

10. 本番環境のデプロイは安全に素早く出来ていますか？
11. パイプラインの状況が見えていますか？

# CI接続編



# ビルドとテストを自動化する環境

- コードのチェックインをトリガーにビルドとテストを実行
- モニタリングとロギング、成果物の保持



# ビルドとテストを自動化する環境

- コードのチェックインをトリガーにビルドとテストを実行
- モニタリングとロギング、成果物の保持



AWS CodeBuild



Jenkins



Bitbucket



CircleCI

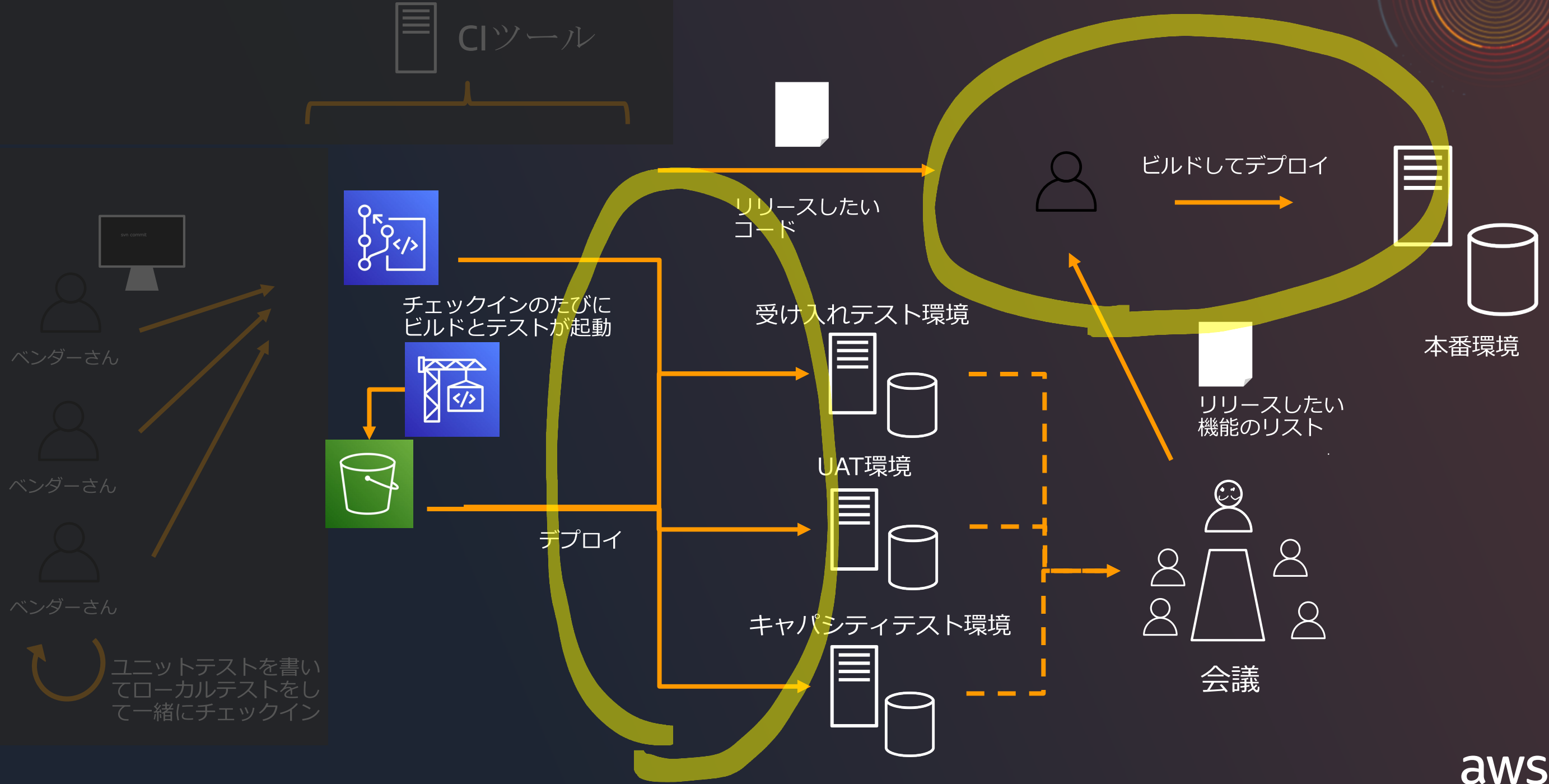


GitHub



GitLab

# CD : パイプライン構築編



ユニットテストを書いてローカルテストをして一緒にチェックイン



# CI/CDの導入

- CI準備編

1. コードのバージョン管理はできていますか？
2. ユニットテスト書いていますか？
3. ビルドは自動化出来ていますか？

- CI接続編

4. ビルドとテストを自動化する環境はありますか？
5. Webhookは使っていますか？
6. チェックインのルールを決めていますか？

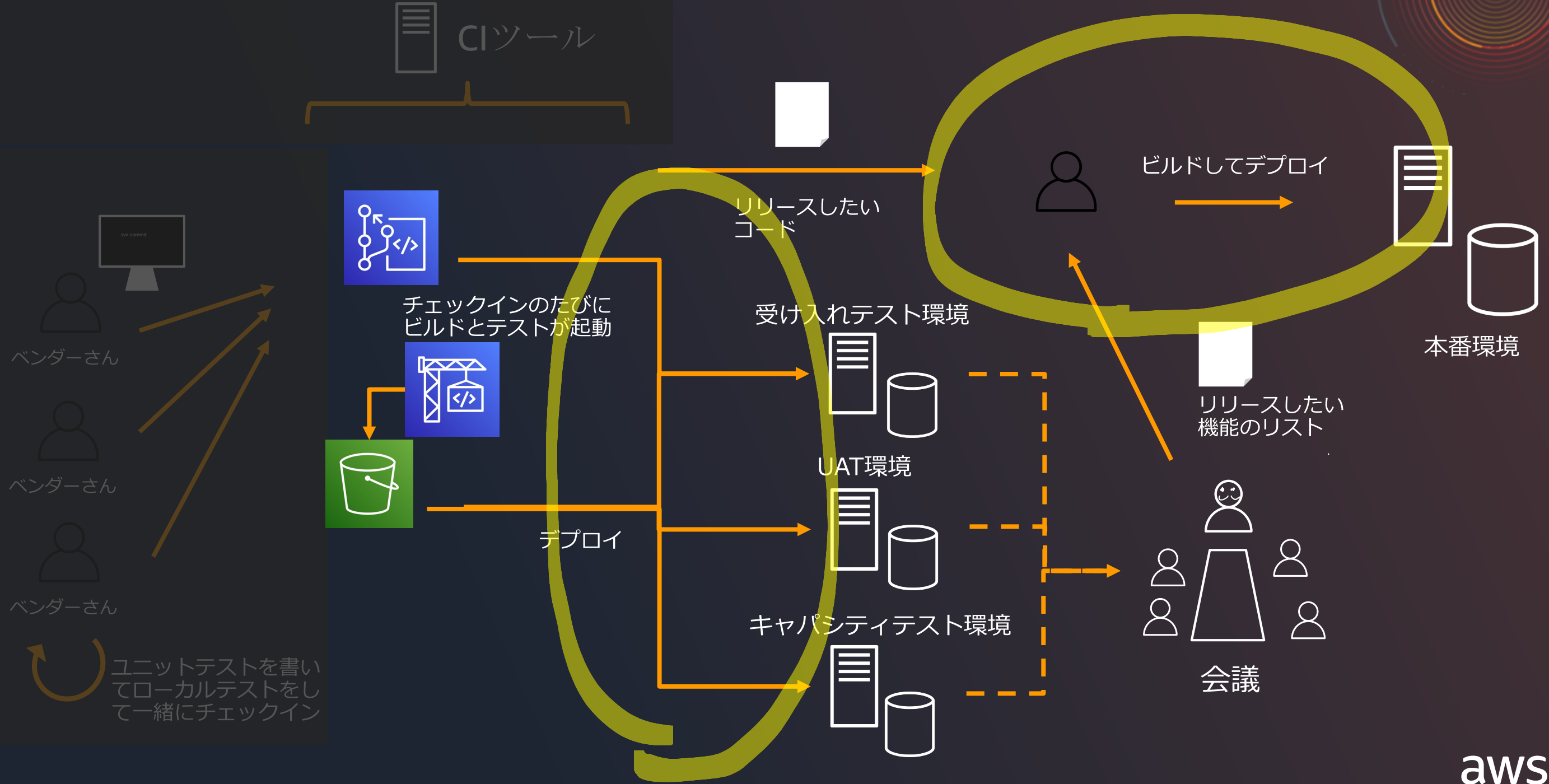
- CDステージ構築編

7. CIできていますか？
8. テストを人に頼っていませんか？
9. デプロイは自動化していますか？

- CDパイプライン構築編

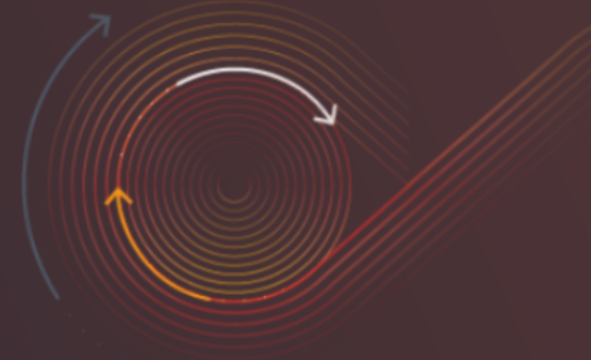
10. 本番環境のデプロイは安全に素早く出来ていますか？
11. パイプラインの状況が見えていますか？

# CD : パイプライン構築編



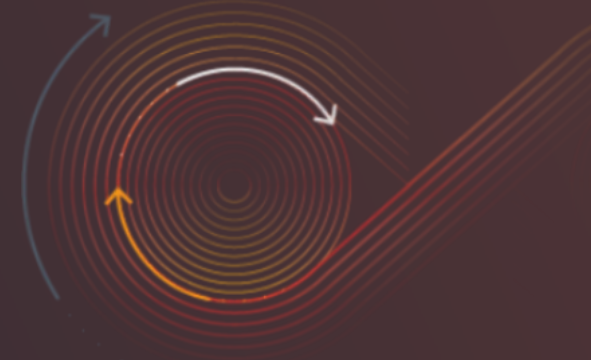
# デプロイの自動化

- 素早く、安全にデプロイできる
- ロールバック機能



# デプロイの自動化

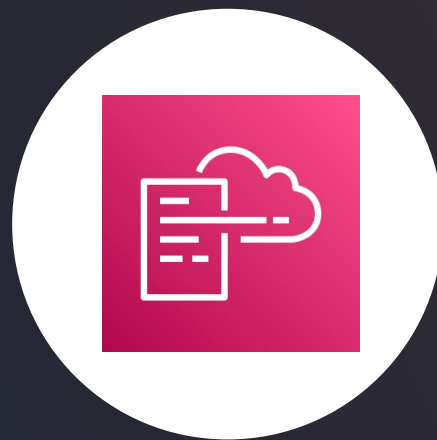
- 素早く、安全にデプロイできる
- ロールバック機能



AWS CodeDeploy



AWS CDK



AWS CloudFormation

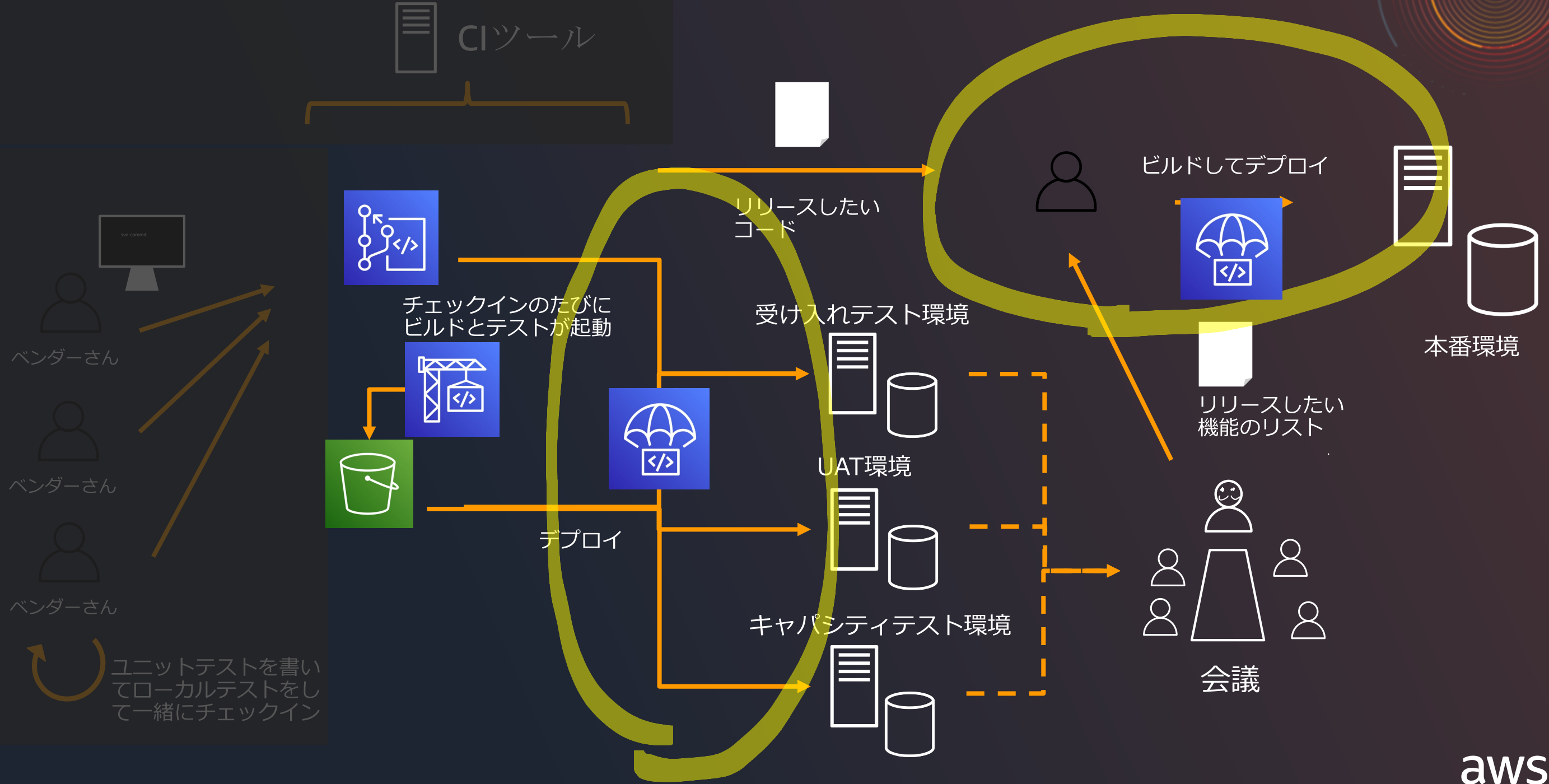


AWS SAM

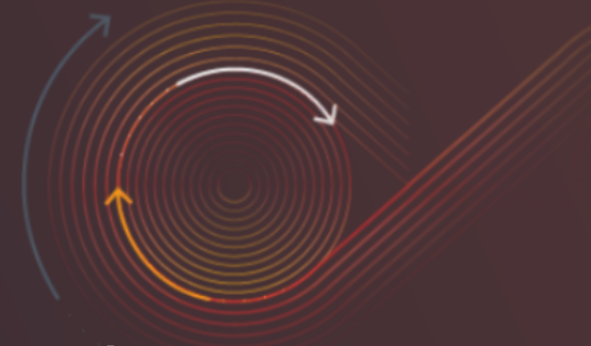


Terraform

# CD : パイプライン構築編



# パイプラインの構築

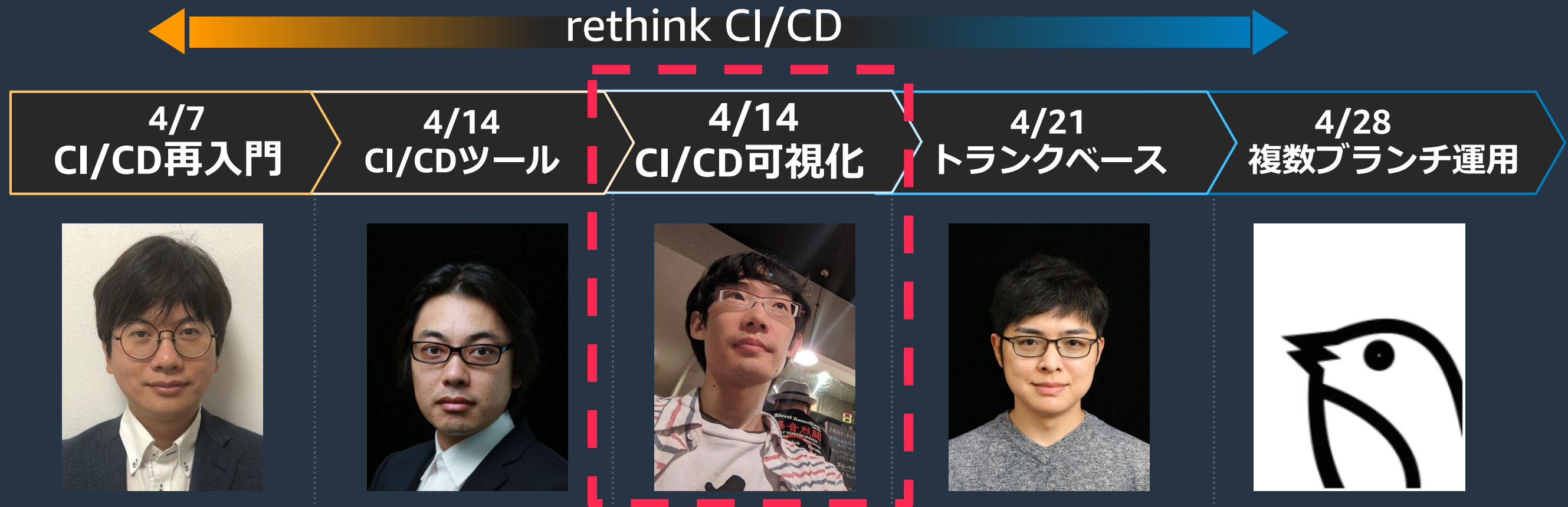


- ソースコードのチェックインをトリガーにした一連の処理を実行
- パイプラインの可視化

# DevAx::connect シーズン 3 「rethink CI/CD」

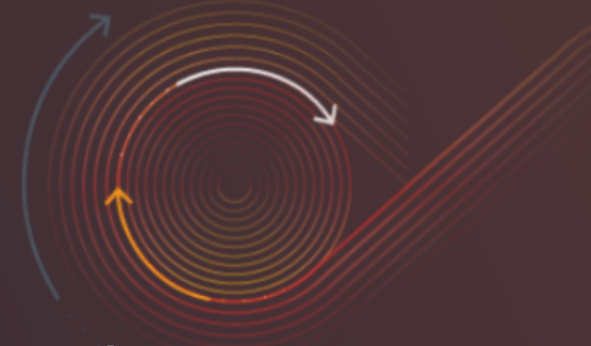
開発者のための開発者による Web セミナーシリーズ

 #devaxconnect



令和も早や 4 年。私たちは「CI/CD」をできているのか

# パイプラインの構築



- ソースコードのチェックインをトリガーにした一連の処理を実行
- パイプラインの可視化



AWS CodePipeline



Jenkins



Bitbucket



CircleCI



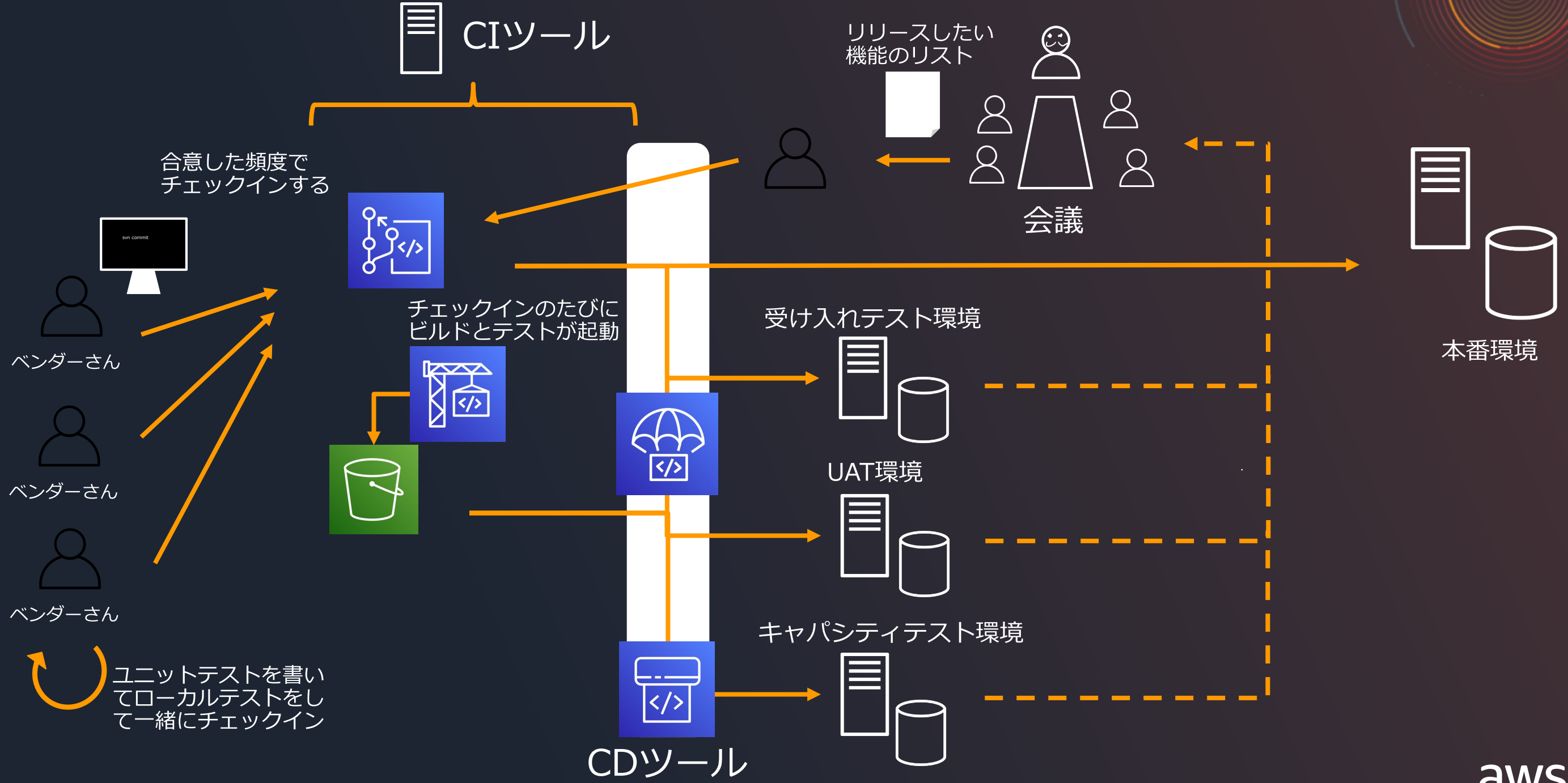
GitHub



GitLab



# CD : パイプライン構築編



よく頂くご相談

# よく頂くご相談



CI/CD パイプラインを構築したいのですが、AWS サービスとサードパーティ製品のどちらを使えばいいですか？

AWS にシステムを移行するのですが、現在利用しているツールも Code サービスへ移行する方が良いですか？



# CI/CD パイプライン構築ツール選定

- これから内製化に着手するなど、ご希望のサービスが特にない場合は、フルマネージドで素早く利用開始できる **Code** サービスをご提案させていただくことが多いです。
- 社内標準ツールがある場合、新しいサービスに乗り換えるには社内調整に時間を要するため、そのツールを継続利用されることもあります。
- 現在ご利用中のツールがある場合、移行コスト、学習コスト等の面から完全移行ではなく **Code** サービスの一部をご利用になられる例もあります。

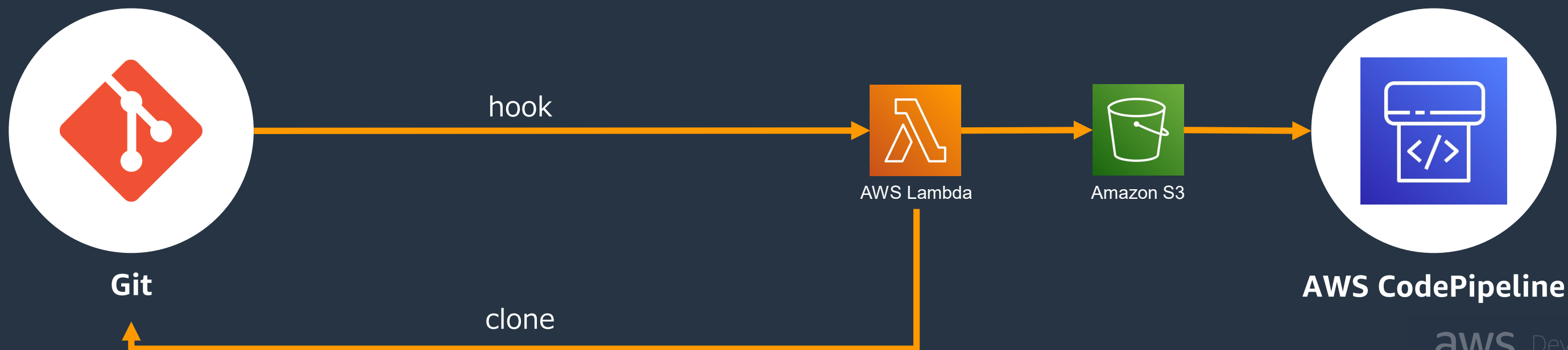
# よく頂くご相談

CodePipelineを利用したいのですが、  
利用しているバージョン管理システムに対応していません。



# 任意のバージョン管理システムと連携させたい

- S3 を経由することで連携が可能です。



# よく頂くご相談



既存のビルドスクリプトやデプロイスクリプトを  
Code サービスで利用できますか？

# 既存のスクリプトを Code サービスで利用したい

- CodeBuild および CodeDeploy で処理できる形式に加工していただくことでご利用いただけます。
- 複雑化した実行条件を整理し、リリースプロセスから属人性を排除することに成功したお客様もいらっしゃいます。



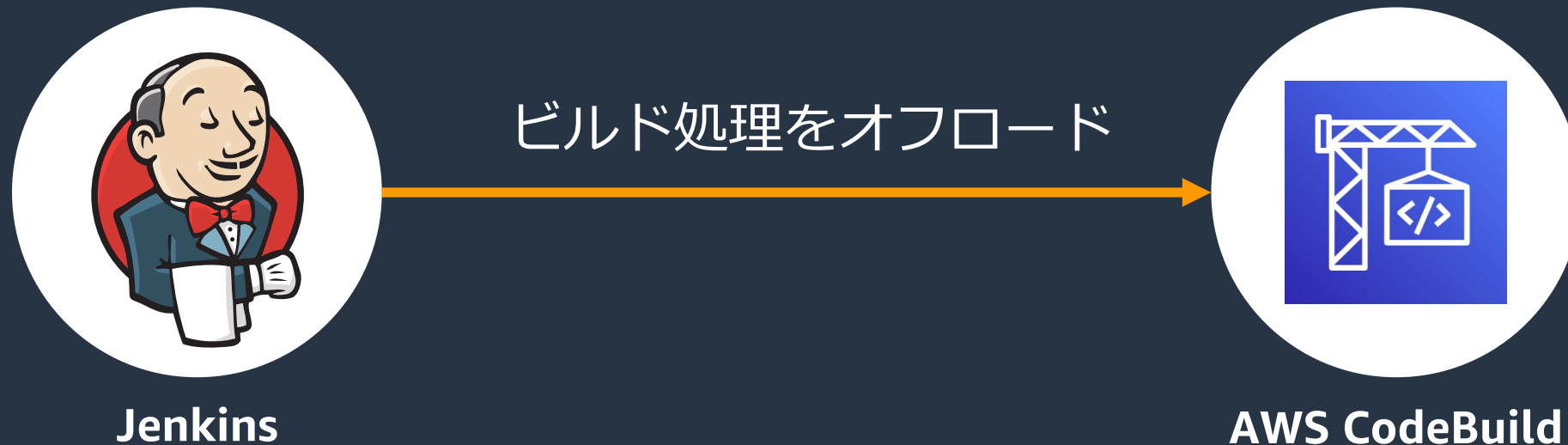
# よく頂くご相談

EC2 上に構築した Jenkins でビルドを行なっています。  
コストを改善する方法はありますか？



# Jenkins サーバの運用コストを改善したい

- コントローラーとは別のサーバーでビルドすることで、コントローラーのインスタンスタイプの見直しができる可能性があります。
- **Jenkins** プラグインを利用すると、**CodeBuild** や **EC2, ECS** へビルド処理をオフロードできます。



**Jenkins**

**AWS CodeBuild**



# よく頂くご相談



PoC を実施していますが、コードを書いているのは私一人ですが、バージョン管理システムを利用した方がよいですか？

# 一人でもバージョン管理した方が良いか

- コードを書く人が一人であっても、バージョン管理を行うことをお勧めします。
- 個人開発であっても変更履歴を残しておくこと、何かあった際のリカバリーが行いやすくなります。
- いつでも以前の状態に戻すことができる環境があれば、大きな変更にもチャレンジしやすくなります。

# よく頂くご相談

CI/CD にチャレンジしたいのですが、ユニットテストを書いていないとメリットがありませんか？



# CI/CD パイプラインの構築にチャレンジしたい

- ユニットテストがあることがベストですが、テストの作成にはそれなりの時間がかかります。
- パイプラインを整備しつつ、変更頻度の高いコードやテスト実施に工数がかかる部分からテストを整備するのが良いかと思われれます。
- **CI/CD** パイプラインの構築を進める中で見えてくる課題もあります。開発チームだけでなく、社内の関連部署とも連携して進める必要があります。

# まとめ



# まとめ

- CI/CD パイプライン構築に利用できるサービスは複数あり、それらを組み合わせて利用することもできる
- 利用するツールの選定時には開発要件、移行・調整コストなど多面的な検討を
- できるところからコツコツと。改善で生まれた余剰時間で、より価値を生む作業に注力を！



# Thank you!

Masahiro Matsumoto  
Senior Solutions Architect  
Amazon Web Service Japan G.K.

# 参考資料

- [\[AWS Black Belt Online Seminar\] AWS CodeCommit & AWS CodeArtifact 資料及び QA](#)
- [\[AWS Black Belt Online Seminar\] AWS CodeStar & AWS CodePipeline 資料及び QA](#)
- [\[AWS Black Belt Online Seminar\] AWS CodeBuild 資料及び QA](#)
- [\[AWS Black Belt Online Seminar\] AWS CodeDeploy 資料及び QA](#)