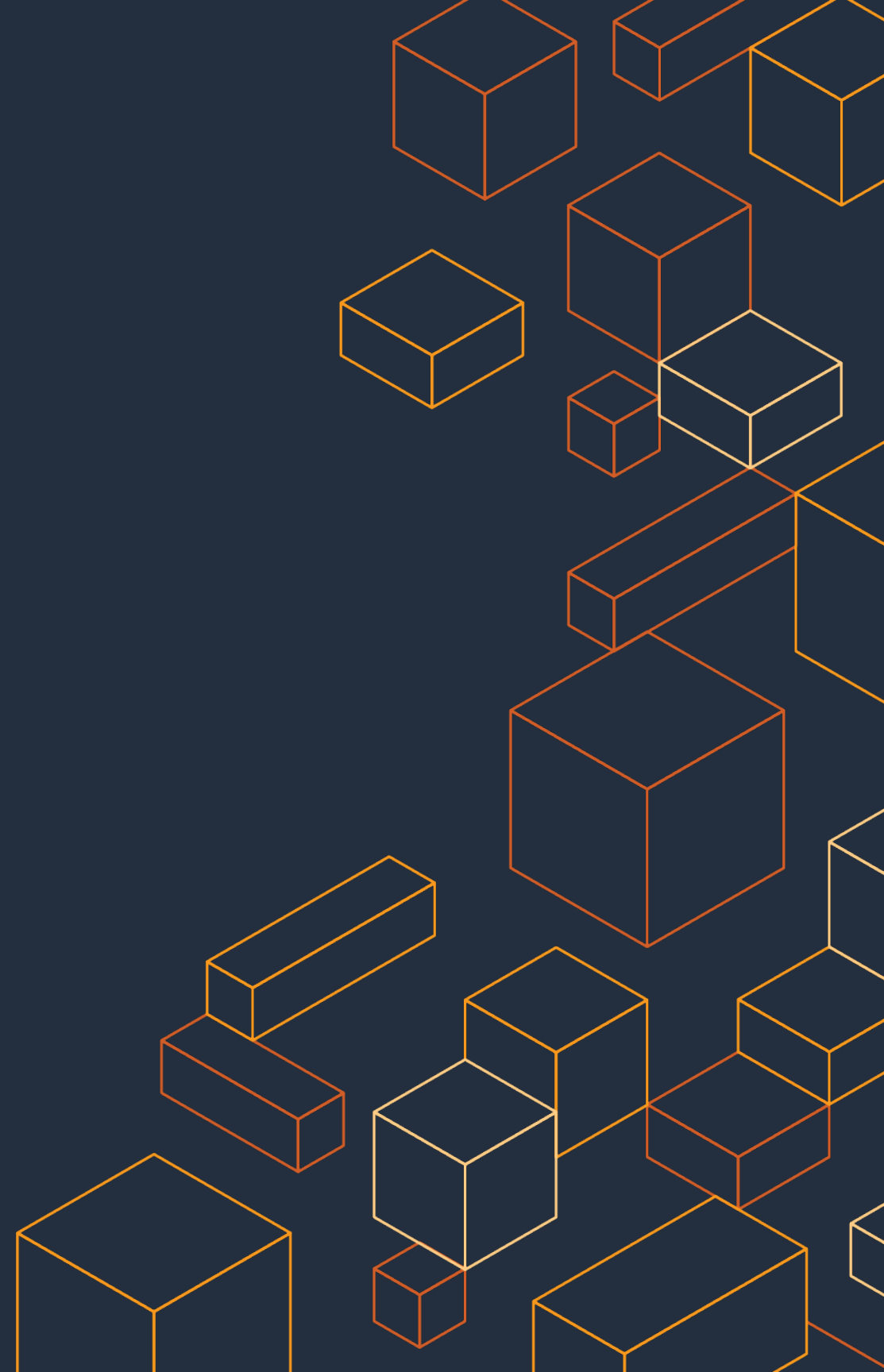




CI/CD on AWS

Accelerate delivery with better quality and control



Agenda

Foundations of CI/CD

Release Process

Amazon CI/CD Tools

Infrastructure as Code

What's Next?



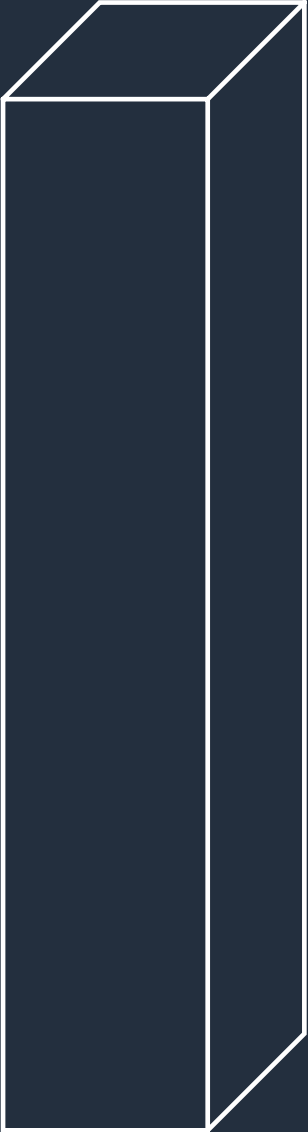
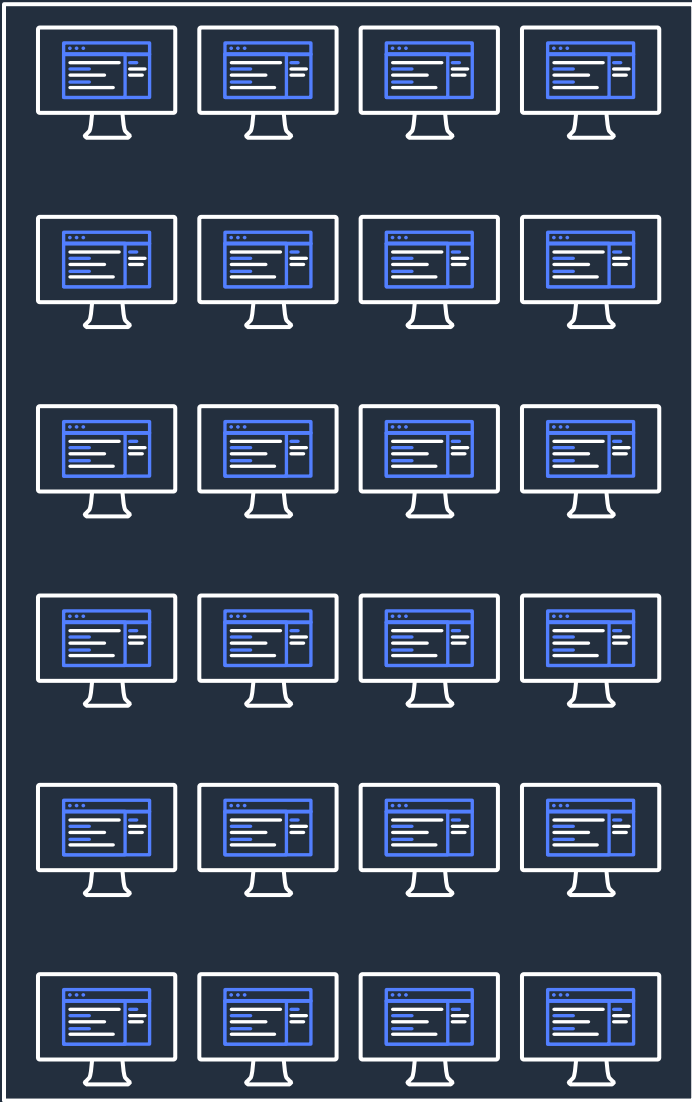
Foundations of CI/CD



Monolith development lifecycle

Developers

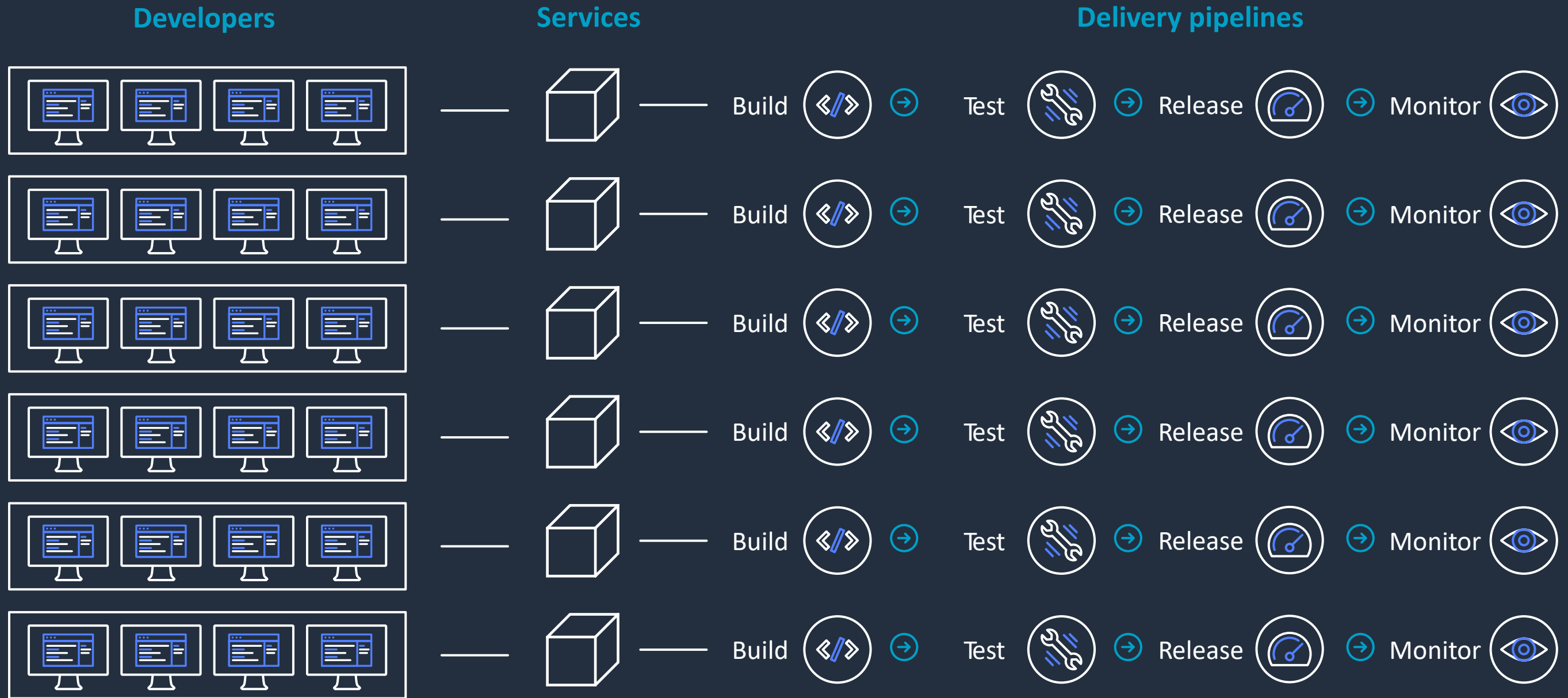
Services



Delivery pipelines



Microservice development lifecycle



Key reasons for organizations to adopt CI/CD principles

- Accelerate the delivery of new, high-quality services
- Reduce the impact of changes
- Gain insight across resources and applications
- Protect customers and the business

Effects of CI/CD

Deployment frequency

Weekly–monthly



Hourly–daily

Change lead time

1–6 months



1–7 days

Change failure rate

46%–60%



0%–15%



48% of
software
teams

Source: 2019 State of DevOps Report, DORA

Release process stages



- Check-in source code such as .java files
- Peer review new code

- Compile code
- Unit tests
- Style checkers
- Create container images and function deployment packages

- Integration testing with other systems
- Load testing
- UI testing
- Security testing

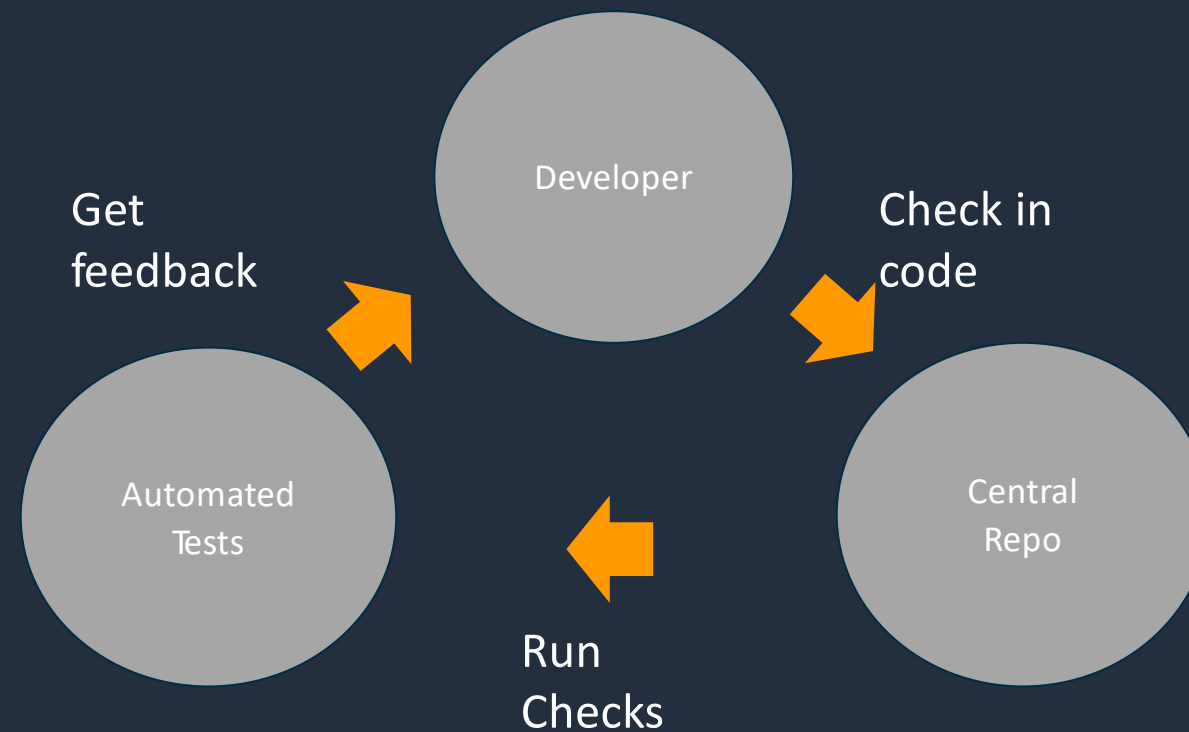
- Deploy to production environments
- Monitor code in production to quickly detect errors

Release lifecycle



Continuous Integration

- Continuous integration is a software development practice where developers regularly merge their code changes, into a central repository, after which automated builds and tests are run.
- The key goals of continuous integration are to find and address bugs quicker, improve software quality, and reduce the time it takes to validate and release new software updates.



Continuous Integration goals



Continuous Integration

1. Automatically kick off a new build when new code is checked in
2. Build and test code in a consistent, repeatable environment
3. Continually have an artifact ready for deployment
4. Continually close feedback loop when build fails

Continuous Delivery

- Continuous delivery is a software development practice where code changes are automatically built, tested, and prepared for a release to production.
- It **expands** upon continuous integration by **deploying** all code changes to a testing environment and/or a production environment **after the build stage**.
- When continuous delivery is implemented properly, developers will always have a deployment-ready build artifact that has passed through a standardized test process.



Continuous Deployment goals



Continuous deployment

1. Automatically deploy new changes to staging environments for testing
2. Deploy to production safely without impacting customers
3. Deliver to customers faster: Increase deployment frequency and reduce change lead time and change failure rate

Continuous Delivery vs Continuous Deployment

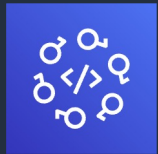


Operation excellence for CI/CD

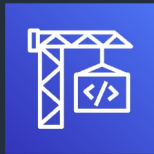
- Monitor the deployment process
- Integration with Incident management System
- Establish Operation Model
- Use Monitoring tools for monitoring and dashboard

By way of introduction...

CI/CD tools



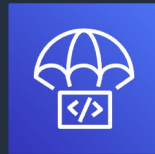
AWS
CodeStar



AWS CodeBuild



AWS
CodeCommit

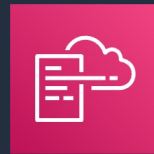


AWS CodeDeploy



AWS
CodePipeline

Infrastructure as code

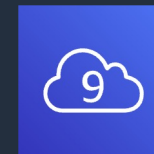


AWS
CloudFormation



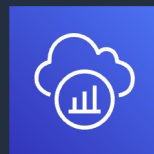
AWS
CDK

IDE

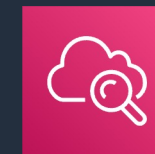


AWS Cloud9

Monitoring & tracing



AWS
X-Ray



Amazon
CloudWatch

Web apps



AWS Elastic
Beanstalk

IDE and DevOps toolkits



Visual Studio
Code



IntelliJ



PyCharm



Visual Studio



Eclipse



VSTS

CLI and scripting tools

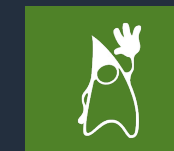


AWS CLI



Tools for
PowerShell

Languages



Amazon
Corretto

Mobile



AWS
Amplify

SDKs



JavaScript



Python



PHP



.NET



Ruby



Java



Go

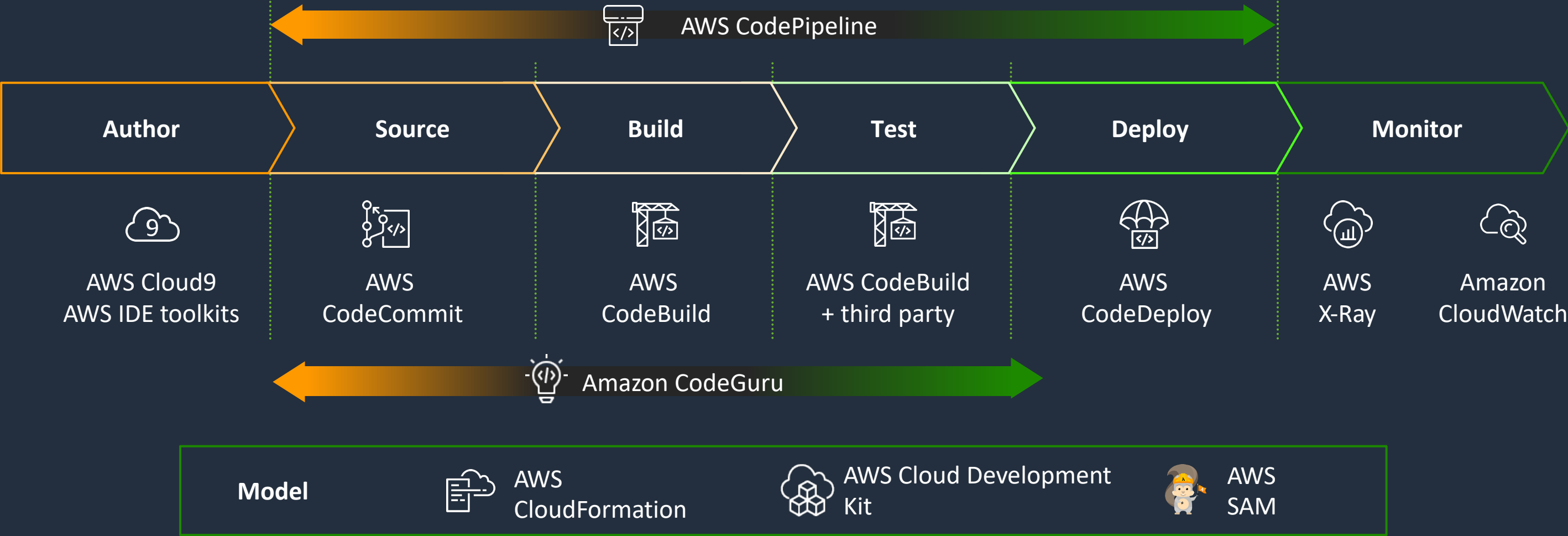


Node.js



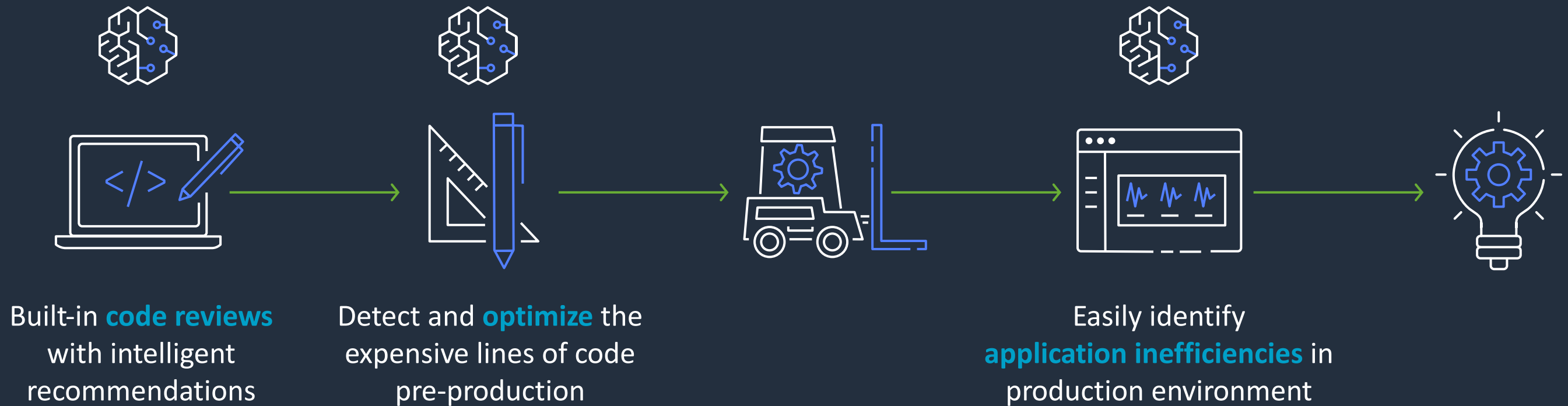
C++

CI/CD for modern software delivery

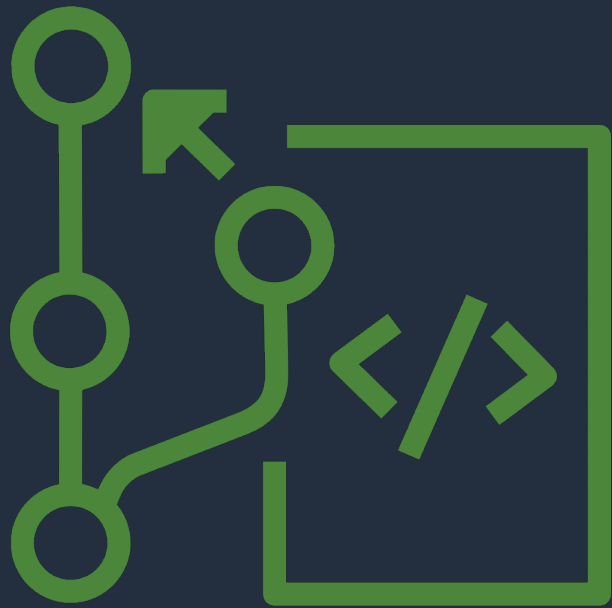


Amazon CodeGuru

USING MACHINE LEARNING (ML) TO BUILD AND RUN HIGH-PERFORMING APPLICATIONS



AWS CodeCommit



- Secure, highly scalable, managed source control service that hosts private git repositories
- Works with existing git tools
- Integrates with AWS services like IAM, Amazon EventBridge, Amazon SNS
- No hardware to provision and scale
- Highly available and durable (backed by s3)

AWS CodeBuild



- Fully managed build service that compiles source code, runs tests, and produces software packages
- Scales continuously and processes multiple builds concurrently
- No build servers to manage
- Pay by the minute, only for the compute resources you use
- Monitor builds through CloudWatch Events

AWS CodeDeploy



- Automates code deployments to any instance and Lambda
- Handles the complexity of updating your applications
- Avoids downtime during application deployment
- Rolls back automatically if failure detected
- Deploys to Amazon EC2, Lambda, or on-premises servers

AWS CodePipeline



- Continuous delivery service for fast and reliable application updates
- Model and visualize your software release process
- Builds, tests, and deploys your code every time there is a code change
- Integrates with third-party tools and AWS

Amazon Web Services (AWS) observability portfolio



Amazon CloudWatch

Complete visibility of cloud resources and applications

Monitor applications

Respond to performance changes

Optimize resource utilization

Get a unified view of operational health



AWS X-Ray

Analyze and debug production, distributed applications

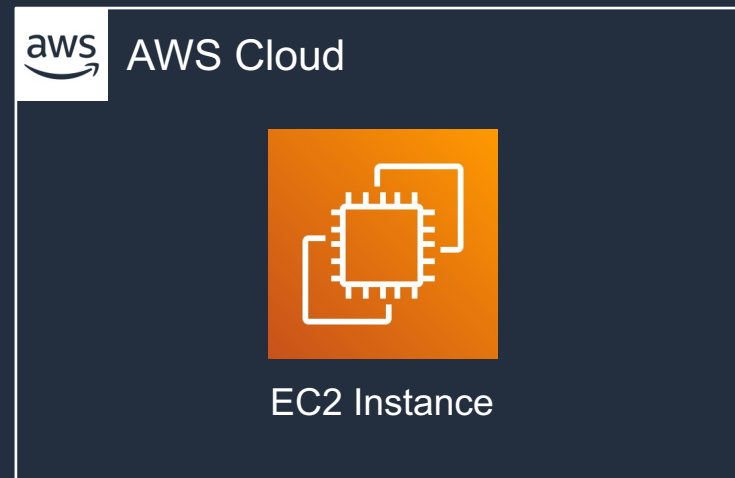
Identify performance bottlenecks

Troubleshoot root cause

Trace user requests

For simple & complex applications

Infrastructure as Code (IaC)



```
AWSTemplateFormatVersion: "2010-09-09"  
Description: Creates an EC2 Instance  
Resources:  
  MyEC2Instance:  
    Type: "AWS::EC2::Instance"  
    Properties:  
      ImageId: "ami-0ff8a91507f77f867"  
      InstanceType: t2.micro
```

What?

1. Writing code to create, configure, and deploy infrastructure components
2. Infrastructure includes: networking, compute, databases, security, management tools, etc.

Why?

1. Makes infrastructure changes **repeatable** and **predictable**
2. Documents your infrastructure
3. Automates the provisioning process
4. Eliminates configuration drift through automation

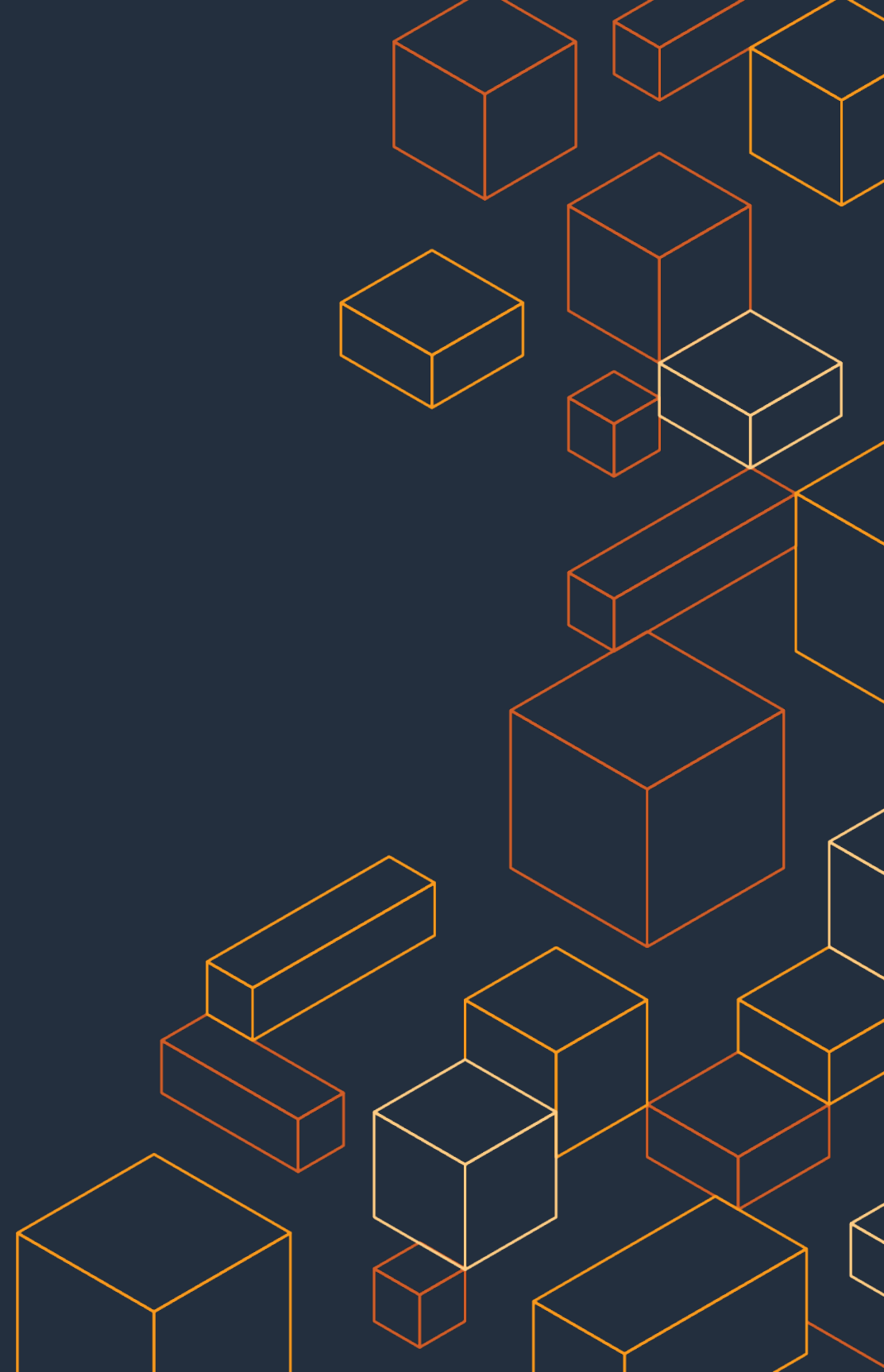
Infrastructure as Code with AWS CloudFormation



- Simplified way to create and manage a collection of AWS resources
- Enables orderly and predictable provisioning and updating of resources
- Enables version control of your AWS infrastructure
- Only pay for the resources you create



What's Next?



Extending CloudFormation with Serverless Application Model (SAM)



- Framework for building serverless applications
- Shorthand syntax to express functions, APIs, databases, and event source mappings
- Model with YAML, deploy using AWS CloudFormation
- Open source

Shorten the learning curve with AWS Cloud Development Kit (CDK)

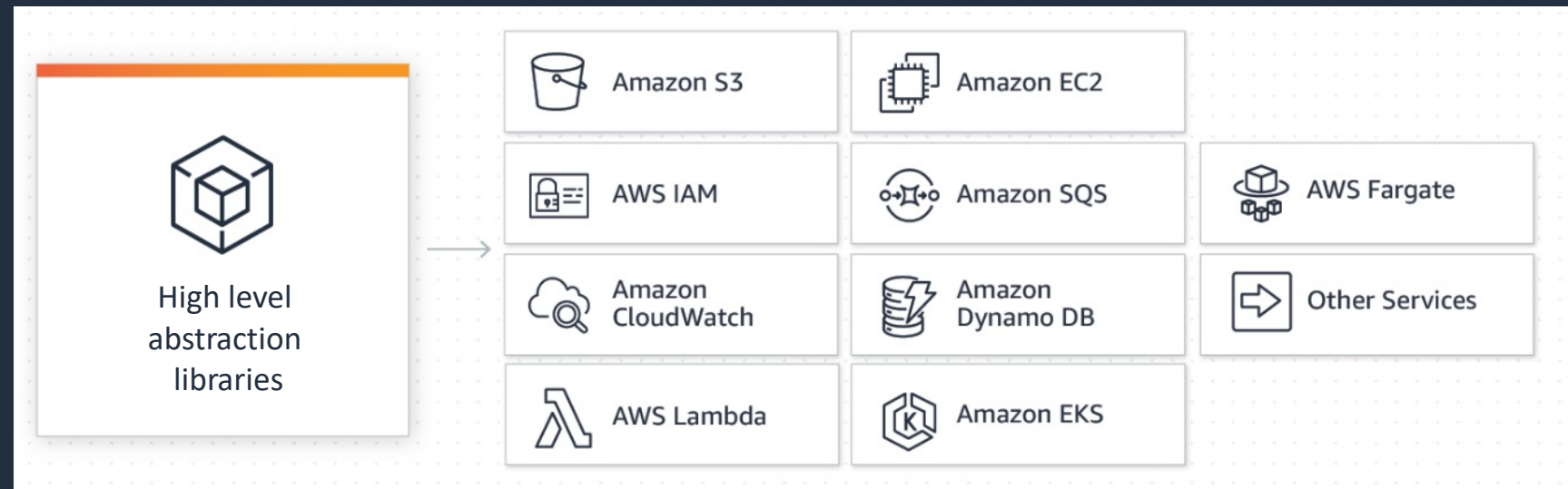
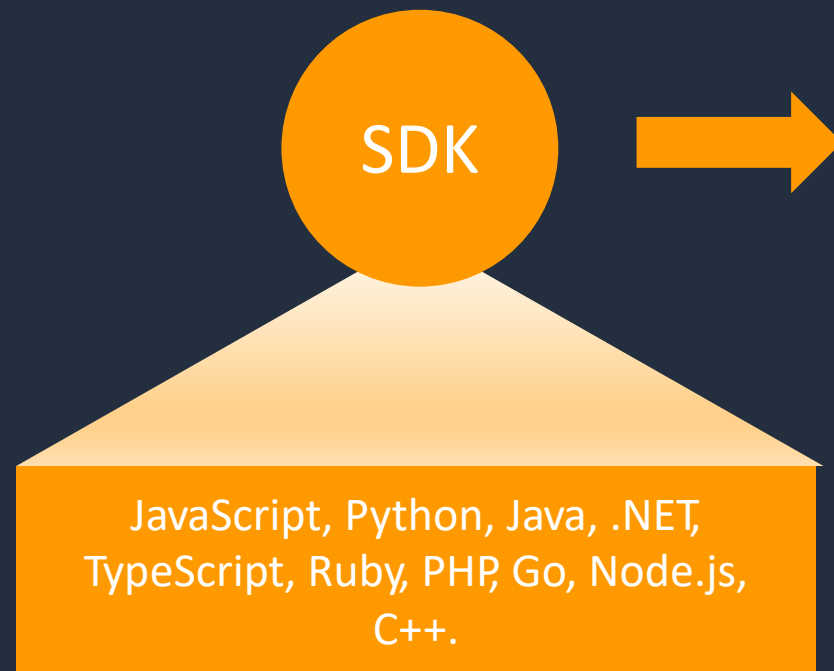
Brings cloud infrastructure to developers in ways they can understand

- Build cloud infrastructure with the languages they already know
- Use their existing tools and workflows
- Helpful abstractions that remove the need to learn the details
- Vibrant and fast-growing community of developers



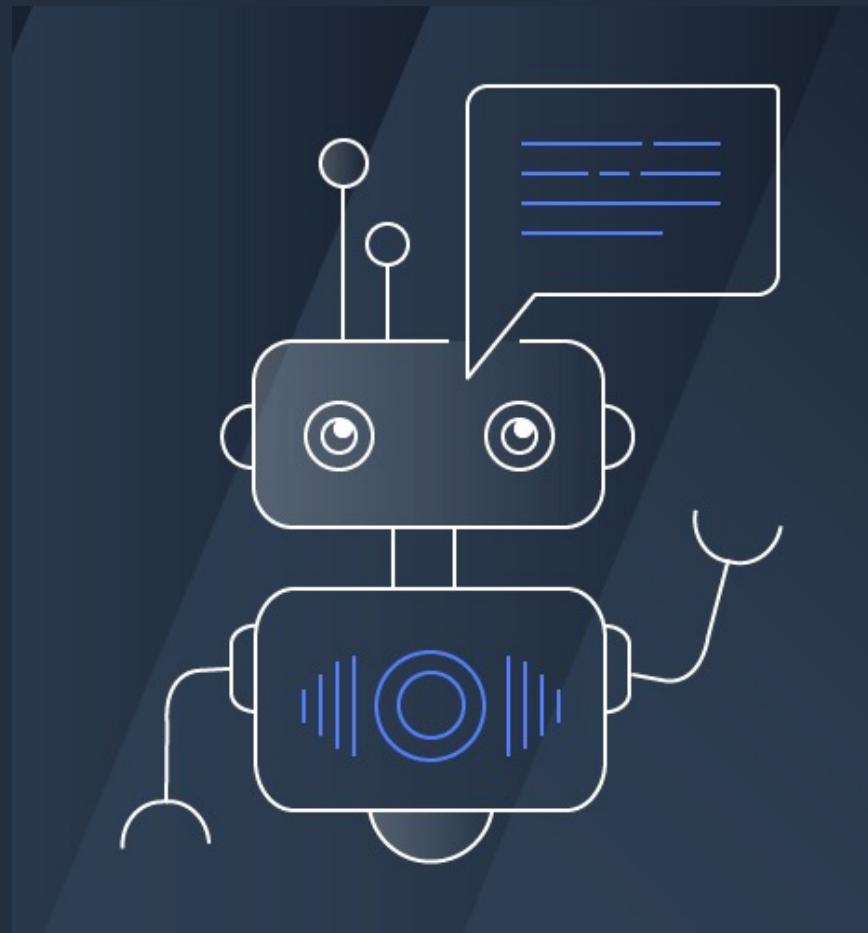
Coming soon

AWS SDKs simplifies the use of AWS services



- Higher level abstractions, saving developers time to concentrate on logic rather than low level API calls
- Best practices by default (e.g., retries, credential handling)

AWS Chatbot can now run commands



AWS Chatbot

Interactive agent for ChatOps on AWS

Receive notifications

Run commands for diagnostic information

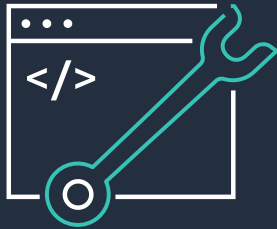
Predefined IAM policy templates

Support for Slack and Chime



Building modern applications on AWS

RESOURCES CREATED BY THE EXPERTS AT AWS TO HELP YOU BUILD AND VALIDATE DEVELOPER SKILLS



Enable rapid innovation by developing your skills in designing, building, and managing modern applications



Learn to modernize your applications with free digital training and classroom offerings, including Architecting on AWS, Developing on AWS, and DevOps Engineering on AWS



Validate expertise with the AWS Certified DevOps – Professional or AWS Certified Developer – Associate exams

Visit the developer learning path at aws.amazon.com/training/learn-about/developer/

Resources

AWS CI/CD Workshop - <https://aws-ci-cd.workshop.aws/>

CodePipeline Tutorial - <https://docs.aws.amazon.com/codepipeline/latest/userguide/tutorials.html>

- Review Youtube Video on CodePipeline - <https://www.youtube.com/watch?v=zMa5gTLrzmQ>

CodeBuild Tutorial - <https://docs.aws.amazon.com/codebuild/latest/userguide/getting-started.html>

- Review Youtube Video on Building CodeBuild Locally - <https://youtu.be/N3pW4ZCeCxA>

CodeDeploy Tutorials - <https://docs.aws.amazon.com/codedeploy/latest/userguide/tutorials.html>

- Review Youtube Video on Blue/Green Deployment with CodeDeploy - <https://www.youtube.com/watch?v=xThOQuhJ2Pw>

CodeCommit Tutorial - <https://docs.aws.amazon.com/codecommit/latest/userguide/getting-started-cc.html>

- Review Youtube Video on CodeCommit - <https://www.youtube.com/watch?v=SWqh7LvXKqI>

CodeArtifact Tutorial - <https://docs.aws.amazon.com/codeartifact/latest/ug/getting-started.html>

- Review Youtube Video on CodeArtifact - <https://www.youtube.com/watch?v=pxV5E83S7Bw>

CDK Workshop - <https://cdkworkshop.com/>

Code Review and App Performance with CodeGuru Workshop - <https://codeguru-codereview-appperformance.workshop.aws/en/>

AWS CI/CD pipeline with CodeGuru & UnitTest to improve code quality - <https://codequality.workshop.aws/en/>

CI/CD for Serverless Applications using SAM - <https://cicd.serverlessworkshops.io/>

Complete CI/CD with AWS CodeCommit, AWS CodeBuild, AWS CodeDeploy, and AWS CodePipeline - <https://aws.amazon.com/blogs/devops/complete-ci-cd-with-aws-codecommit-aws-codebuild-aws-codedeploy-and-aws-codepipeline/>



Thank you !

