AWS re:Invent

NOV. 29 – DEC. 3, 2021 | LAS VEGAS, NV

DOP306

Implementing release management strategies for CI/CD

Brian Beach
Principal Solutions Architect
AWS

Viyoma Sachdeva
Senior DevOps Consultant
AWS

Sandeep Gawande
Consultant
AWS

Pragtideep Singh
Senior Consultant
AWS

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Workshop agenda

1) Presentation and setup (30 minutes)
2) Workshop (90 minutes)
3) Q&A (15 minutes)

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

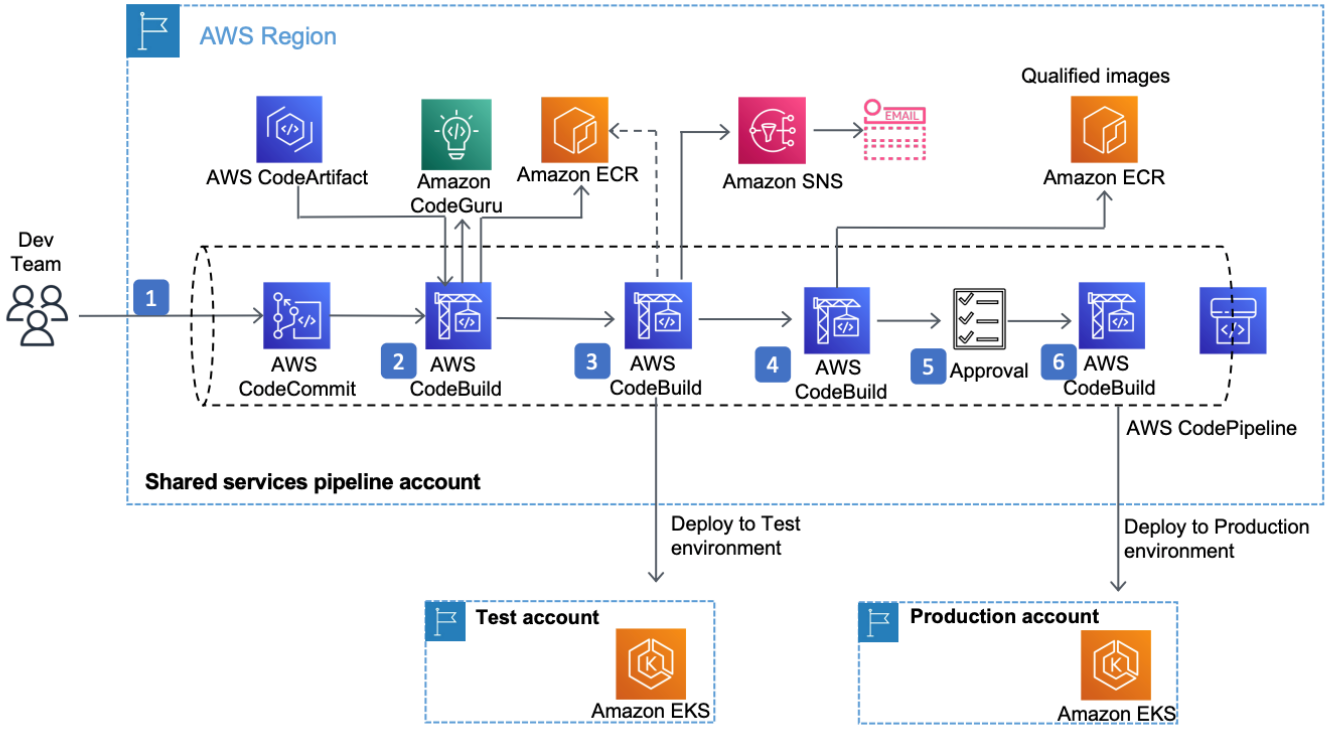
Workshop architecture

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Reference architecture

AWS CodePipeline to qualify and deploy a microservice

Integrating best practices in qualifying and promoting images using AWS managed services



* Please check the support for programming languages

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

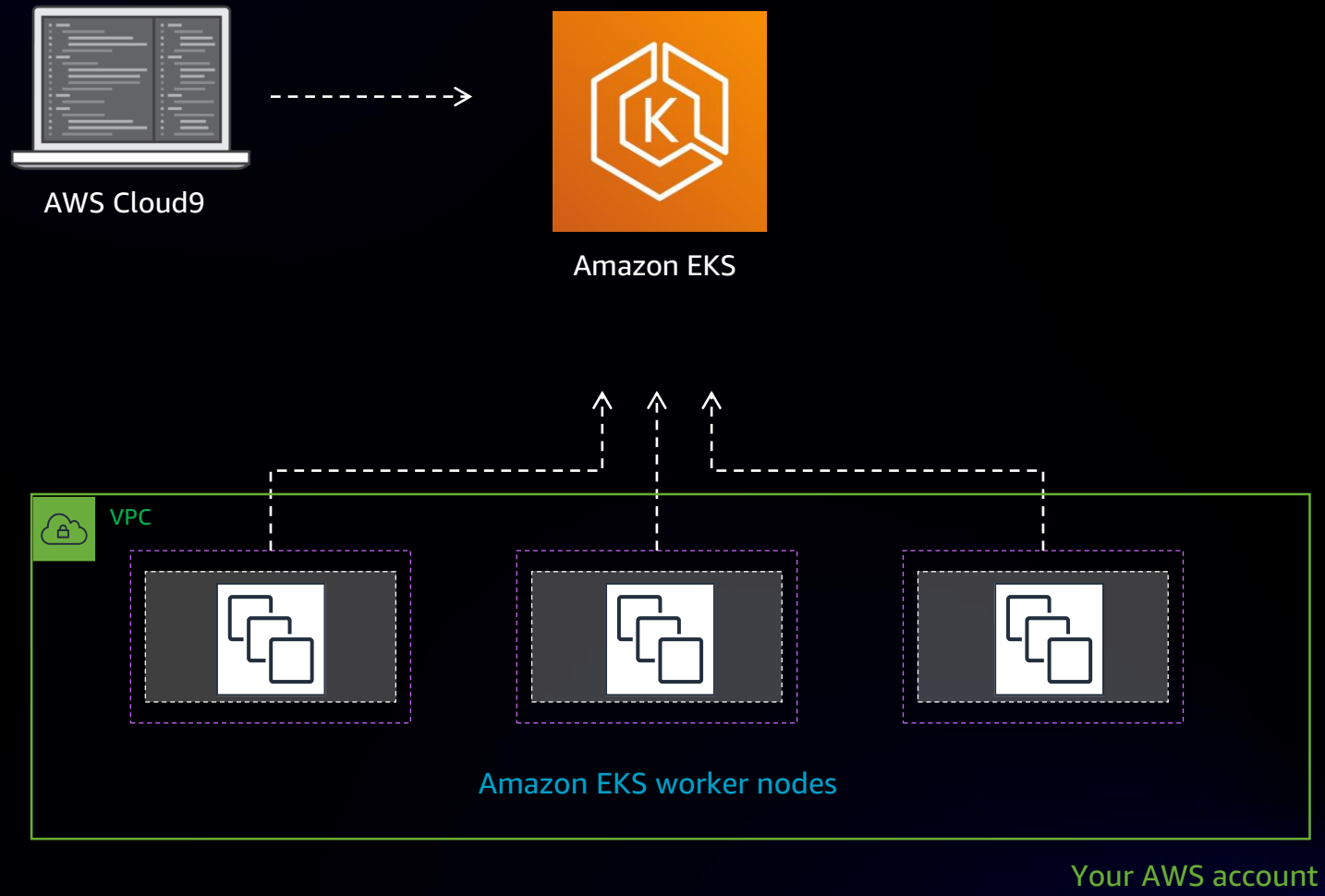
AWS Reference Architecture

CI/CD Pipeline for deploying to EKS

1. **Code Push**: Developer pushes the code to CodeCommit repository, which triggers an EventBridge rule to kick-off the AWS CodePipeline execution.
2. **Build and Containerize:** The image is built along with dependencies that are downloaded through AWS CodeArtifact (artifact repository).
   Security analysis is recommended here with static code analysis (e.g. Amazon CodeGuru*) and software composition analysis for CVE vulnerabilities.
   Later the image is containerized and pushed to an ECR repository that is configured with "Scan On Push" setting.
3. **Validate and Deploy to Test:** This stage will make an API call to ECR to validate the scan results. It will fail the build in case of critical CVE vulnerabilities detected by ECR and send an SNS message notification.
   Additionally, it can be configured to open a ticket with an incident management system. If the scan results successfully passes the criteria, it would deploy the image to the test environment. The deployment could be using kubectl OR helm commands.
4. **Test and Promote:** Conduct testing on the Test environment (integration, functional, penetration). You may also run Dynamic Analysis to evaluate application performance (e.g., CodeGuru Profiler*) and security. If all tests are successful, promote the image by re-tagging the container image and pushing it to the Production ECR repository. This repository holds qualified images, in case need to rollback.
5. **Approval (Optional)**: If needed to conduct any manual testing / verification on the Test environment, you can have a manual approval gate before moving to the Production deploy stage.
6. **Deploy to Production:** Deploy the image from Production ECR to production environment and run sanity tests. If the deployment or tests fail, rollback to an earlier version OR latest image from the Production ECR. You may also monitor application performance (e.g., CodeGuru profiler*).

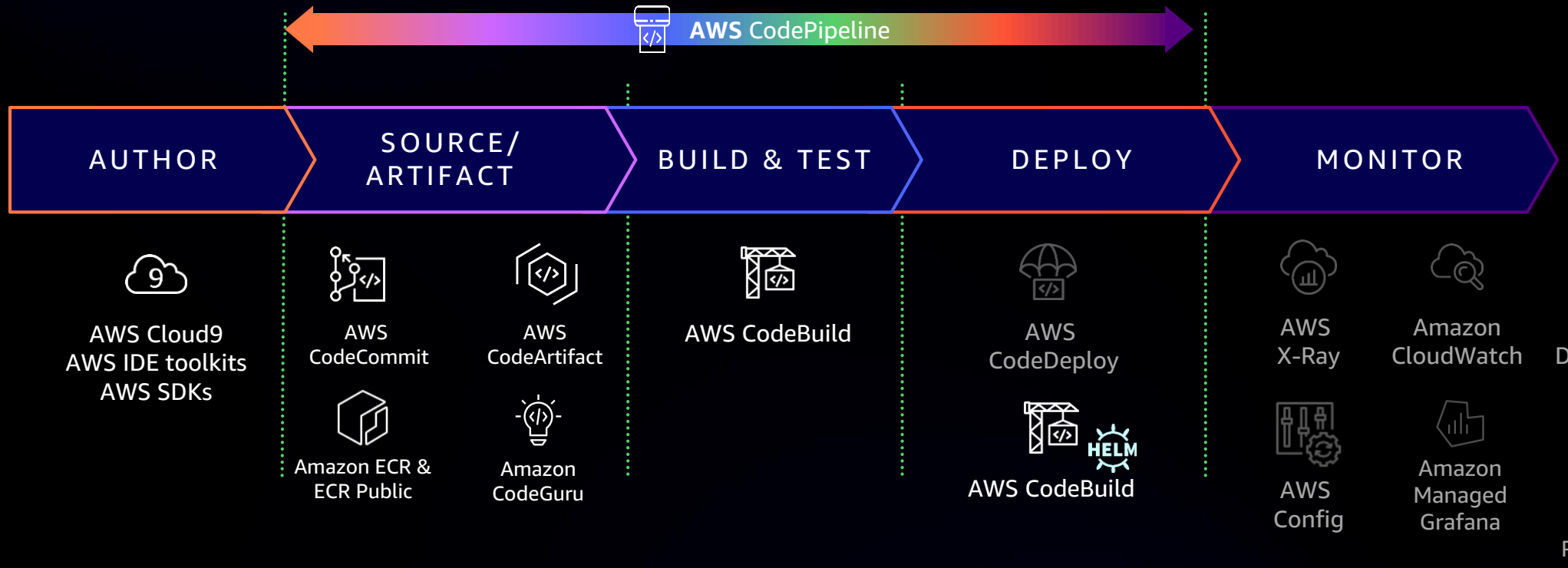
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon EKS architecture



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

CI/CD pipeline



MODEL **AWS** CloudFormation | AWS Cloud Development Kit (AWS CDK) | AWS Serverless Application Model (AWS SAM)

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Configuration



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Configuration

Let’s complete the first module together

https://bit.ly/tbd

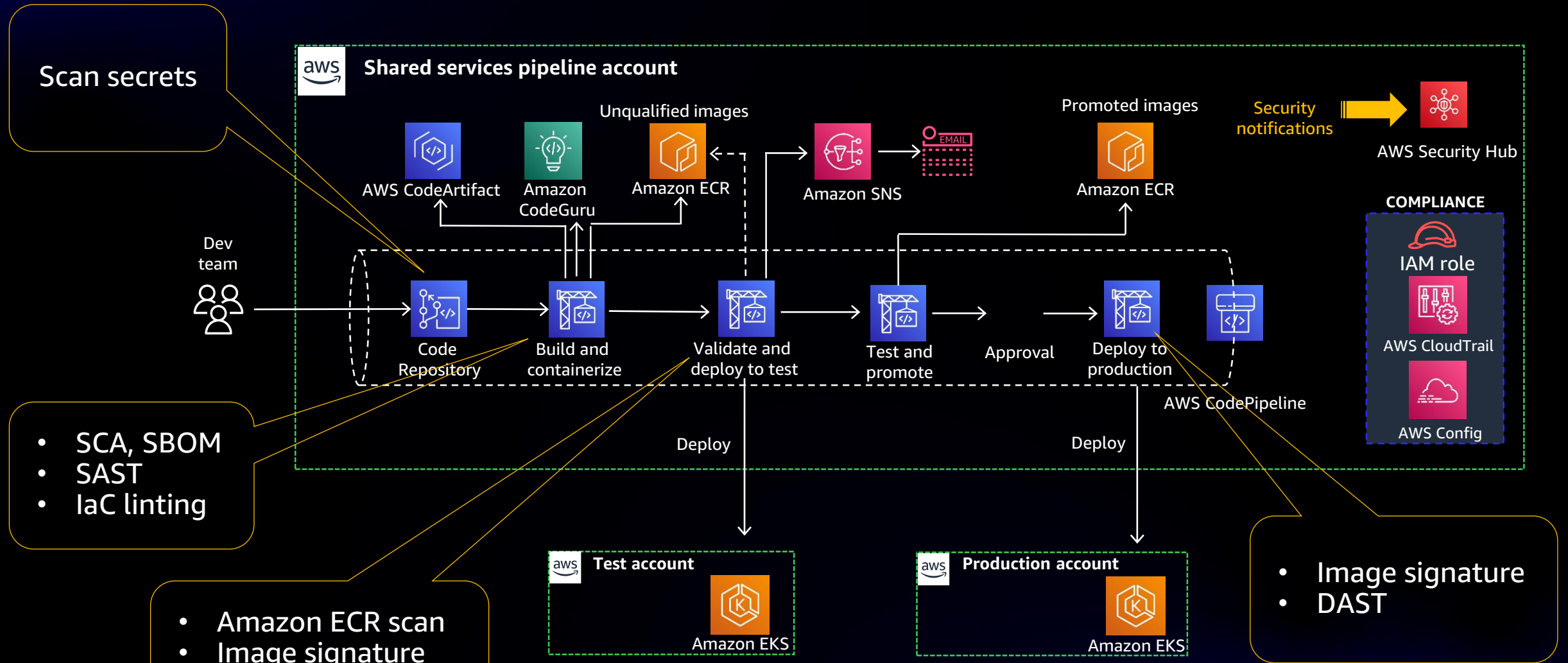


© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Walkthrough

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Integrating security checkpoints



AWS reference architecture

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

1) Code push/pull request

- Developer pushes to CodeCommit or approves a pull request
- CodeGuru performs an automated code review
- Amazon EventBridge triggers CodePipeline

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

2) Build and containerize

- Code is compiled using dependencies from CodeArtifact
- Unit tests and static code analysis are executed
- Container is built and pushed to Amazon ECR **test** repository
- Amazon ECR scans for common vulnerabilities and exposures (CVE)

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

3) Validate and deploy to test

- Validate CVE scan results using Amazon ECR API
- Deploy to Amazon EKS test environment/namespace using helm
- Wait for successful deployment and rollback on failure

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

4) Test and promote

- Conduct integration, functional, penetration, etc., testing
- Check for image integrity
- Runtime analysis (e.g., CodeGuru Profiler) is recommended
- Push to Amazon ECR production repository

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

5) Manual approval (optional)

- Pipeline is paused waiting for manual review/approval

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

6) Deploy to production

- Deploy to Amazon EKS production environment/namespace using helm
- Wait for successful deployment and rollback on failure

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Release management best practices summary

- Development workflow practices
- Pipeline workflow practices
  - Artifact management
  - Release promotion
  - Deployment and rollback mechanisms
- Multiple compute types
- Monitoring workflow practices

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Workshop

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Workshop

Complete the remaining steps on your own

Raise your hand if you need help

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

References

AWS DevOps Monitoring Dashboard
https://aws.amazon.com/solutions/implementations/aws-devops-monitoring-dashboard/

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.