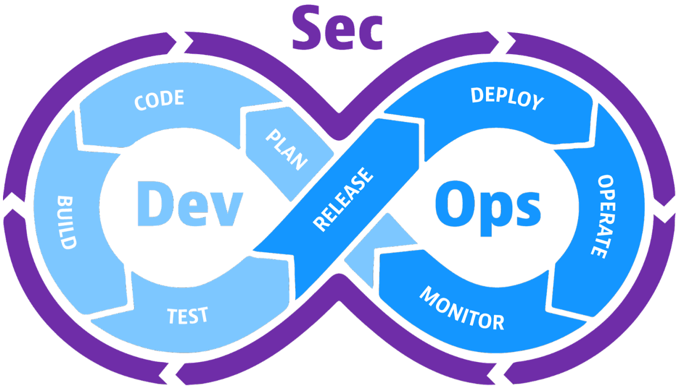
April 2024

Building a Secure DevOps Pipeline Requirements

Requirements



Plan

- **Tools:**
    - **Jira**: Used for project management, task tracking, and sprint planning.
- **Requirements:**
    - **Functionality**: Integrate Jira with GitHub to link code commits and branches with Jira tasks and user stories, ensuring traceability from planning to deployment.
    - **Security**: Utilize Jira's security features to control access based on roles, ensuring that sensitive information is protected.
    - **Integration Points**: Link Jira workflows directly to CI/CD triggers in AWS CodePipeline for automated progression of tasks through stages based on code commits and deployment statuses.

Code

- **Tools:**
    - **GitHub**: Manages source control.
    - **Git**: Manages version control.
    - **VSCode**: Recommended IDE.
- **Requirements:**

    - **Functionality**: Implement branch policies to manage code reviews and pull requests effectively. Automate static code analysis using GitHub Actions to scan pull requests.
    - **Security**: Integrate AWS Secrets Manager to manage and access secrets securely without hardcoding them in source files.
    - **Integration Points**: Set up GitHub to trigger AWS CodeBuild for continuous integration upon pull request approval.

Build

- **Tools:**
    - **AWS CodeBuild**: Manages the build process in a secure, scalable environment.
- **Requirements:**
    - **Functionality**: Configure AWS CodeBuild projects to compile code, run unit tests, and produce artifacts. Utilize Docker containers to ensure consistent environments.
    - **Security**: Implement encryption for building artifacts using AWS Key Management Service (KMS).
    - **Integration Points**: Use build specifications in AWS CodeBuild to integrate with AWS CodePipeline for a seamless transition from build to test stages.

Test

- **Tools:**
    - Open-source tools for SCA and SAST, like SonarQube.
    - Dynamic testing tools like OWASP ZAP.
- **Requirements:**
    - **Functionality**: Integrate automated testing within the CI/CD pipeline, including security tests at each stage.
    - **Security**: Conduct both static (SAST) and dynamic (DAST) security scans during the build or pre-deployment phases to identify vulnerabilities.
    - **Integration Points**: Ensure testing results from SonarQube and ZAP feed directly into the decision-making process for deployment readiness in AWS CodePipeline.

Release

- **Tools:**
    - **AWS CodePipeline**: Manages the release process.
- **Requirements:**
    - **Functionality**: Automate release pipelines to manage deployments across different environments (development, test, staging, production).
    - **Security**: Set up mandatory approval processes in AWS CodePipeline for releases, especially for production.
    - **Integration Points**: Ensure audit trails via integration with AWS CloudTrail for compliance and security reviews.

Deploy

- **Tools:**

    - **AWS CodeDeploy**: Automates application deployments.
- **Requirements:**
    - **Functionality**: Configure deployment strategies like blue-green and canary to minimize downtime and risk.
    - **Security**: Use AWS CodeDeploy to manage permissions strictly using AWS IAM roles.
    - **Integration Points**: Monitor deployments using AWS CloudWatch to ensure successful rollouts and quick issue identification and resolution.

Operate

- **Tools:**
    - **Amazon CloudWatch**
- **Requirements:**
    - **Functionality**: Define and provision infrastructure using AWS CloudFormation and AWS CDK. Automate build, test, and deployment stages using AWS CodePipeline.
    - **Security**: Use AWS Identity and Access Management (IAM) for secure access control.
    - **Integration Points**: Continuous monitoring with feedback loops to AWS Systems Manager for operational resilience and security compliance.

Monitor

- **Tools:**
    - **Amazon CloudWatch**: Used for monitoring and operational health insights.
    - **AWS CloudTrail**: Tracks user activity and API usage.
- **Requirements:**
    - **Functionality**:
        - Configure Amazon CloudWatch to collect and track metrics, set alarms, and automatically react to changes in AWS resources.
        - Use AWS CloudTrail to enable governance, compliance, and operational and risk auditing of the AWS account.
    - **Security**:
        - Ensure all logs in CloudWatch are encrypted at rest using AWS KMS.
        - Configure CloudTrail to log all API calls, which serves as a critical component of the security audit trail.
    - **Integration Points**:
        - Integrate CloudWatch with AWS Lambda for real-time monitoring of automated tasks within the CI/CD pipeline.
        - Link CloudWatch and AWS CodePipeline to track the deployment status and trigger alerts on failure.