GSAW 2022

Earth Observing Data Distribution and Visualization Using the Cloud - DevSecOps

Author:
Jay Pennington

Co-Authors:
Will Coffey
Spencer Drakontaidis
Morgan Williams
Alan Christopher
Phillip Jasper

February 23, 2022

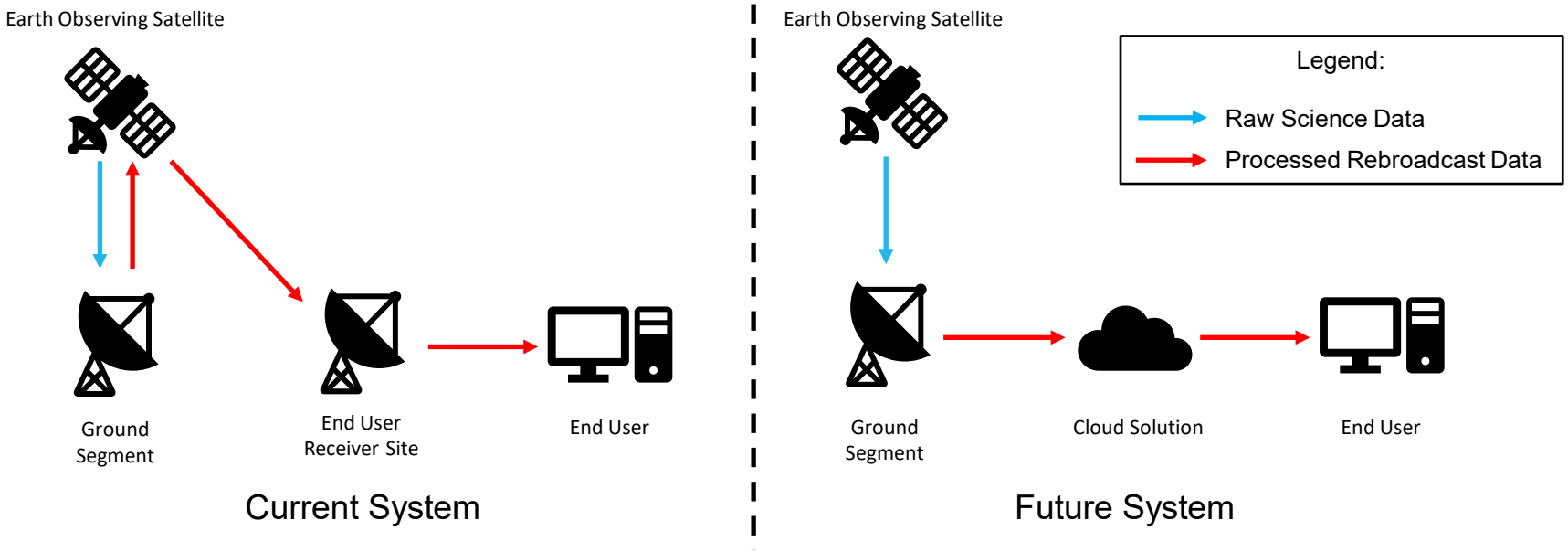
DevSecOps Introduction

- Development, Security, Operations (DevSecOps)
  - *A holistic approach that combines software development, security, and IT operations*
- Continuous Integration / Continuous Deployment (CI/CD) Pipeline
  - *Series of linked tools that provide rapid, automated, repeatable software deployment*
- Infrastructure as Code (IaC)
  - *Defining and deploying IT infrastructure using machine-readable scripts*
- DevSecOps is more than just using tools, it is a cultural change!
  - *Effective implementation requires buy-in from all stakeholders*

But…Why DevSecOps?

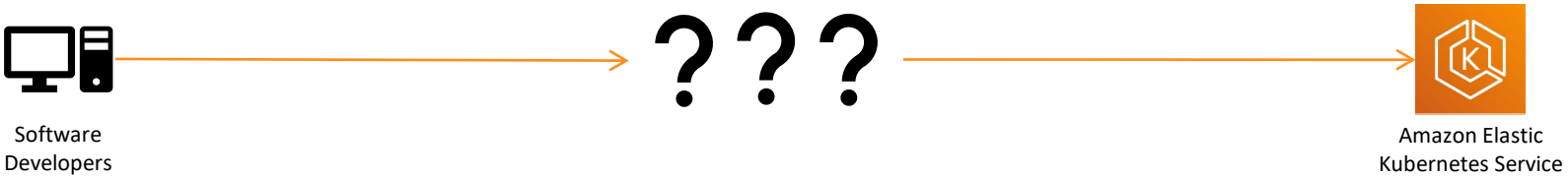
- Customer needs and timelines require faster software development cycles
  - *Holistic approach allows problems to be fixed much earlier in the software lifecycle*
  - *Automated pipelines allow deployment in seconds/minutes instead of days/weeks*
- Customers should focus on competitive advantages
  - *Software development offers competitive advantage, but deployment does not*
  - *Automating deployment allows developers to spend more time writing software*
- Product quality is improved
  - *Automated, repeatable processes reduces the likelihood of human error*
  - *Automated security and quality scans provide feedback to developers*

Satellite Data Rebroadcast



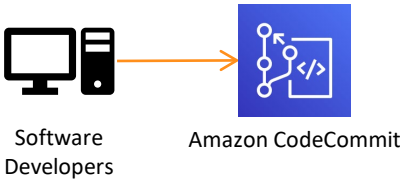
- Mission: earth observing science data streaming
  - *Data is collected, processed, and rebroadcast to end users*
- Problem: both satellite and end user require dedicated antennas
  - *Satellite rebroadcast and end user receiver antennas can be costly*
- Hypothesis: processed data can be rebroadcast via the cloud
  - *Remove rebroadcast antenna to reduce size, weight, and power on next-gen satellites*
  - *Remove the need for end users to procure and maintain receiver antennas*
- This use case is illustrative for the purposes of this presentation
  - *DevSecOps can be used for all kinds of software development!*

Cloud Software Deployment



- Goal: deploy software written by developers into cloud infrastructure
  - *We chose to use Kubernetes as our development and deployment environment*
    - Open-source orchestration tool for management of multiple applications
    - Provides infrastructure resiliency via cluster of multiple nodes
    - Automatically scale applications in real-time
  - *Amazon Elastic Kubernetes Service (EKS) is a managed Kubernetes service*
    - Amazon manages cluster nodes, redundancy, and patches for a fee
    - Create a Kubernetes cluster in a matter of minutes using automated scripts
- What process can we use to deploy our applications into Kubernetes?
  - *Traditional: developers manually go through several deployment procedures*
  - *DevSecOps: create an automated CI/CD deployment pipeline*

Stage 1: Amazon CodeCommit



- Amazon CodeCommit is a git-based version control service
  - *Provides central repository to track file version history*
  - *Allows developers to work simultaneously on the same application*
  - *Provides one source of truth for all project artifacts*
- All project resources are tracked in CodeCommit
  - *Application source code*
    - Go code, Python code, Java code
  - *Infrastructure as code*
    - Crossplane scripts, Kubernetes manifests, Helm charts
- In our pipeline: software developers push their code to CodeCommit
  - *This invokes the rest of the automated CI/CD pipeline*

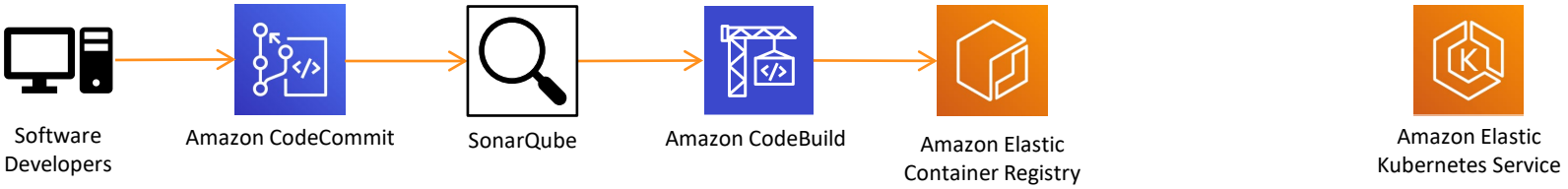
Stage 2: SonarQube

- SonarQube is an open-source static analysis tool
  - *Reads source code to find bugs, security vulnerabilities, and code smells*
  - *Works with most common programming languages*
- SonarQube findings are provided as feedback to developers
  - *Provides detailed findings on ways to improve code quality and security*
  - *Provides consistent time estimates for fixes (accuracy may vary)*
  - *Allows development teams to monitor trends over time*
- Quality thresholds can be set as part of CI/CD pipeline
  - *Code that does not meet standards is automatically blocked from deployment*
- Can integrate with other quality/security tools with custom metrics
- In our pipeline: CodeCommit code is sent to SonarQube
  - *Code must pass all quality/security thresholds before moving to next step*

Stage 3: Amazon CodeBuild

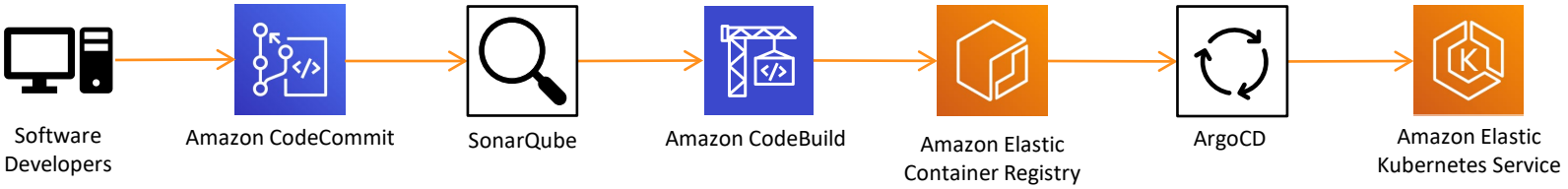
- Amazon CodeBuild is a pipeline orchestration tool
  - *It ties the pipeline together by triggering other stages*
- CodeBuild creates artifacts
  - *Builds executables from source code*
  - *Builds containers for executables*
    - Containers bundle application with required dependencies
    - Containers allow portability between environments
    - Containers are built step-by-step in “layers”
- In our pipeline: code that has passed SonarQube tests is built by CodeBuild
  - *Source code is compiled into executable applications*
  - *Applications and dependencies are bundled into containers*

Stage 4: Amazon Elastic Container Registry



- Amazon Elastic Container Registry (ECR) is a container version control service
  - *Container images are hosted here*
  - *Services can push or pull container images from the registry*
- Clair container scanning
  - *Open-source security scanning tool for Docker containers*
  - *Inspects Docker images layer-by-layer for known vulnerabilities*
  - *Integrates natively with Amazon ECR*
- In our pipeline: CodeBuild pushes containers to ECR
  - *Containers must pass Clair scans before moving to next step*

Stage 5: ArgoCD



- ArgoCD synchronizes actual cluster state with desired cluster state
  - *Synchronization can be trigged as needed*
  - *Synchronization can occur on regular intervals (e.g., every 5 minutes)*
- Synchronization has many benefits
  - *Add desired deployments to the cluster*
  - *Remove undesired deployments from the cluster*
  - *Prevent drift of actual state away from desired state*
- In our pipeline: ArgoCD is invoked to synchronize cluster state
  - *ArgoCD deploys new/modified containers from ECR to the Kubernetes cluster*

Impact of DevSecOps

- Holistic approach to software development
  - *Two developers, one security expert, and one operations expert work side-by-side*
  - *Several key problems were fixed much earlier in the development process*
- Deployment process is streamlined
  - *Software now deploys in as little as 30 seconds*
  - *Automated, repeatable processes prevented human errors that previously occurred*
  - *Security/quality checks uncovered key bugs, vulnerabilities, and code improvements*
- Development team can focus on competitive advantage
  - *Developers now trigger deployment seamlessly as part of development*

Lessons Learned

- AWS tools were mostly easy to integrate with each other
- Amazon CodeCommit lacks several key features for developers
  - *No way to assign individual code reviewers*
  - *No way to notify specific people of events (such as code approvals)*
  - *Custom Lambda scripts were written to solve these issues*
- Amazon CodeBuild integration was sometimes difficult
  - *Experienced networking problems when integrating with SonarQube*
  - *Extra network components were deployed to solve these issues*
- Amazon CodeDeploy lacks a key pipeline feature
  - *No way to directly link CodeBuild with Elastic Kubernetes Service*
  - *ArgoCD (third-party tool) was deployed to solve this issue*
- AWS technical support provided fast and effective assistance

Further Exploration

- This presentation is just a small sample of DevSecOps possibilities
  - *Some CI/CD pipelines have more than a dozen tools*
- Many other tools exist for each stage
  - *CI/CD pipeline tools: Jenkins, Gitlab, Azure Pipelines, Google Cloud Build*
  - *Repository tools: Bitbucket, Azure Repos, Google Cloud Source Repositories*
  - *Static analysis tools: Fortify, CodeSonar, Checkmarx*
  - *Container registry: Harbor, Azure Container Registry, Google Container Registry*
  - *Container scanning tools: Anchore, Aqua Security, StackRox, Prisma*
  - *There are many other stages/tools. Explore and experiment!*
- There are different ways to get tools
  - *Host on premises or in the cloud*
  - *Vendor-managed tools or self-managed tools*


Thank You!