

Solution Architecture

AWS native DevOps Platform

V1 | February 18, 2019

Raghuveer Varahagiri

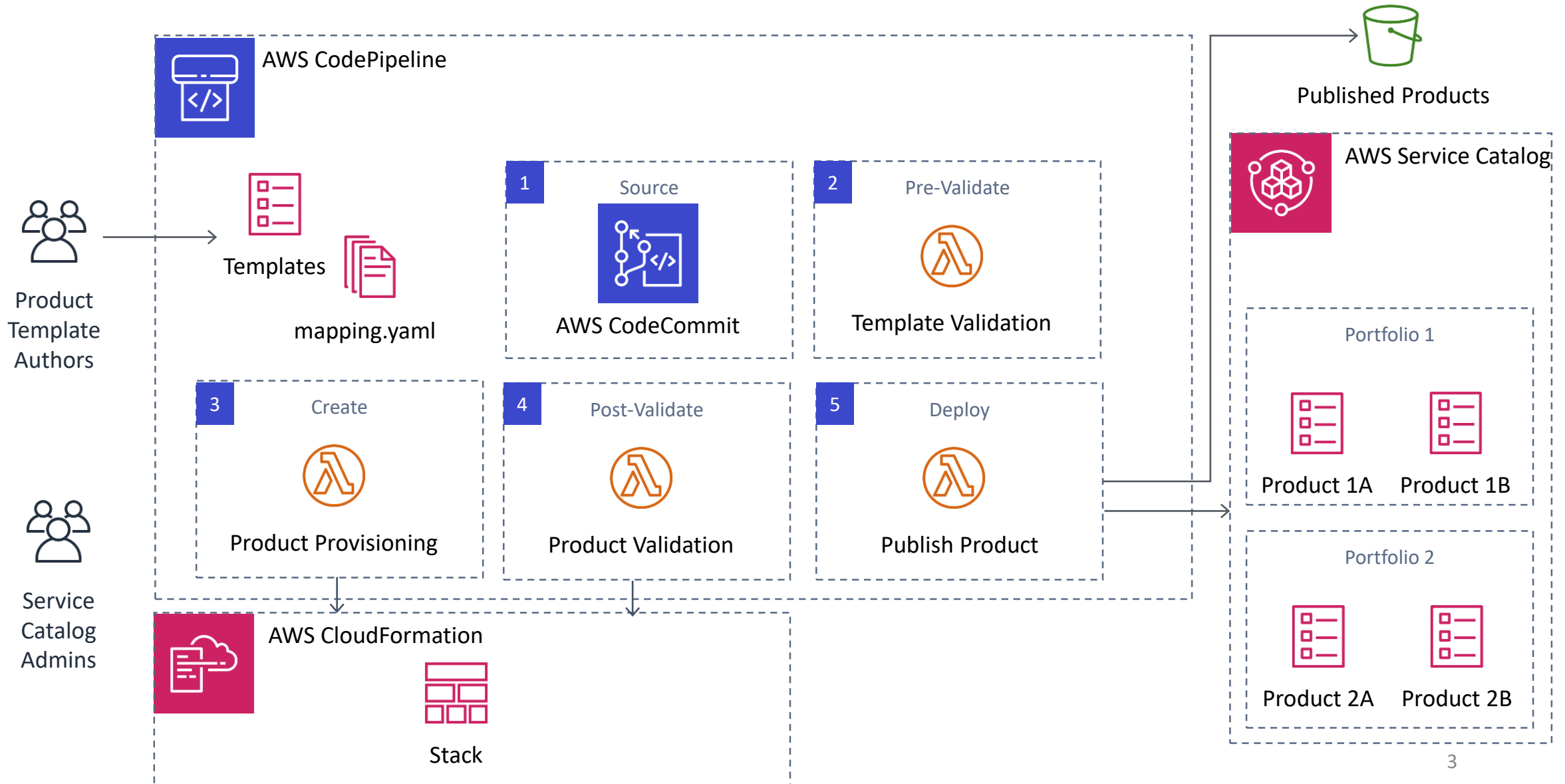


High-Level Architecture

1. [CI/CD Pipeline #1] CF templates → ServiceCatalog : (x1)
 - CodePipeline
 - Source: CloudFormation templates in a CodeCommit repo
 - Build: CloudFormation API
 - Test: CloudFormation, Lambda, cfn-nag / cfn-python-lint
 - Approve: Manual (optional)
 - Deploy: AWS Service Catalog
 - Users / Roles:
 - Service Catalog Admins, Product Template Authors, Catalog Users - Developers, App Admins
2. [CI/CD Pipeline #2] Build & Deploy a Java Web Application : (xN on-demand)
 - CodePipeline
 - Source: Java web application source code in CodeCommit or GitHub repo <parameter>
 - Build: Jenkins / Maven on EC2
 - Test: Blazemeter
 - Approve: Manual (optional)
 - Deploy: EC2
 - Users / Roles:
 - Developers, App Admins
 - Demo Application:
 - Java Web Application running on a EC2 / Apache tomcat
3. [Integrations – Alexa and ELK] :
 - Alexa integration – making monitoring reports / stats / alerts available via voice commands
 - ELK Stack – monitoring based on resource tags
4. [Integrations – SNS, CloudTrail, Alarms]
 - SNS Notifications – topics for budget alerts, infrastructure health, etc
 - CloudWatch – budget alerts (passive) based on resource tags

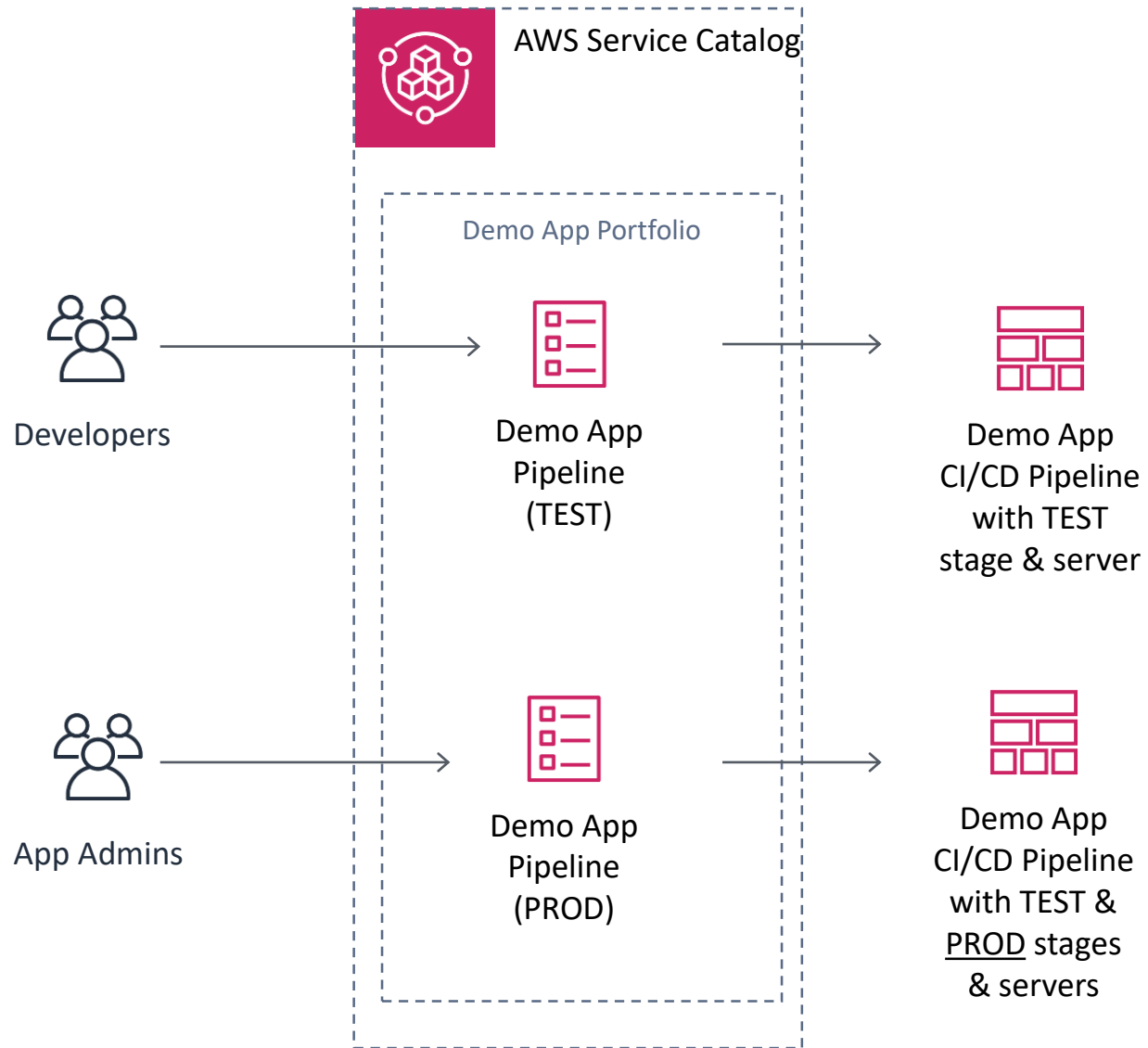
High-Level Architecture

[CI/CD Pipeline #1] CF templates → ServiceCatalog : (x1)



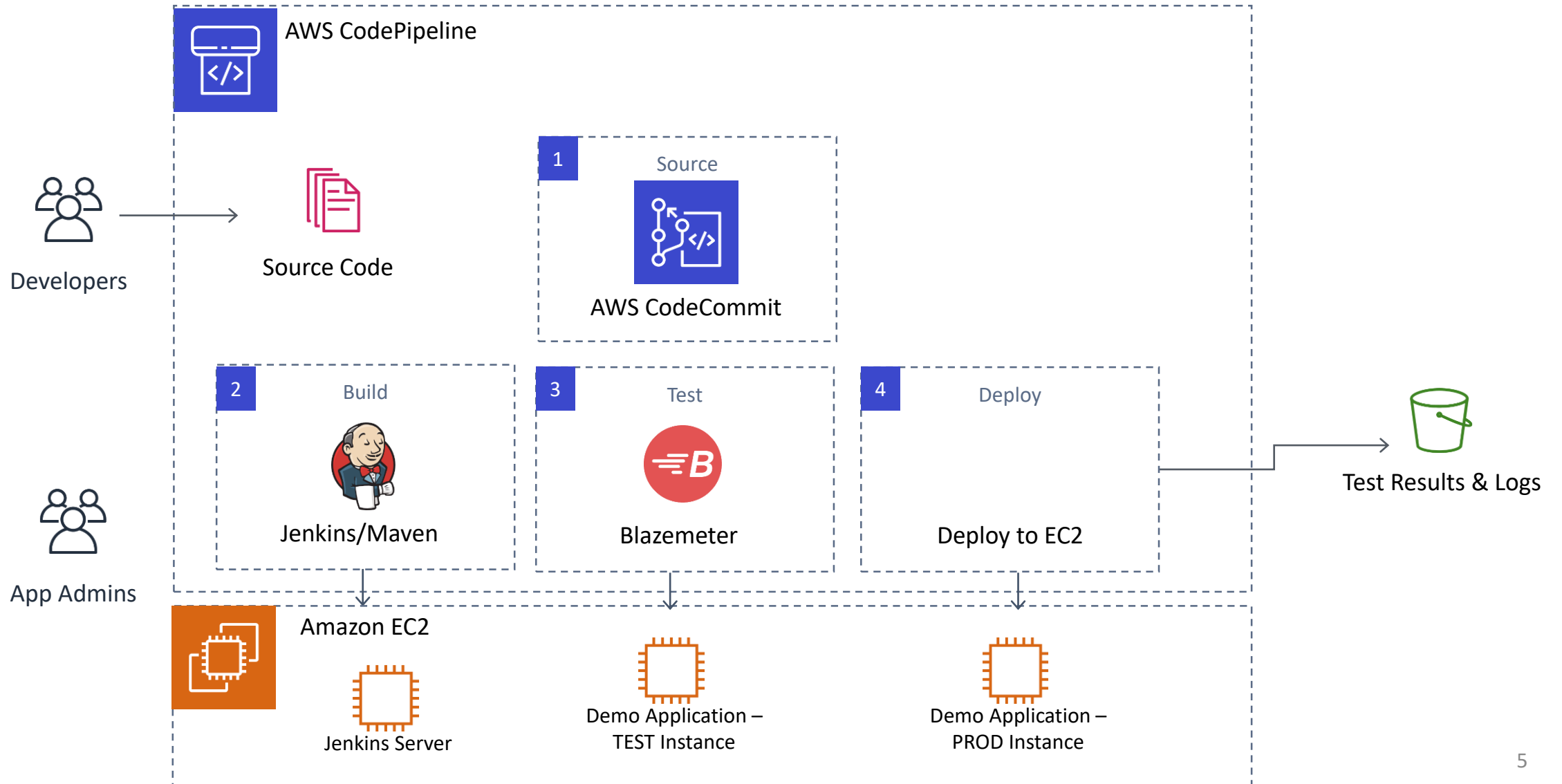
High-Level Architecture

Usage of AWS Service Catalog



High-Level Architecture

[CI/CD Pipeline #2] Build & Deploy a Java Web Application : (xN on-demand)



Notes

3. [Integrations – Alexa and ELK] :

- Alexa integration
- ELK Stack

4. [Integrations – SNS, CloudTrail, Alarms]

- SNS Notifications
- CloudWatch