AWS CICD：CodeCommit,CodePipeline,C...

CICD

Introduction

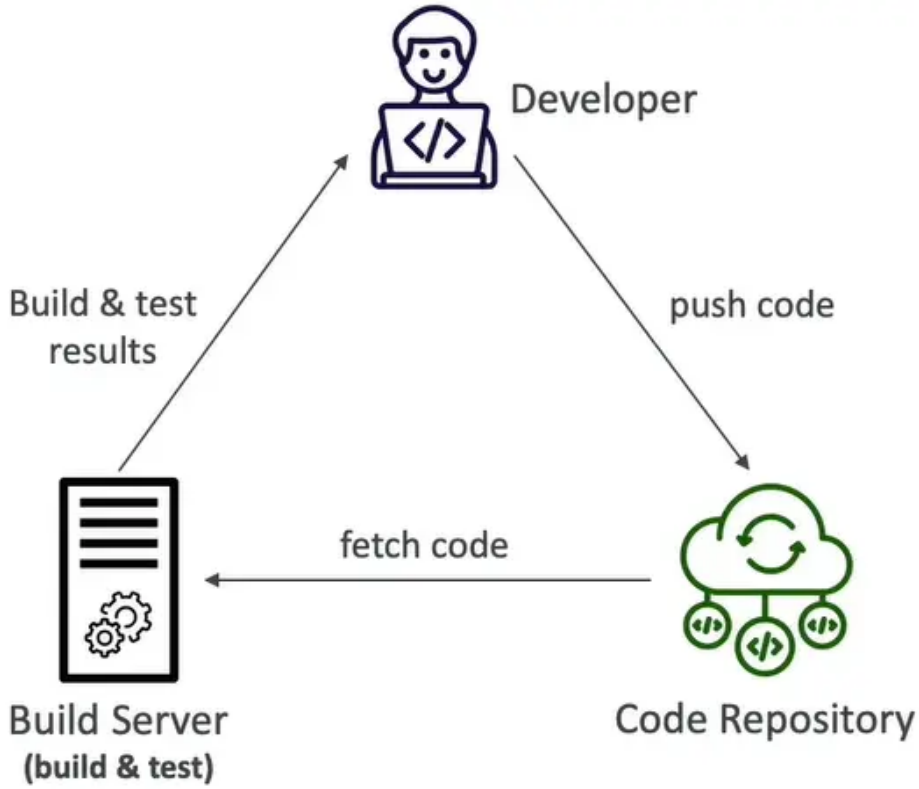
- we have learned how to
  - create AWS resource,manually (fundamentals)
  - interact [交互] with AWS programmatically (AWS CLI)
  - deploy code to AWS using Elastic Beanstalk
- all these manual steps make it very likely fro us to do mistakes

- we would like our code "in a repository" and have it deployed onto AWS
  - automatically
  - the right way
  - making sure it's tested before being deployed
  - with possibility to go into different stages (dev,test,staging,prod)
  - with manual approval [批准] where needed
- to be a proper [合格] AWS developer ... we need to learn AWS CICD

- this section is all about automating the deployment we've done so far [目前为止] while adding increased safety

- we'll learn about
  - AWS CodeCommit – storing our code
  - AWS CodePipeline [代码管道] – automating our pipeline code to Elastic Beanstalk
  - AWS CodeBuild – building and testing our code
  - AWS CodeDeploy – deploying the code to EC2 instances (not Elastic Beanstalk)
  - AWS CodeStar – manage software development activities in one place
  - AWS CodeArtifact – store,publish, and share softeware packages
  - AWS CodeGuru – automated code reviews using Machine Learning

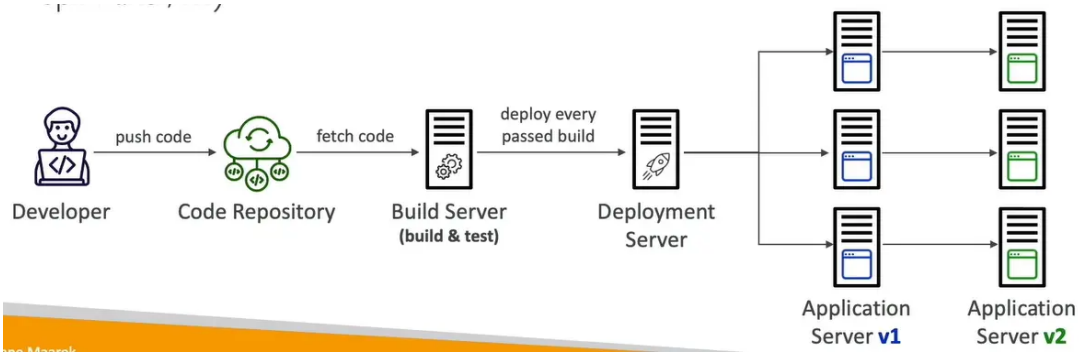
Continuous Integration (CI)



- developers push the code to a code repository often (eg.GitHub, CodeCommit,Bitbucket..)
- a testing / build server checks the code as soon as it's pushed (CodeBuild,Jenkins,CI..)
- the developers gets feedback about the tests and checks that have passed / failed

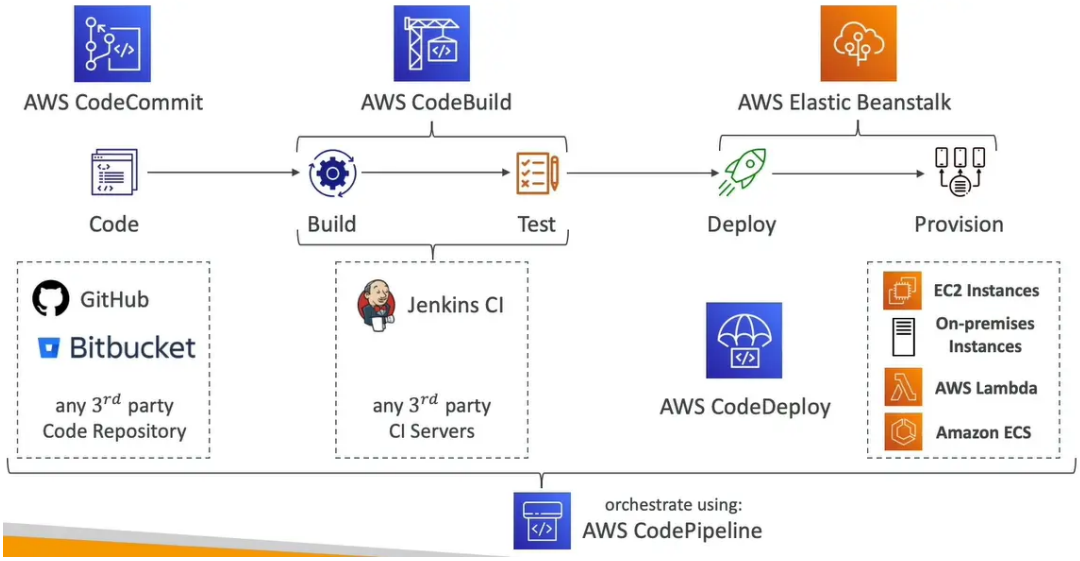
- find bugs early, then fix bugs
- deliver faster as the code is tested
- deploy often
- happier developers, as they're unblocked

Continuous Delivery [交付](CD)



- ensures that the software can be released [发布] reliably [可靠地] whenever needed
- ensures deployments happen often and are quick
- shift away from "one release every 3 months" to "5 releases a day"
- that usually means automated deployment (eg. CodeDeploy,Jenkins CD,Spinnaker...)

Technology Stack for CICD



AWS CodeCommit

- Version Control is the ability to understand the various changes that happened to the code over time (and possibly roll back)
- all these are enable by using a version control system such as Git
- a git repository can be synchronized on your compute, but it usually is uploaded on a central online repository
- benefits are
  - collaborate [合作] with other developers
  - make sure the code is back-up somewhere
  - make sure it's fully viewable and auditable [可审计的]

- git repositories can be expensive
- the industry include GitHub,GitLab,Bitbucket..
- and AWS CodeCommit
  - private Git repositories
  - no size limit on repositories (scale seamlessly [无缝地])
  - fully managed , highly available
  - code only in AWS Cloud account => increased security and compliance
  - Security (encrypted,access control...)
  - integrated with Jenkins, AWS CodeBuild, and other CI tools

Security

- interaction [交互] are done using Git (standard)
- Authentication [验证]
  - SSH Keys – AWS users can configure SSH keys in their IAM Console
  - HTTPS – with AWS CLI Credential helper or Git Credentials for IAM user
- Authorization [授权]
  - IAM policies to manage users / roles permissions to repositories
- Encryption
  - repositories are automatically encrypted at rest using AWS KMS
  - encrypted in transit [传输过程中] (can only use HTTPS or SSH – both secure)
- Cross-account Access

- do NOT share your SSH keys or your AWS credentials
- use an IAM Role in your AWS account and use AWS STS (AssumeRole API)

CodeCommit vs. GitHub

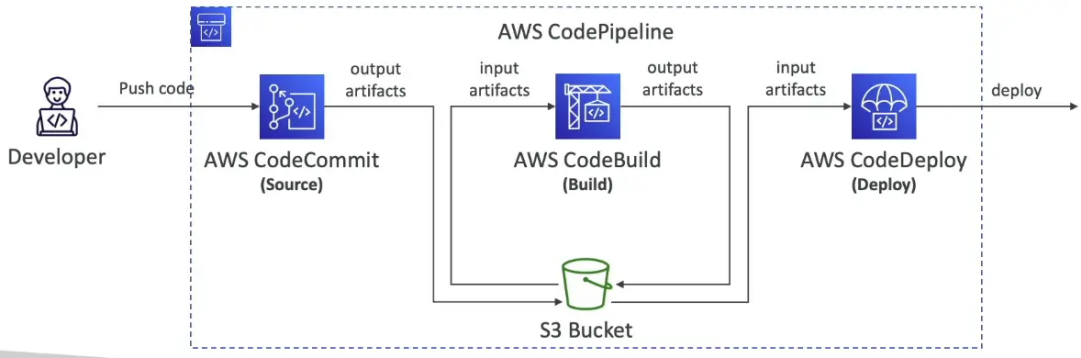
	CodeCommit	GitHub
Support Code Review (Pull Requests)	✔	✔
Integration with AWS CodeBuild	✔	✔
Authentication (SSH & HTTPS)	✔	✔
Security	IAM Users & Roles	GitHub Users
Hosting	Managed & hosted by AWS	- Hosted by GitHub - GitHub Enterprise: self hosted on your servers
UI	Minimal	Fully Featured

AWS CodePipeline

- Visual Workflow to orchestrate [编排] your CICD
- Source – CodeCommit, ECR,S3,Bitbucket,Github
- Build – CodeBuild, Jenkins,CloudBees,TeamCity
- Test – CodeBuild,AWS Device Farm,3rd party tools...
- Deploy – CodeDeploy, Elastic Beanstalk,CloudFormation,ECS,S3...
- Consist of stages:
  - each stage can have sequential [顺序的] actions and / or parallel actions
  - Example:
  - manual approval can be defined at any stage

Artifacts

- each pipeline stage can create artifacts
- Artifacts stored in an S3 bucket and passed on to the next stage



Troubleshooting

- for CodePipeline Pipeline/Action/Stage Execution State Changes
- use CloudWatch Events (Amazon EventBridge). Example
  - you can create events for failed pipelines
  - you can create events for cancelled stages

- if CodePipeline fails a stage, you pipeline stops, and you can get information in the console
- if pipeline can't perform an action, make sure the "IAM Service Role" attached dose have enough IAM permissions (IAM Policy)

- AWS CloudTrail can be used to audit AWS API calls

AWS CodeBuild

CodeBuild containers are deleted at the end of their execution (success or failure). You can't SSH into them, even while they're running.

You can configure CodeBuild to run its build containers in a VPC, so they can access private resources in a VPC such as databases, internal load balancers, ...

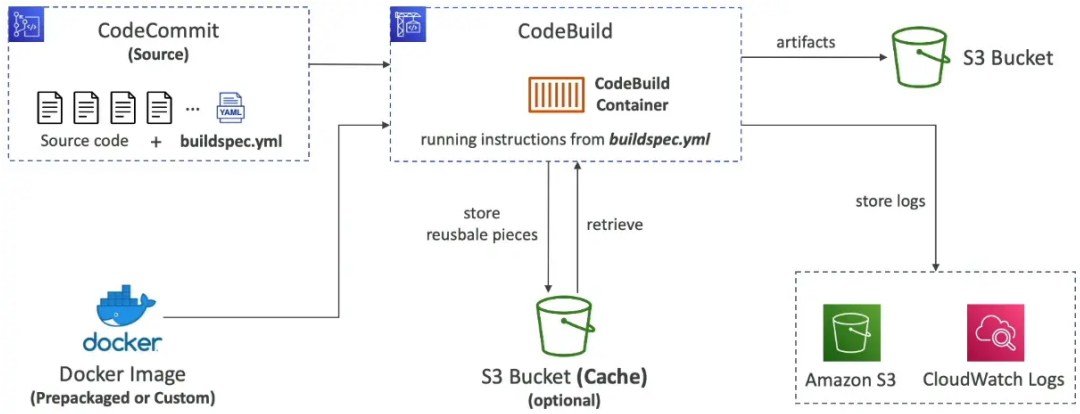
- Source – CodeCommit,S3,Bitbucket,Github
- Build instructions [说明]: Code file buildspec.yml or insert manually in Console
- Output logs can be stored in Amazon S3 & CloudWatch Logs
- use CloudWatch Metrics to monitor build statistics
- use EventBridge to detect failed builds and trigger notifications
- use CloudWatch Alarms to notify if you need "thresholds" for failures

- build projects can be defined within CodePipeline or CodeBuild

Supported Environments

- java
- ruby
- python
- go
- node.js
- android
- .net core
- php
- docker – extend any environment you like

how it works

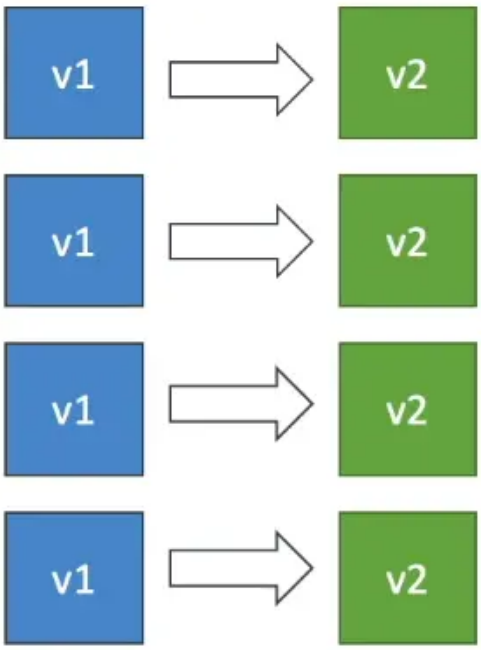


buildspec.yml

```yaml
version: 0.2

env:
  variables:
    JAVA_HOME: "/usr/lib/jvm/java-8-openjdk-amd64"
  parameter-store:
    LOGIN_PASSWORD: /CodeBuild/dockerLoginPassword

phases:
  install:
    commands:
      - echo "Entered the install phase..."
      - apt-get update -y
      - apt-get install -y maven
  pre_build:
    commands:
      - echo "Entered the pre_build phase..."
      - docker login -u User -p $LOGIN_PASSWORD
  build:
    commands:
      - echo "Entered the build phase..."
      - echo "Build started on `date`"
      - mvn install
  post_build:
    commands:
      - echo "Entered the post_build phase..."
      - echo "Build completed on `date`"

artifacts:
  files:
    - target/messageUtil-1.0.jar

cache:
  paths:
    - "/root/.m2/**/*"
```

- buildspec.yml file must be at the root of your code
- env – define environment variables
  - variables – plaintext [文本] variables
  - parameter-store – variables stored in SSM Parameter Store
  - secrets-manager – variables stored in AWS Secrets Manager
- phases [阶段] – specify commands to run
  - install – install dependencies you may need for your build
  - pre_build – final commands to execute before build
  - Build – actual build commands
  - post_build – finishing touches (eg,zip,output)
- artifacts – what to upload to S3 (encrypted with KMS)
- cache – files to cache (usually dependencies) to S3 for future build speedup

AWS CodeDeploy



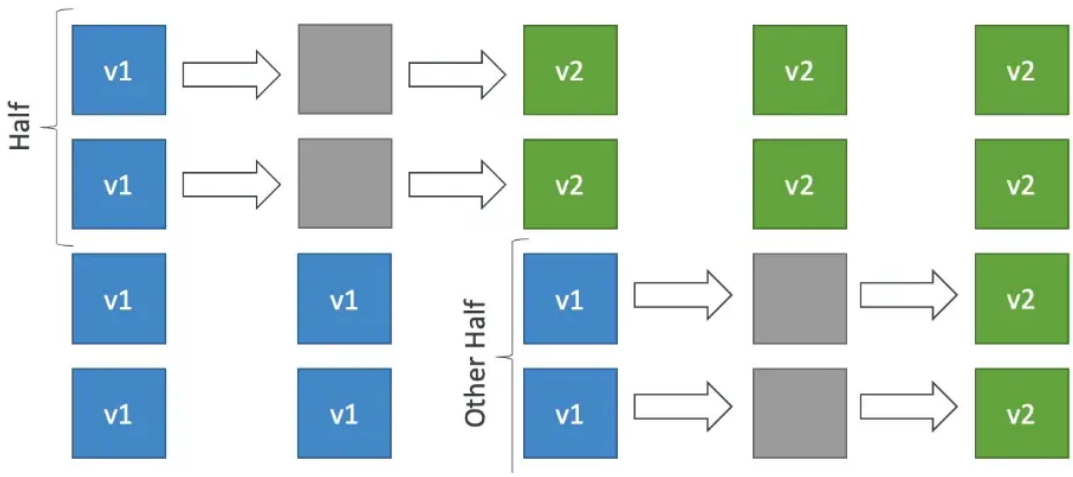
- deployment service that automates application deployment
- deploy new application versions to EC2 Instances,on-premisses services, Lambda functions, ECS services
- automated rollback capability in case of failed deployments , or trigger CloudWatch Alarm
- gradual [逐步] deployment control
- a file named appspec.yml defines how the deployment happens

EC2/ on-premises platform

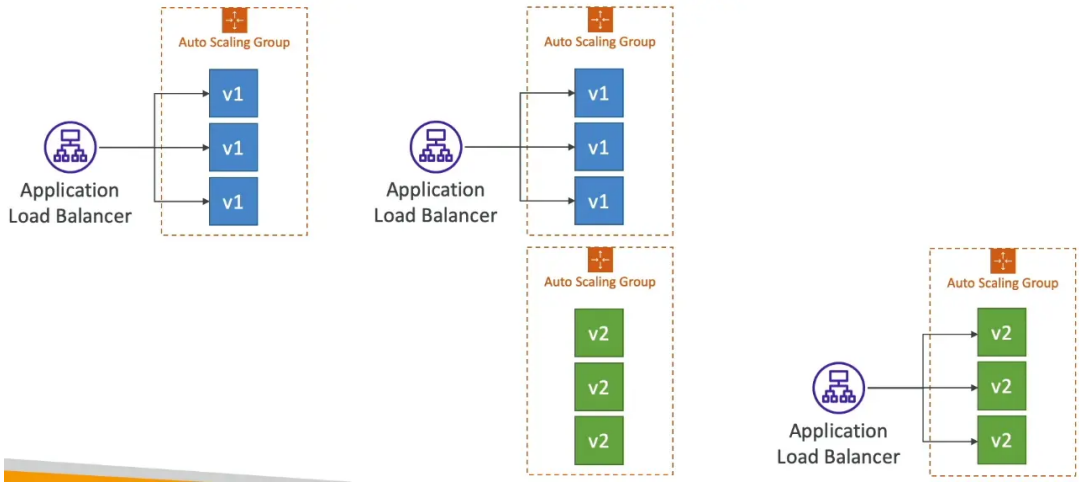
- can deploy to EC2 instance & on-premises servers
- perfom in-place [就地] deployments or blue/green deployments
- must run the CodeDeploy Agent on the target instances
- define deployment speed
  - AllAtOnce : most downtime
  - HalfAtATime: reduced capacity by 50%
  - OneAtATime: slowest,lowest availability impact
  - Custom: define your %

In-Place Deployment

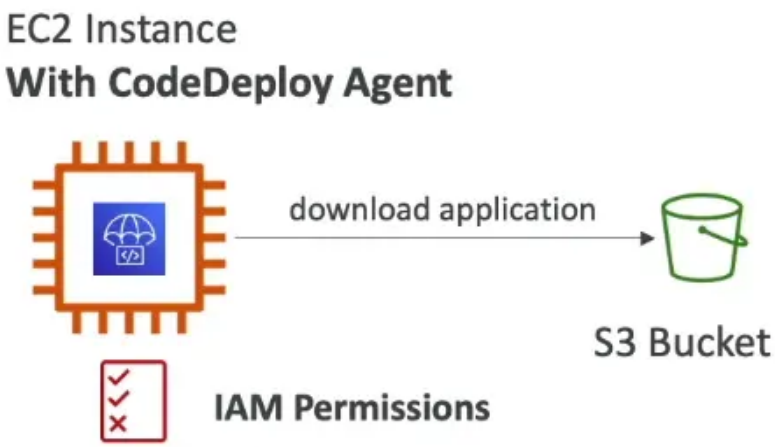
half at a time



Blue–Green Deployment



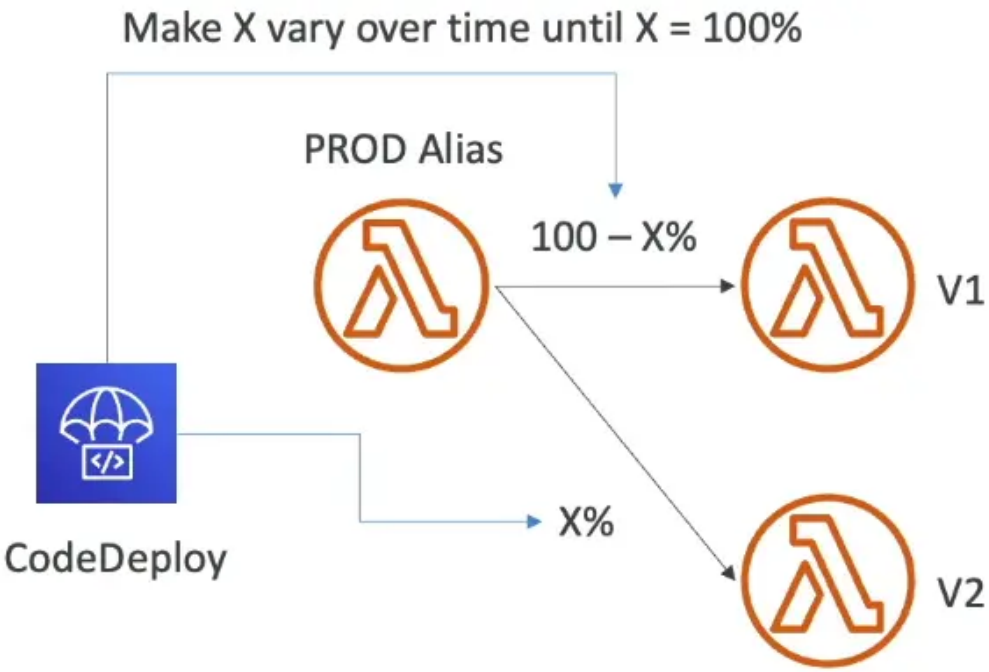
CodeDeploy Agent




```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "s3:Get*",
                "s3:List*"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

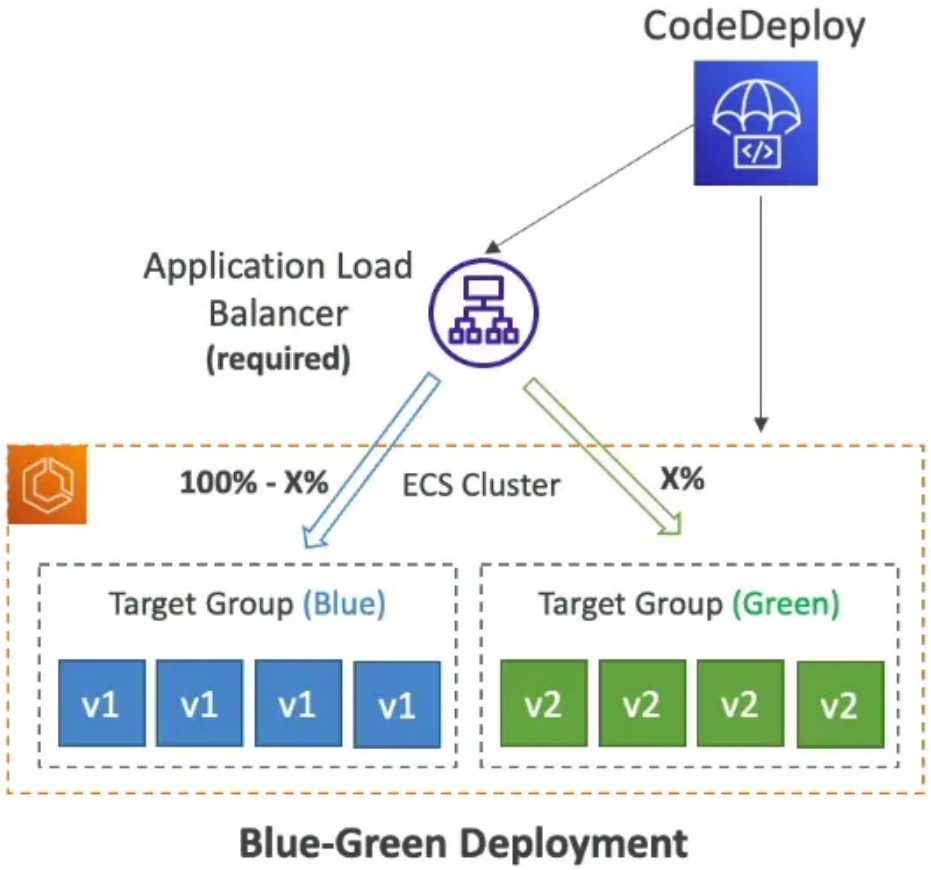
- the CodeDeploy Agent must be running on the EC2 instances as a prerequisites [先决条件]
- it can be installed and updated automatically if you're using system manger
- the EC2 instances must have sufficient [充足的] permissions to access Amazon S3 to get deployment bundles [捆绑包]

Lambda Platform



- CodeDeploy can help you automate traffic shift for Lambda aliases
- Feature is integrated [集成] within the SAM framework
- Linear: grow traffic every N minutes until 100%
  - lambdaLinear 10PercentEvery3Minutes
  - lambdaLinear 10PercentEvery10Minutes
- canary: try X percent then 100%
  - lambdaCanary10PercentEvery5Minutes
  - lambdaCanary10PercentEvery30Minutes
- AllAtOnce: immediate

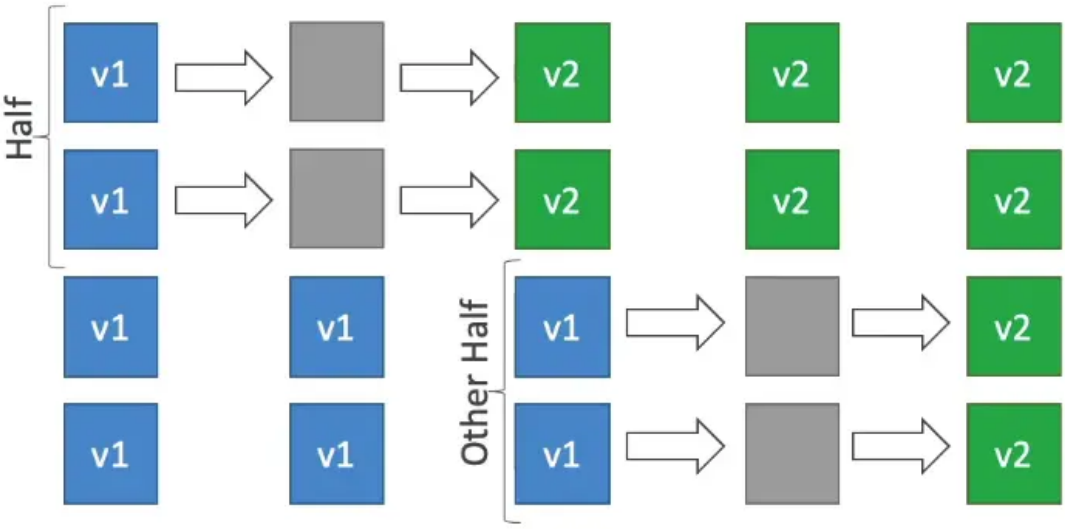
ECS Platform



Blue-Green Deployment

- CodeDeploy can help you automate the deployment of a new ECS Task Definition
- Only Blue/Green Deployments
- Linear: grow traffic every N minutes until 100%
  - lambdaLinear 10PercentEvery3Minutes
  - lambdaLinear 10PercentEvery10Minutes
- canary: try X percent then 100%
  - lambdaCanary10PercentEvery5Minutes
  - lambdaCanary10PercentEvery30Minutes
- AllAtOnce: immediate

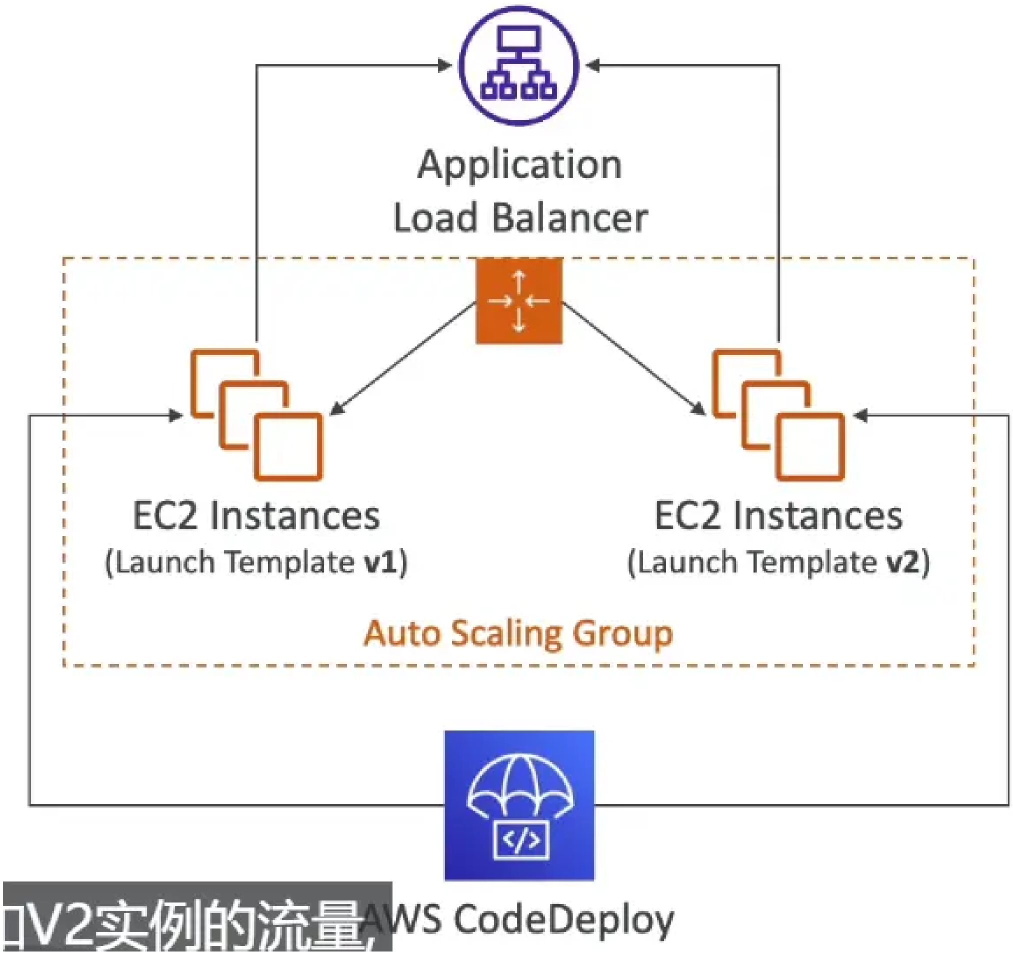
Deployment to EC2

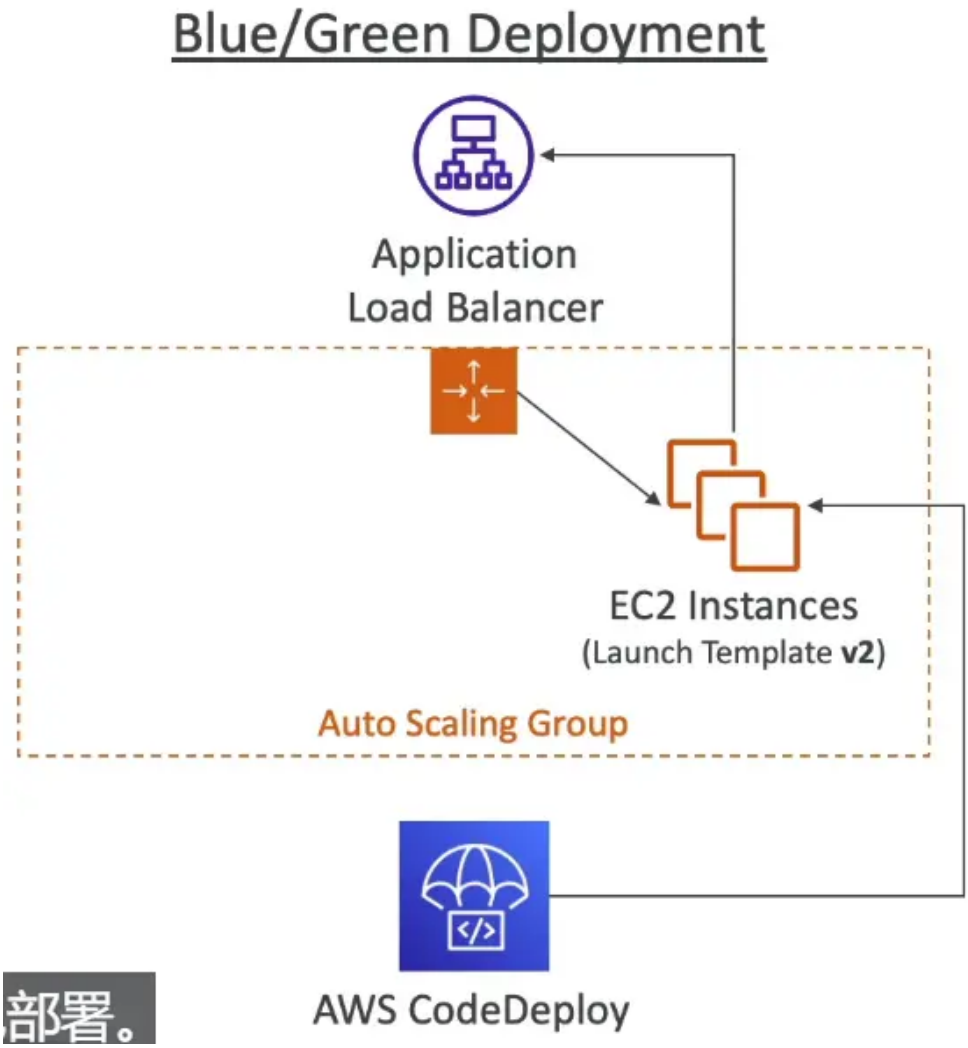


- define how to deploy the application using appspec.yml + Deployment Strategy
- will do In-place update to your fleet of EC2 instances
- can use hooks to verify the deployment after each deployment phase [阶段]

Deploy to an ASG

Blue/Green Deployment





部署。

- in-place Deployment
  - updates existing EC2 instances
  - newly created EC2 instances by an ASG will also get automated deployments
- Blue / Green Deployment
  - a new Auto-Scaling Group is created (settings are copied)
  - choose how long to keep the old EC2 instances (old ASG)
  - must be using an ELB

Redeploy & Rollbacks

- rollback = redeploy a previously deployed revision of your applicaition
- deployments can be rolled back
  - Automatically – rollback when a deployment fails or rollback when ClouWatch Alarm thresholds are met
  - Manually
- disable rollbacks -- do not perform rollbacks for this deployment

- if a roll back happens , CodeDeploy redeploys the last known good revision as a new deployment (not a restored version)

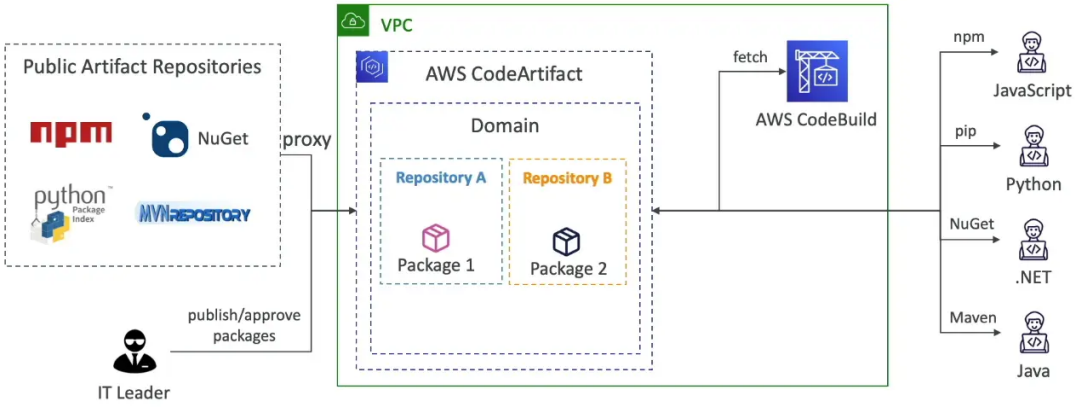
AWS CodeStar [replacement "CodeCatalyst"]

- an integrate solution that groups : github, codecommit, codebuild,codedeploy,cloudformation,codepipeline, cloudwatch
- quickly creat "CICD-ready" projects for EC2, lambda, Elastic Beanstalk

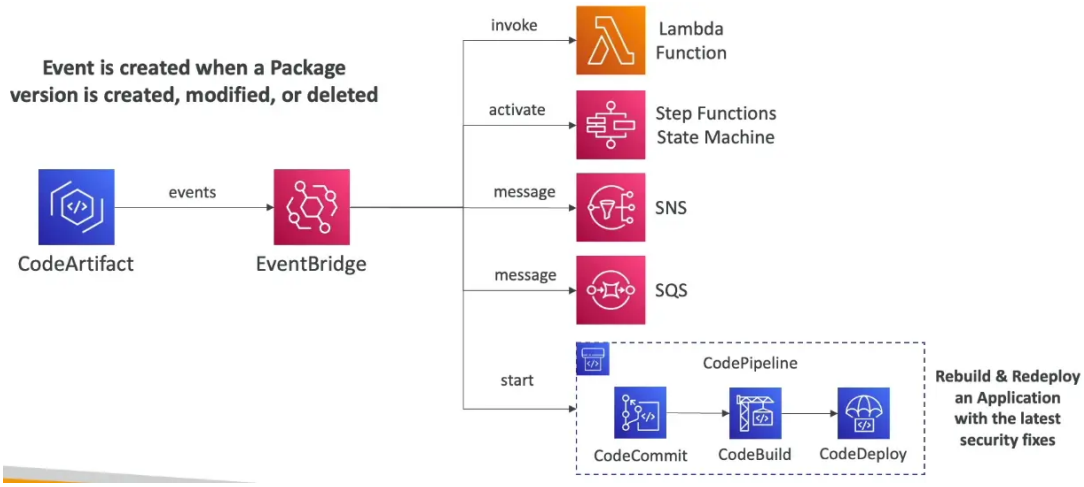
- supported language : C#,Go,HTML5,Java,Node.js,PHP,Python,Ruby
- issue tracking integration with JIRA / Github Issues
- ability to integrate with Cloud9 to obtain a web IDE (not all regions)
- one dashboard to view all your components
- free services, pay only for the underlying usage of other services
- limited customization

AWS CodeArtifact

- software packages depend on each other to be built (also called code dependencies) ,and new ones are created
- storing and retrieving these dependencies is called **artifact management**
- traditionally you need to setup your own artifact management system
- **CodeArtifact** is a secure,scalable, and cost-effective **artifact management** for software development
- works with common dependency management tools such as Maven,Gradle,npm, yarn,twine,pip,and NuGet
- **Developers and CodeBuild can then retrieve dependencies straight form CodeArtifact**



EventBridge Integration



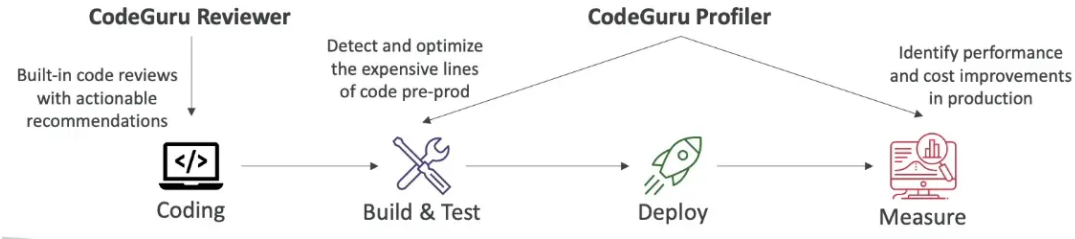
Resource Policy

- can be used to authorize another account to access CodeArtifact
- a given principal can either read all the packages in a repository or none of them


```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "codeartifact:DescribePackageVersion",
                "codeartifact:DescribeRepository",
                "codeartifact:GetPackageVersionReadme",
                "codeartifact:GetRepositoryEndpoint",
                "codeartifact:ListPackages",
                "codeartifact:ListPackageVersions",
                "codeartifact:ListPackageVersionAssets",
                "codeartifact:ListPackageVersionDependencies",
                "codeartifact:ReadFromRepository"
            ],
            "Principal": {
                "AWS": [
                    "arn:aws:iam::123456789012:root",
                    "arn:aws:iam::222333344555:user/bob"
                ]
            },
            "Resource": "*"
        }
    ]
}
```

Repository Resource Policy

CodeGuru



- an ML–powered service for **automated code reviews** and **application performance recommendations**
- provides two functionalities
  - **CodeGuru Reviewer**: automated code reviews for static code analysis (development)
  - **CodeGuru Profiler**: visibility/ recommendations about application performance during runtime (production)

CodeGuru Reviewer

- identity critical issues,security vulnerabilities [漏洞], and hard-to-find bugs
- example: common coding best parctices, resource leaks, security detection, input validation
- use machine learning and automated reasoning
- hard-learned lessons across millions of code reviews on 1000s of open-source and Amazon repositories
- supports Java and Python
- integrates with GitHub, Bitbucket, and AWS CodeCommit

CodeGuru Profiler

- helps understand the runtime behavior of your application
- example: identify if your application is consuming [消耗] excessive [过多的] CPU capacity on a logging routine
- features
  - identify and remove code inefficiencies
  - improve application performance (eg,reduce CPU utilization)
  - decrease compute costs
  - provides heap summary (identify which objects using up memory)
  - anomaly detection [异常检测]
- support applications running on AWS or on premise [在本地]
- minimal overhead on application [应用程序开销最小]

Agent Configuration

- MaxStackDepth – the maximum depth of the stacks in the code that is repressented in the profile
  - example: if CodeGuru profiler finds a method A, which call methodB,which calls method C, which calls method D, then the depth is 4
  - if the MaxStackDepth is set to 2, then the profiler evaluates [评估] A and B
- MemoryUsageLimitPercent – the memory percentage used by the profiler
- MinimumTimeForReportingInMilliseconds – the minimum time between sending reports (millseconds)
- ReportingIntervalInMilliseconds – the reporting interval used to report profiles (milliseconds)
- SamplingIntervalInMillseconds – the sampling interval that is used to profile samples (milliseconds)
  - reduce to have a higher samping rate

Cloud 9

- Cloud-based Integrated Development Environment (IDE)
- code editor,debugger,terminal in a brower
- work on your projects from anywhere with an internet connection
- prepackaged with essential tools for popular programming languages (javascript,python,php...)
- share your development environment with your team (pair programming)

- fully integrated with AWS SAM & Lambda to easily build serverless application