How Code star differ from codepipeline/gitlab.

AWS CodePipeline vs AWS CodeStar: What are the differences?

Developers describe **AWS CodePipeline** as "*Continuous delivery service for fast and reliable application updates*". CodePipeline builds, tests, and deploys your code every time there is a code change, based on the release process models you define. On the other hand, **AWS CodeStar** i s detailed as "*Quickly Develop, Build, and Deploy Applications on AWS*". Start new software projects on AWS in minutes using templates for web applications, web services and more.

AWS CodePipeline and AWS CodeStar are primarily classified as **"Continuous Deployment"** and **"Platform as a Service"** tools respectively.

Some of the features offered by AWS CodePipeline are:

- Workflow Modeling
- AWS Integrations
- Pre-Built Plugins

On the other hand, AWS CodeStar provides the following key features:

- Start developing on AWS in minutes
- Manage software delivery in one place
- Work across your team securely.

Comparisons:-

AWSCodestar and Codepipeline

Feature	AWS codestar	AWS codepipeline	AWS codecommit	AWS codebuild	AWScodedeploy
Workflow Modeling	no	yes	N/A	N/A	N/A
AWS services Integrations	no but integrate with codepipeline/build/commit/deploy	yes all services integrates	N/A	yes	yes
Pre-Built Plugins	yes limited upto github&jira	yes	N/A	yes	yes
Start developing on AWS in minutes	yes	no	N/A	N/A	N/A
Manage software delivery in one place	yes	no	N/A	N/A	N/A
Work across your team securely	yes	no customize with IAM roles	no customize with IAM roles	no customize with IAM roles	no customize with IAM roles
integrate with IDE	yes cloud9, eclipse, CLI,visual studio.	no	yes customize with any IDE	N/A	N/A
Jira tracking	yes	no	N/A	N/A	N/A
integrate with jenkins	no	yes customize in jenkins jobs with pipeline	yes pass repo in build jobs	yes integrate with jenkins jobs	yes
integrate with github	yes	yes	N/A	yes	yes
Built for using containers and Docker	yes	yes	N/A	yes	N/A
Application performance monitoring	yes	yes	N/A	N/A	yes
Application performance alerts	yes	yes	N/A	N/A	yes

A comprehensive API	yes	yes	yes	yes	yes
Scheduled triggering of pipelines	yes	yes	N/A	yes	yes
Support for Amazon ECS and AWS Fargate	yes	yes	N/A	yes	yes
support for lamda	yes	yes	N/A repo store only	yes	yes
support for ec2.	yes	yes	N/A	yes	yes
support for beanstack	yes	yes	N/A	yes	yes
defined templates	yes	no	N/A repo format	N/A	N/A
supported for all programming langages	no limit upto Node js, expressjs, java spring, python Django, asp.net, html,go,ruby,php	yes we can custom the build tools in dockerfile in ECR.	yes	yes	yes
source provider	Code commit, github.	S3, Code commit, Ecr, github.	N/A	Bitbucket,github S3, codecommit	S3, github,
backup with S3	yes	yes	yes	yes	yes
logs	yes cloudwatch	yes cloudwatch	yes cloudwatch	yes codewatch	yes cloudwatch
history	yes	yes	yes	yes	yes
deploy targets	Ec2, lamda, Beanstack.	Cloud formation,Code deploy,Elastic beanstack Opsworks,Service catalog Alexa skills kit,ECS S3.	N/A	S3 (deafult) can customize.	ec2 instaces econtainer service, lamda
integrate with codestar	N/A	yes	yes	yes	yes
integrate with codepipeline	yes	N/A	yes	yes	yes

Comparison

Code commit	Code build	Code deploy	Code pipeline	Code star

Step 1: Prerequisites

You must use a Git client that supports Git version 1.7.9 or later to connect to an AWS CodeCommit repository

Step 2: Git credentials

Create Git credentials for your IAM user, if you do not already have them. Download the credentials and save them in a secure location.

Step 3: Clone the repository

Clone your repository to your local computer and start working on code.

1. https and ssh are both available.
2. Commit, author, email, commit id provides.
3. Rest of same as git.
4. Autenticate with AWS IAM by creating ssh key and provide the repo local.

Source available:

Bitbucket,github

S3, codecommit

Give the authentication process for this Bitbucket url and user name.

Build environment:

1. Build will runs in container and should map the docker images.
2. Custome one or default once select and authentication process, select the repo of image-ECR.
3. provide the buildspe.yml file path.Artifacts:

Artifacts:

1. Create a zip file in s3.
2. Logs cloud watch

Application:

Compute platform:

-ec2 instaces

-econtainer service

- lamda

select the deployment.

target deployments type

Ec2:-

Amazon EC2 Auto Scaling groups, Amazon EC2 instances, and on-premises instances to add to this deployment

---

Econtainer service:

Give cluster and service

Lambda functions:

provide the function details

Source :

S3, github,

appspe.

specify the source of code to deploy

Source providers:

S3,

Code commit,

Ecr, github.

Build:

Jenkins build---pipeline&publisher

Code build

select any one.

Deploy:

Cloud formation

Code deploy

Elastic beanstack

Opsworks

Service catalog

Alexa skills kit

ECS

S3.

Release changes.

select the deployment service.

Project templates:

Node js, expressjs, java spring, python Django, asp.net, html,go ,ruby,php

Aws platforms:

Ec2, lamda,

Beanstack.

Source provider:

Code commit, github.

Ide: configure for aws.

pipeline creates automatically.

Jira extension

Team members restrict

Build spec:

```
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

    finally:

      - echo This always runs even if the
update or install command fails

  pre_build:

    commands:

      - echo Entered the pre_build phase...

      - docker login –u User –p
$LOGIN_PASSWORD

    finally:

      - echo This always runs even if the login
command fails

  build:

    commands:

      - echo Entered the build phase...

      - echo Build started on `date`
```

```
    - mvn install
  finally:
    - echo This always runs even if the install command fails
 post_build:
  commands:
    - echo Entered the post_build phase...
    - echo Build completed on `date`
artifacts:
 files:
  - target/messageUtil-1.0.jar
 discard-paths: yes
 secondary-artifacts:
  artifact1:
   files:
    - target/messageUtil-1.0.jar
   discard-paths: yes
  artifact2:
   files:
    - target/messageUtil-1.0.jar
   discard-paths: yes
cache:
 paths:
  - '/root/.m2/**/*'
```

CODE PIPELINE
CODE COMMIT + CODE BUILD + CODE DEPLOY(custom at any stage)

CODE STAR(defined templates)
CODE PIPELINE(no customized templates avail)
CODE COMMIT + CODE BUILD + CODE DEPLOY

Codepipeline use cases:

- Use CodePipeline with Amazon S3, AWS CodeCommit, and AWS CodeDeploy
- Use CodePipeline with Third-party Action Providers (GitHub and Jenkins)
- Use CodePipeline with AWS CodeStar to Build a Pipeline in a Code Project
- Use CodePipeline to Compile, Build, and Test Code with CodeBuild
- Use CodePipeline with Amazon ECS for Continuous Delivery of Container-Based Applications to the Cloud
- Use CodePipeline with Elastic Beanstalk for Continuous Delivery of Web Applications to the Cloud
- Use CodePipeline with AWS Lambda for Continuous Delivery of Lambda-Based and Serverless Applications
- Use CodePipeline with AWS CloudFormation Templates for Continuous Delivery to the Cloud.

CodeStar use cases:

- Use Codestar project with application category, programming langauage, aws service(ec2, lamda,beanstack)
- Use CodePipeline with AWS CodeStar to Build a Pipeline in a Code Project.

AWS codestar and gitlab.

feature	AWS codestar	gitlab
Free CI/CD with shared or personal Runners	no	yes

Application performance monitoring	yes	yes
Application performance alerts	yes	yes
Preview your changes with Review Apps	no	yes
A comprehensive API	yes	yes
Built for using containers and Docker	yes	yes
Comprehensive pipeline graphs	no	yes
Scheduled triggering of pipelines	yes	yes
Multi-project pipeline graphs	no	yes
Run CI/CD jobs on Windows	no	yes
Run CI/CD jobs on macOS	no	yes
Easy integration of existing Kubernetes clusters	no	yes
Minimal CI/CD configuration	no	yes
View Kubernetes pod logs	yes	yes
Windows Container Executor	no	yes
Visual Reviews	no	yes

thats all done !!!