Comparison between CodeStar/codepipeline/gitlab/jenkins.

In this document, im going to explain Comparison between CodeStar/codepipeline/gitlab/jenkins.

- About tools
- Comparison between Code Star and code-pipeline.
- Comparison between code-star, Jenkins and git-lab.
- Conclusion

So lets start quickly.

Process of individual tool:

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" phases: install: commands: - apt-get install -y maven pre_build: commands: - 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 Build completed on `date` artifacts: files: - target/messageUtil-1.0.jar discard-paths: yes			

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 programming language template and deploy in aws service ec2.
- Use Codestar project with programming language template and deploy in aws service lamda.
- Use Codestar project with programming language template and deploy in aws service elastic beanstack.

Comparison between Code Star and code-pipeline.

Feature	AWS codestar	AWS codepipeline	AWS codecommit	AWS codebuild	AWScodedeploy
A comprehensive API	yes	yes	yes	yes	yes
Application performance alerts	yes	yes	N/A	N/A	yes
Application performance monitoring	yes	yes	N/A	N/A	yes

AWS services Integrations	no but integrate with codepipeline/build/commit/deploy	yes all services integrates	N/A	yes	yes
backup with S3	yes	yes	yes	yes	yes
Built for using containers and Docker	yes	yes	N/A	yes	N/A
defined templates	yes	no	N/A repo format	N/A	N/A
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
history	yes	yes	yes	yes	yes
integrate with codepipeline	yes	N/A	yes	yes	yes
integrate with codestar	N/A	yes	yes	yes	yes
integrate with github	yes	yes	N/A	yes	yes
integrate with IDE	yes cloud9, eclipse, CLI,visual studio.	no	yes customize with any IDE	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
Jira tracking	yes	no	N/A	N/A	N/A
logs	yes cloudwatch	yes cloudwatch	yes cloudwatch	yes codewatch	yes cloudwatch
Manage software delivery in one place	yes	no	N/A	N/A	N/A
Pre-Built Plugins	yes limited upto github&jira	yes	N/A	yes	yes
Scheduled triggering of pipelines	yes	yes	N/A	yes	yes
source provider	Code commit, github.	S3, Code commit, Ecr, github.	N/A	Bitbucket,github S3, codecommit	S3, github,
Start developing on AWS in minutes	yes	no	N/A	N/A	N/A
Support for Amazon ECS and AWS Fargate	yes	yes	N/A	yes	yes
support for beanstack	yes	yes	N/A	yes	yes
support for ec2.	yes	yes	N/A	yes	yes
support for lamda	yes	yes	N/A repo store only	yes	yes
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

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
Workflow Modeling	no	yes	N/A	N/A	N/A

Comparison between Code Star, jenkins,Gitlab.

feature	AWS codestar	gitlab	Jenkins
Free CI/CD with shared or personal Runners	no	yes	yes open source
Application performance monitoring	yes	yes	yes can customize with plugin
Application performance alerts	yes	yes	yes can customize with plugin
Preview your changes with Review Apps	no	yes	yes can customize with plugin
A comprehensive API	yes	yes	yes can customize with plugin groovy
Built for using containers and Docker	yes	yes	yes need to add docker as slave
Comprehensive pipeline graphs	no	yes	yes
Scheduled triggering of pipelines	yes	yes	yes
Multi-project pipeline graphs	no	yes	yes
Run CI/CD jobs on Windows	no	yes	yes
Run CI/CD jobs on macOS	no	yes	yes
Easy integration of existing Kubernetes clusters	no	yes	yes
Minimal CI/CD configuration	no	yes	yes
View Kubernetes pod logs	yes	yes	yes
Windows Container Executor	no	yes	yes
Visual Reviews	no	yes	yes
autentication and authorization	yes	yes	yes plugin needed
devops score(Auto devops)	yes	yes	yes with jenkins x
workflow policies	yes	yes	no
project managment	yes	yes	yes plugin needed
source code managment	yes	yes	yes plugin
wiki	yes	yes	yes
web IDE	yes	yes	yes
code quality	yes	yes	yes
CI	yes	yes	yes
testing	yes	yes	yes
package registry	no	yes	yes
container registry	yes	yes	yes
continuous delivery	yes	yes	yes
release orchestration	yes	yes	yes
auto kubernetes configurations	no	no	yes with jenkins x

server-less	yes	yes	yes
metrics, logging, tracking.	yes	yes	yes
multi-branching pipelines	no	yes	yes
use code as a build	yes buildspec.yml	yes	yes jenkinsfile
developed languages	no limit upto Node js, expressjs, java spring, python Django, asp.net, html,go,ruby,php	yes to all langauages	yes to all
spinning of agent on demand	no	no	yes
cost	yes	yes	no

conclusion:

Even though CodePipeline and Jenkins operate as solo CI/CD tools, you can actually use them together in a multi-stage deployment pipeline. For example, you can create a four-stage pipeline in AWS CodePipeline that utilizes Jenkins as a build server. Of course, you will still rely on a source repository, such as GitHub or GitLab, and need a delivery mechanism -- AWS CodeDeploy, most likely -- for the built code to push to a server.

thank you.