**01. EKS DevOps - Introduction**

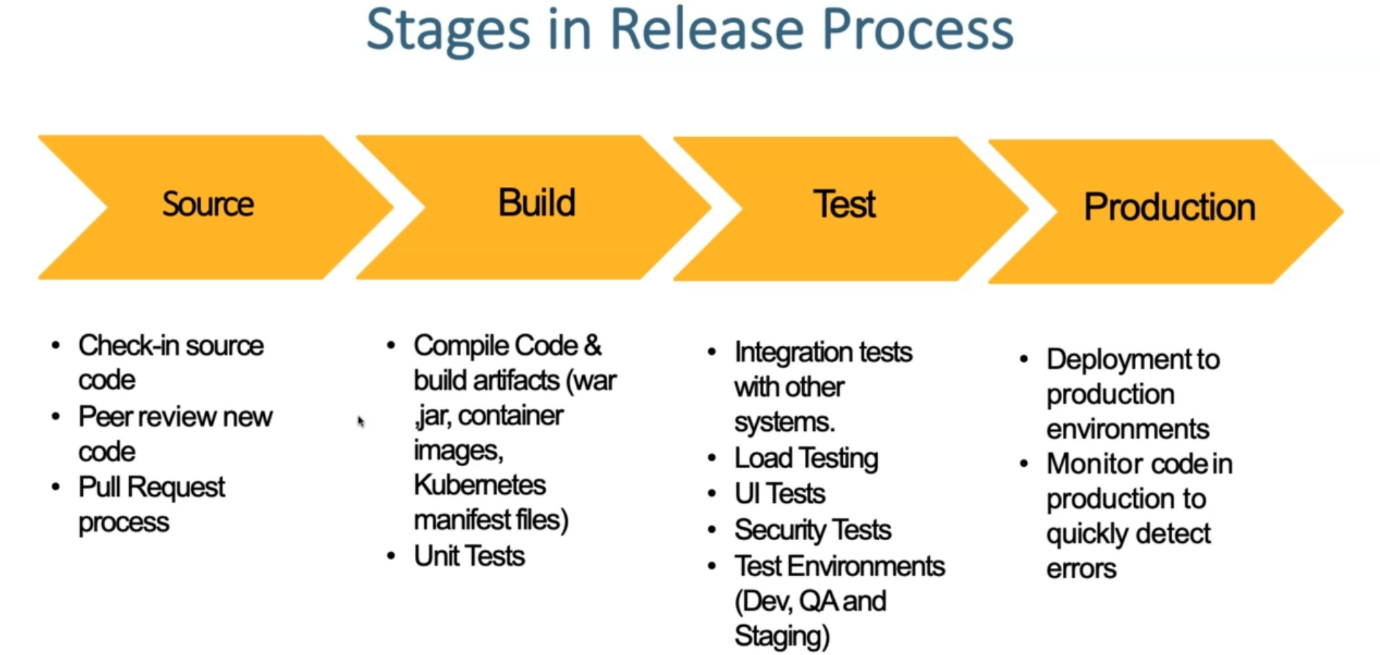
--- in this lecture, we are going to understand about core devops concepts like continuous integration and continuous delivery or deployment on a high level.

--- To achieve this on aws in EKS. we can use few tools like code commit, code build and then code pipelining in AWS.

--- using these things, you're are going to implement continuous integration and then continuous delivery.

--- let's see the stages in release process now.

**Stages in release process**



**Which stage belongs to which type**

--- **source** and **build stage** just come under continuous integration

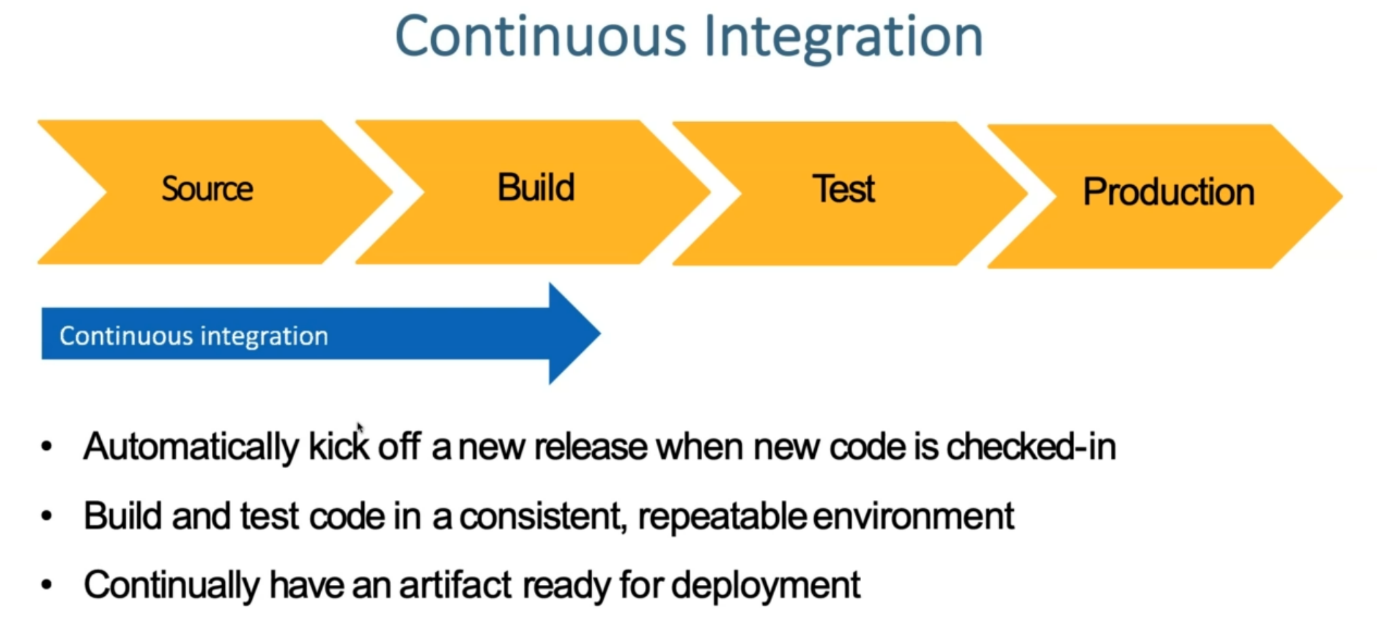
--- test and production come’s under continuous delivery or continuous deployment.

--- one more thing here is infrastructure as a code.

--- If you see continuous integration, continuous delivery, continuous deployment. how these are related to our regular release processes.

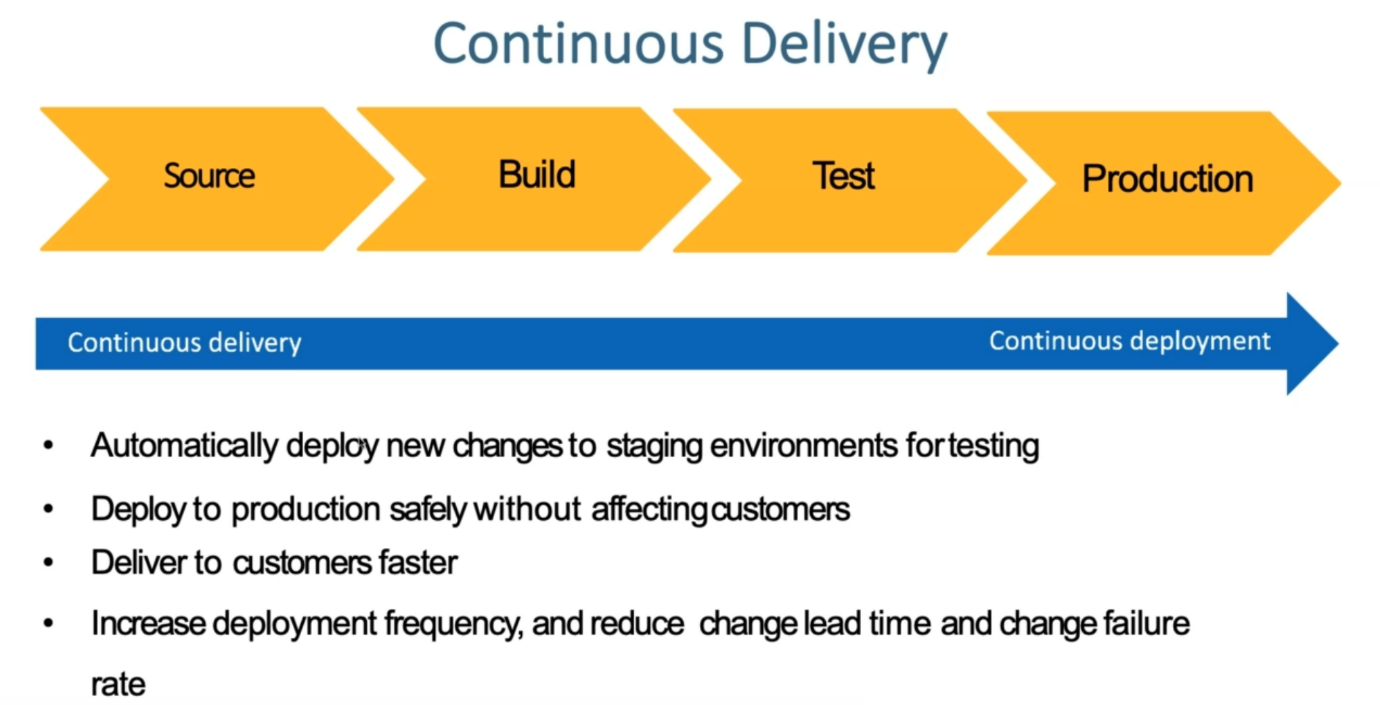
--- **Infrastructural score** - whatever we have in source build and test, productions. all these stages, if you want to automate using aws cloud formation so you can automate that.

**continuous integration**



--- **continuous integration** - next thing is continuous integration. what does this mean on a high level? automatically kick off a new release. when new code is checked in and build and test that code in consist able, repeatable environment and continually have an artifact ready for deployment.

**Continuous delivery**

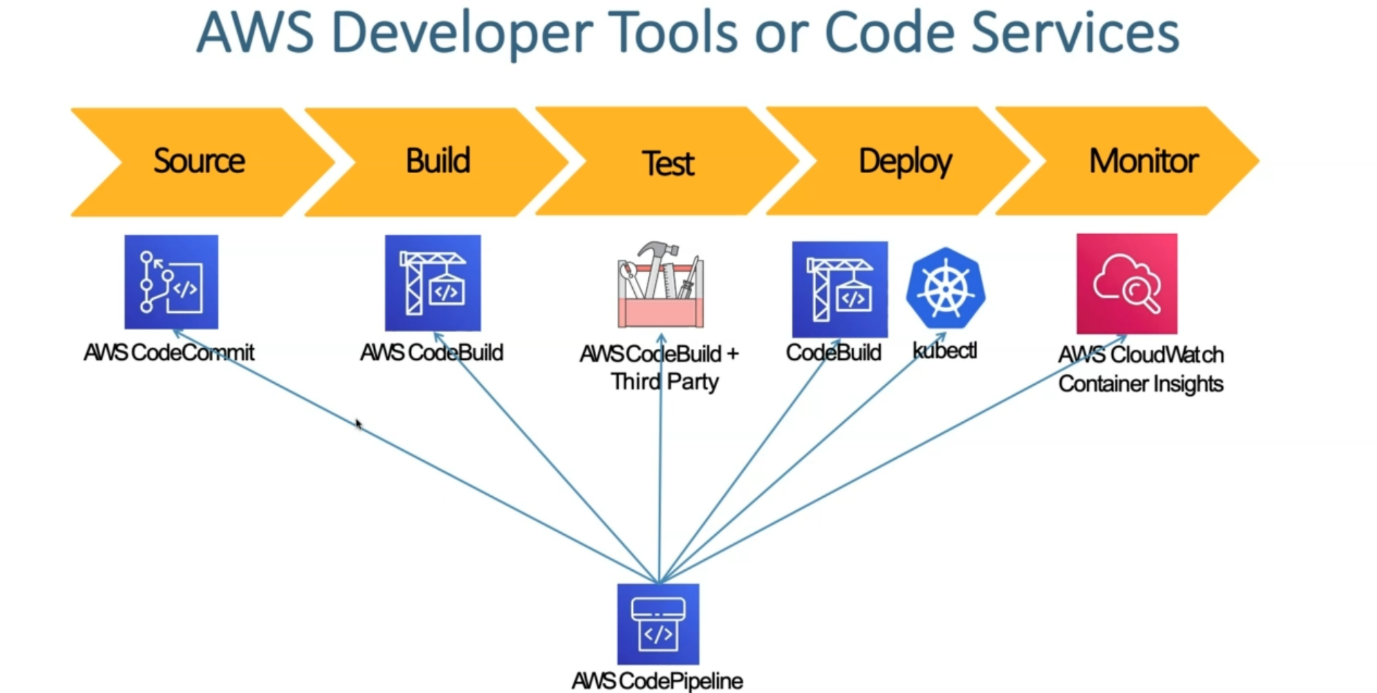


--- in continuous delivery or deployment, automatically deploy new changes to staging environments for testing.

--- So deployed to production safely without affecting the customers and deliver to customers faster and increase deployment frequency and reduce change lead time and then change failure rate.

--- these are the advantages of continuous delivery process if we implement it.

**AWS developer tools or code services**



--- how these developer tools or code services fit into this continuous integration and continuous delivery pipeline.

--- aws has something called code Pipeline, which will be in integration with the different aws tools. example for Source, it is going to use the aws codecommit, which is something like a github repository. here we call it as codecommit repository.

--- **aws codebuild** - for building, we usually see JENKINS, we might have seen in many places but in AWS, we can use codebuild and which is a tool which provides us to build our artifacts and then push you to s3 buckets in aws.

--- **test** - either you can use AWS codebuild plus third-party tools to do our testing related things.

--- **deploy** - in our case. For kubernetes, we are using codebuild in combination with kubectl to the deployment in our use case. I brought them under the deploy stage here.

--- **monitoring** - for monitoring perspective, you will have something called cloud watch container insights. using that, we are going to do the end-to-end monitor off our container applications deployed on kubernetes cluster.

**How these tools fit in which category**

