

Introduction of Jenkins X

Overview

What does a classic CI/CD platform look like and what are its shortcomings?

Introduction to Jenkins X and how it addresses these shortcomings

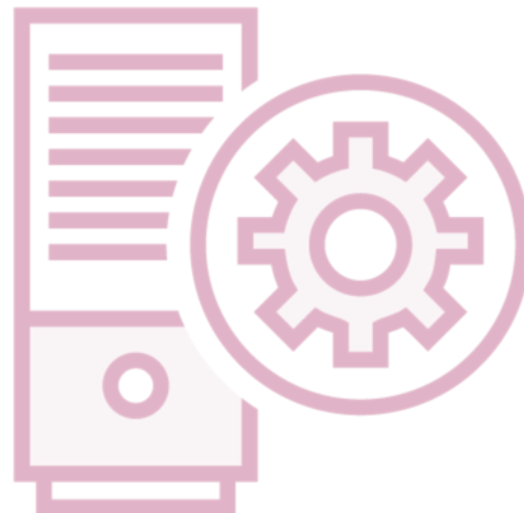
Jenkins X architecture

Differences between classic Jenkins and Jenkins X

A Typical CI/CD System



Local code

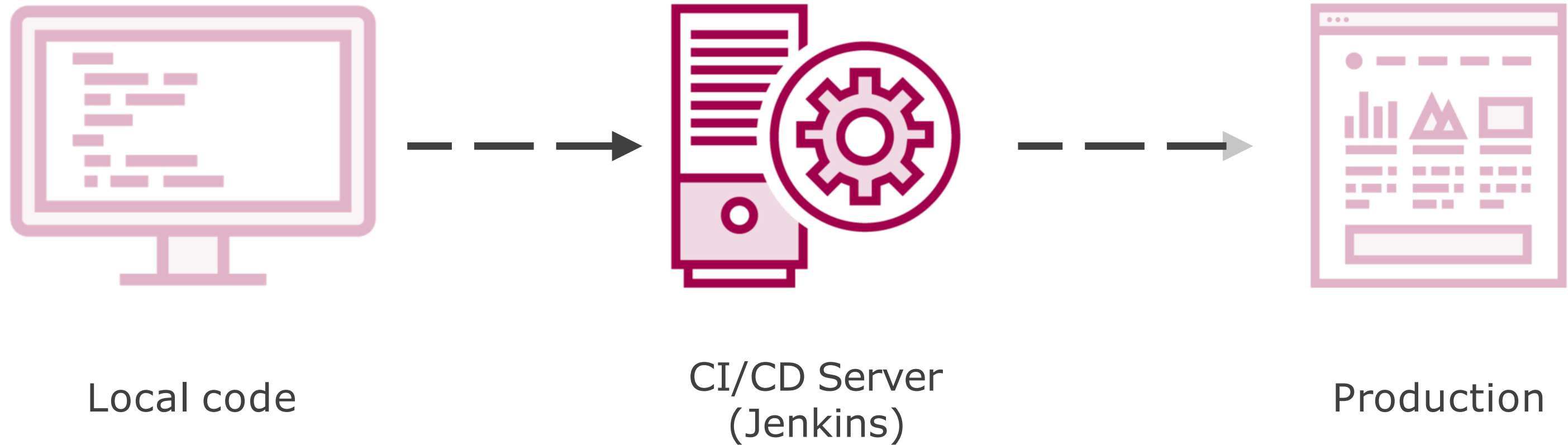


CI/CD Server
(Jenkins)

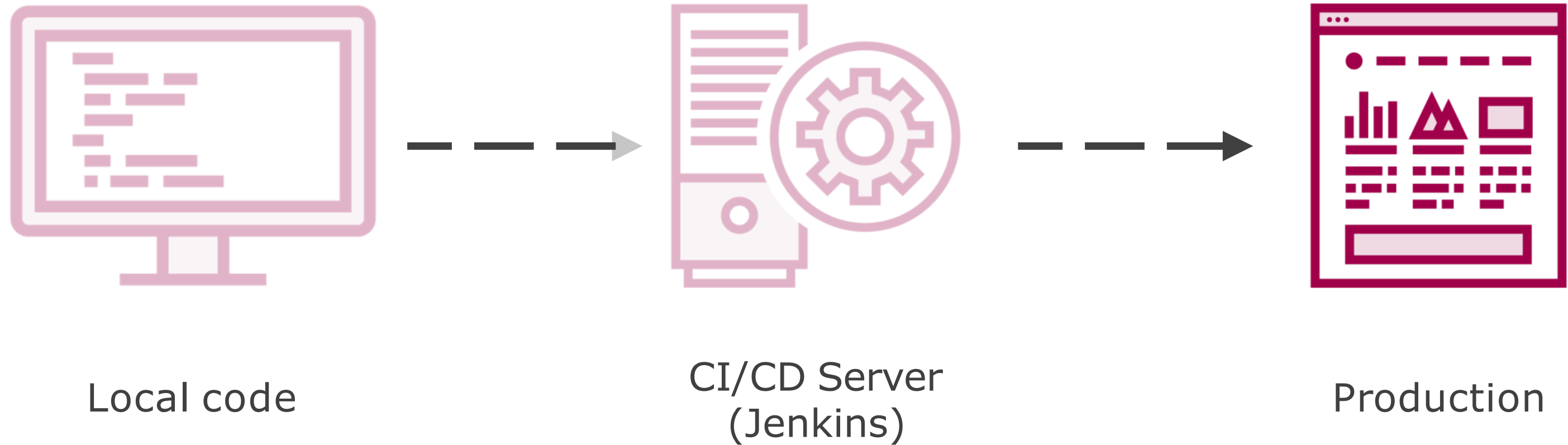


Production

A Typical CI/CD System



A Typical CI/CD System



A core problem with traditional CI/CD platforms like classic Jenkins is that they are un-opinionated and require heavy customization

Too Much Extra Scaffolding



Installation and configuration

Version control platform integration

Environment creation

Project creation

Custom pipelines

Packaging strategy

Registry provisioning

Deployment strategies

Running in the Cloud



Not cloud-native

Running in the Cloud



Not cloud-native



Static and resource
hungry

Running in the Cloud



Not cloud-native



Static and resource hungry



Provisioning new agents

Custom Deployment Strategy



Deployments may not be versioned in Git (GitOps)

Custom Deployment Strategy



Deployments may not be versioned in Git (GitOps)



They may not be observable and give free reign to the operations cowboy

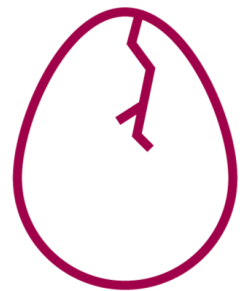
Custom Deployment Strategy



Deployments may not be versioned in Git (GitOps)



They may not be observable and give free reign to the operations cowboy



They may not be declarative and idempotent

Pipeline Repetition

```
pipeline {
  agent { docker { image 'maven:3.3.3' } }
  stages {
    stage('build') {
      steps {
        sh './mvn install'
      }
    }
    stage('dockerize') {
      steps {
        sh 'docker build'
      }
    }
  }
}
```

Pipeline Repetition

```
pipeline {  
  agent { docker { image 'maven:3.3.3' } }  
  
  stages {  
    stage('build') {  
      steps {  
        sh './mvn install'  
      }  
    }  
    stage('dockerize') {  
      steps {  
        sh 'docker build'  
      }  
    }  
  }  
}
```

← — — — — Same for all Maven builds

Pipeline Repetition

```
pipeline {  
  agent { docker { image 'maven:3.3.3' } }  
  
  stages {  
    stage('build') {  
      steps {  
        sh './mvn install'  
      }  
    }  
  
    stage('dockerize') {  
      steps {  
        sh 'docker build'  
      }  
    }  
  }  
}
```



Same for all Docker projects

Jenkins X is an opinionated,
cloud-native CI/CD platform
built on top of Kubernetes

Jenkins X is an open-source automation platform that helps teams automate the development, testing, and deployment of cloud-native applications. It is based on Jenkins, an open-source automation server, and Tekton, a cloud-native pipeline orchestration system.

Jenkins X is a powerful tool that can help teams automate the development, testing, and deployment of cloud-native applications. It is a good choice for teams that are looking to adopt a CI/CD pipeline and/or move to the cloud.

Jenkins X Feature

- **Continuous integration and continuous delivery (CI/CD):** Jenkins X can automate the entire CI/CD pipeline, from building and testing code to deploying it to production.
- **GitOps:** Jenkins X uses GitOps to manage its configuration. This means that all of the configuration for Jenkins X is stored in a Git repository, which makes it easy to track changes and collaborate with others.
- **Pre-built pipelines:** Jenkins X comes with a number of pre-built pipelines that can be used to automate common tasks, such as building and deploying Java applications.
- **Out-of-the-box integrations:** Jenkins X integrates with a number of popular cloud-native tools, such as Kubernetes, Docker, and Helm.

Jenkins X Benefits

Here are some of the benefits of using Jenkins X:

- **Reduced time to market:** Jenkins X can help teams automate the CI/CD pipeline, which can significantly reduce the time it takes to get new features to market.
- **Improved quality:** Jenkins X can help teams improve the quality of their code by automating the testing process.
- **Increased collaboration:** Jenkins X makes it easy for teams to collaborate on code and deployments.
- **Lower costs:** Jenkins X can help teams lower the costs of development and deployment by automating the process.

The Opinions of Jenkins X

Our application runs
on Kubernetes

Kubernetes



Don't worry if you're a beginner!

Container orchestration tool built for the cloud

- Unified declarative deployment model
- Service-discovery and load balancing
- Horizontal scaling
- Self-healing

Check out a Kubernetes Pluralsight course

The Opinions of Jenkins X

Our application runs
on Kubernetes

We use Helm and
Docker

The Opinions of Jenkins X

Our application runs
on Kubernetes

We use Helm and
Docker

We use GitOps

The Opinions of Jenkins X

Our application runs
on Kubernetes

We use Helm and
Docker

We use GitOps

Each language has a
re-useable default
structure (buildpack)

The Opinions of Jenkins X

Our application runs
on Kubernetes

We use Helm and
Docker

We use GitOps

Each language has a
re-useable default
structure (buildpack)

Pipelines are
extendable and re-
usable

The Opinions of Jenkins X

Our application runs
on Kubernetes

We use Helm and
Docker

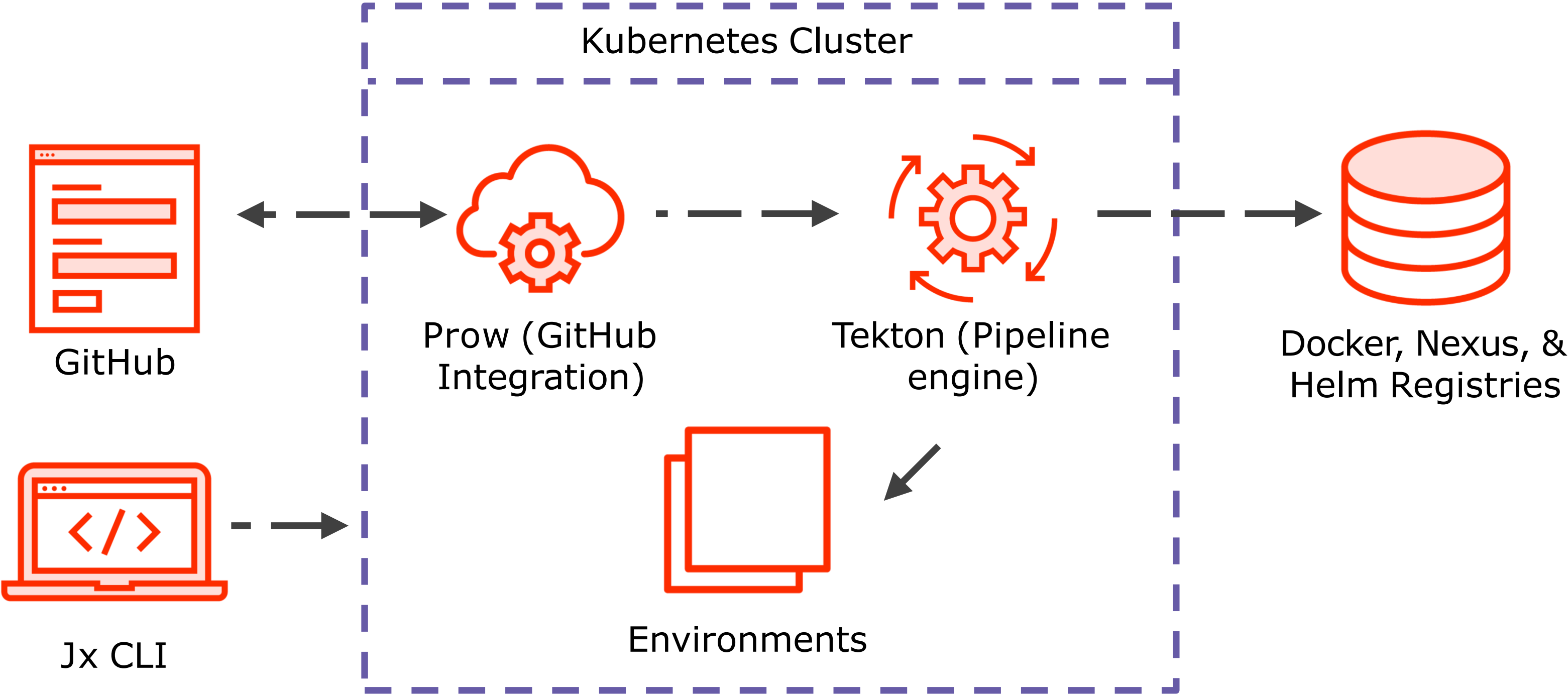
We use GitOps

Each language has a
re-useable default
structure (buildpack)

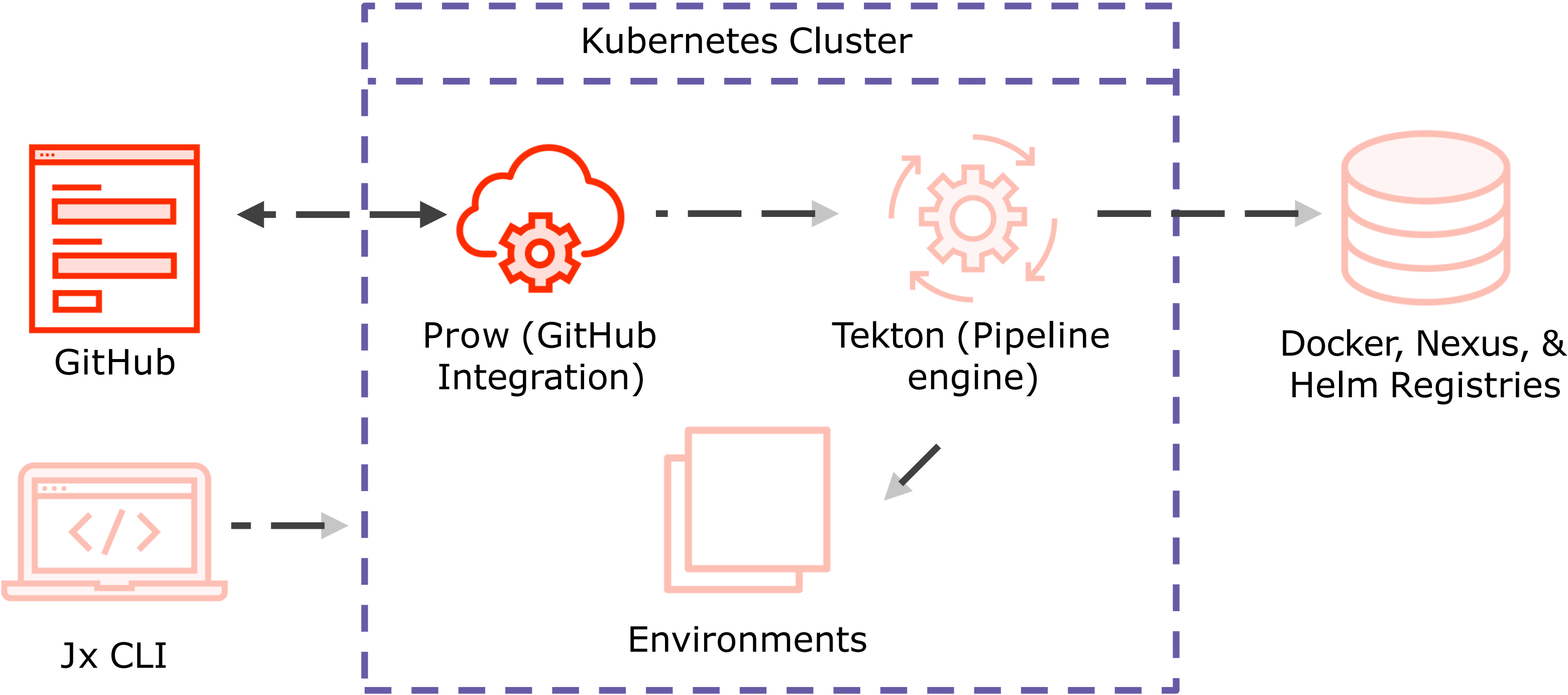
Pipelines are
extendable and re-
usable

Its components are
elastic

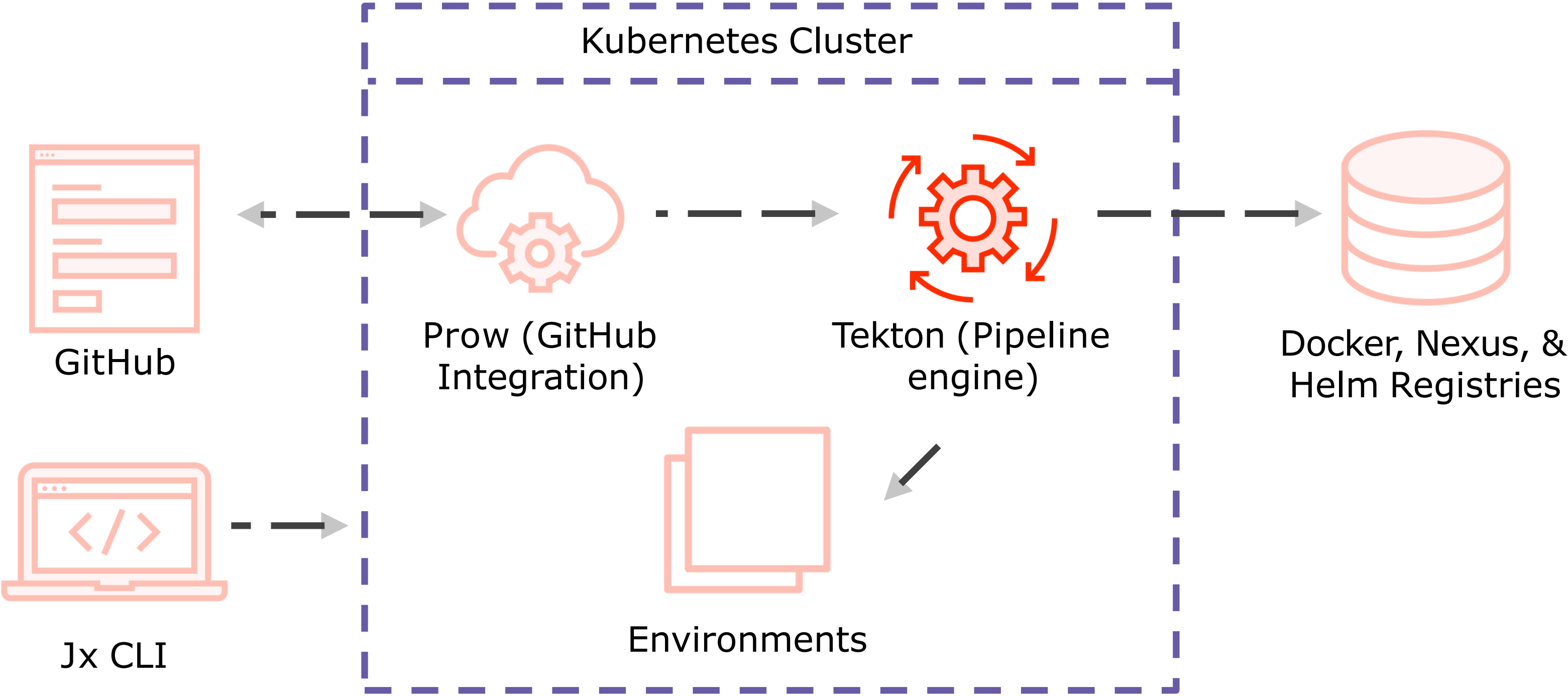
High-level Jenkins X Architecture



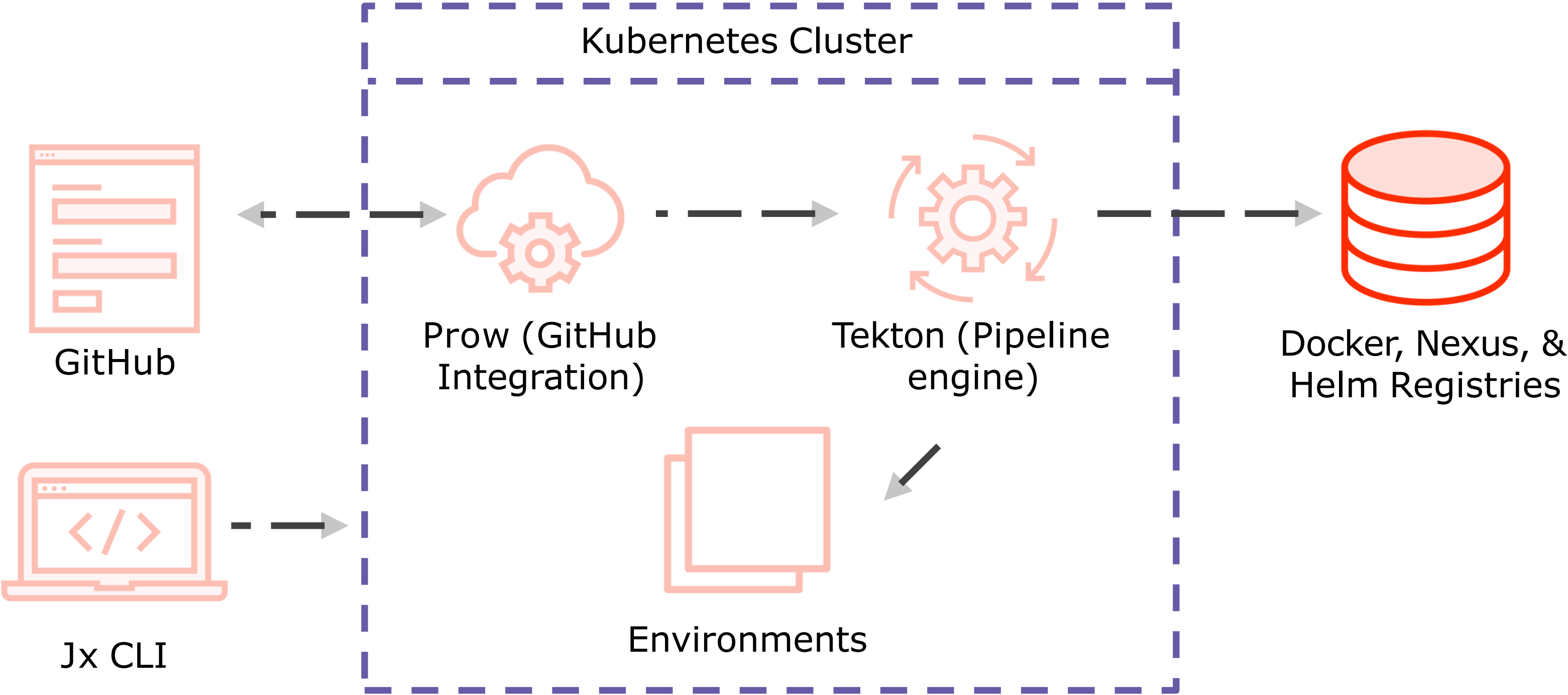
High-level Jenkins X Architecture



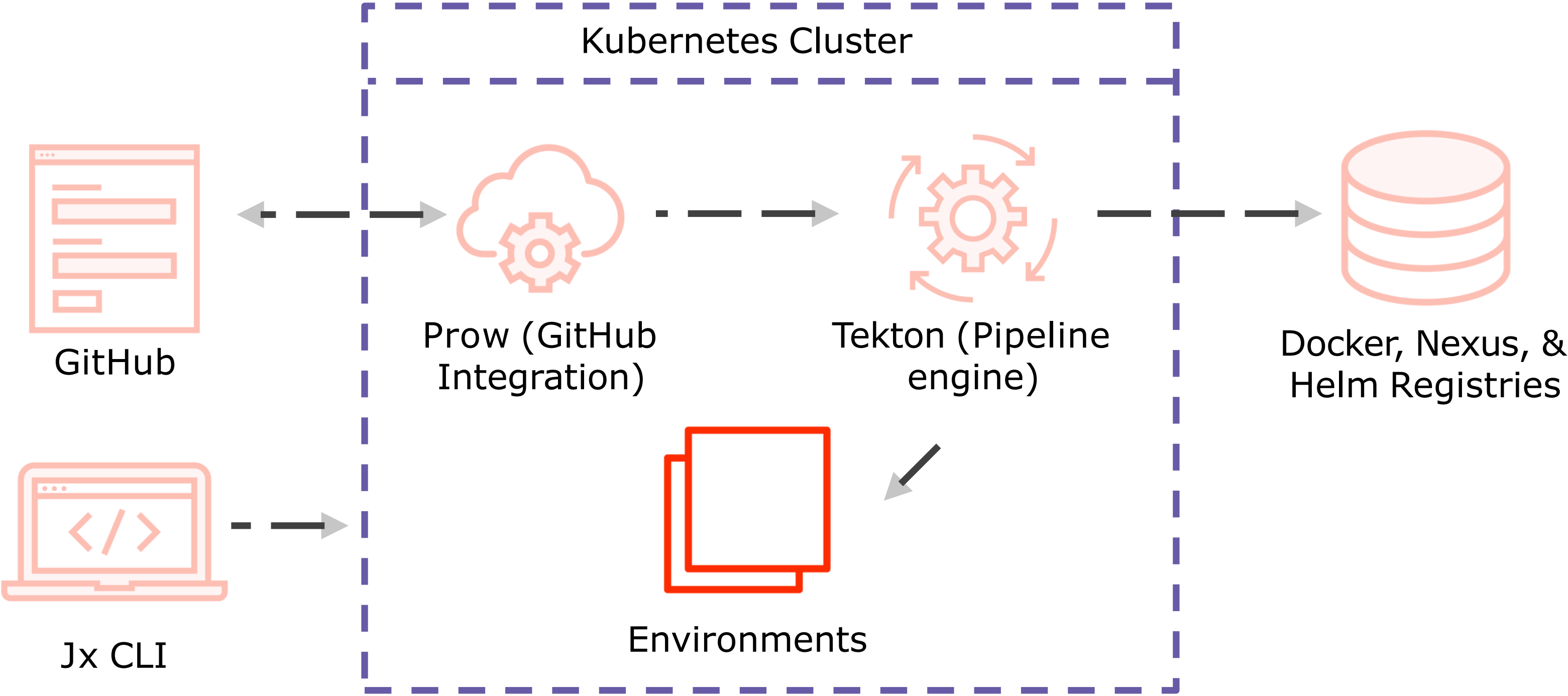
High-level Jenkins X Architecture



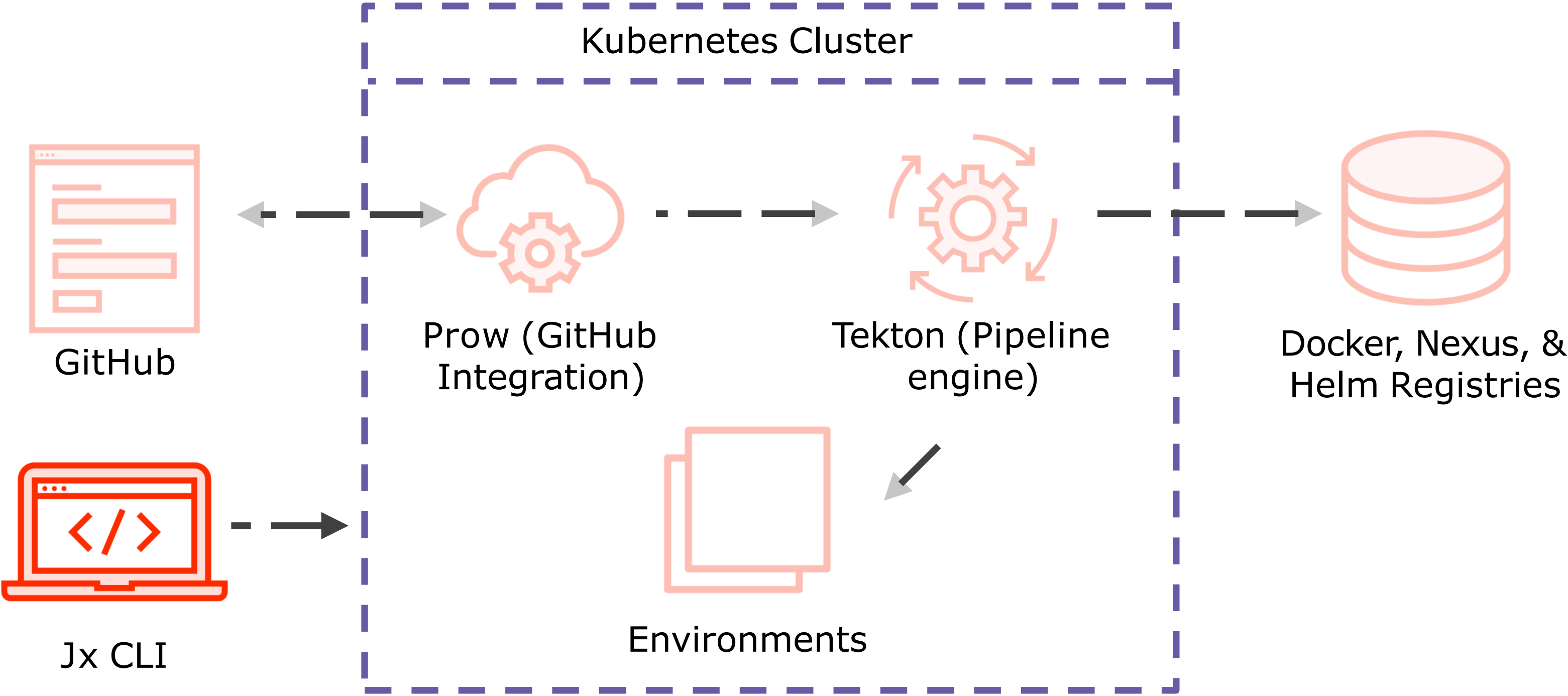
High-level Jenkins X Architecture



High-level Jenkins X Architecture



High-level Jenkins X Architecture



The Differences between Classic Jenkins and Jenkins X

Classic Jenkins

General purpose CI/CD server

Runs wherever you like

Requires a custom deployment and packaging implementation

You must set up your own infra for hosting environments

Pipeline and project creation boilerplate

Was originally the pipeline engine for Jenkins X

Jenkins X

Entire end-to-end CI/CD platform

Runs on Kubernetes

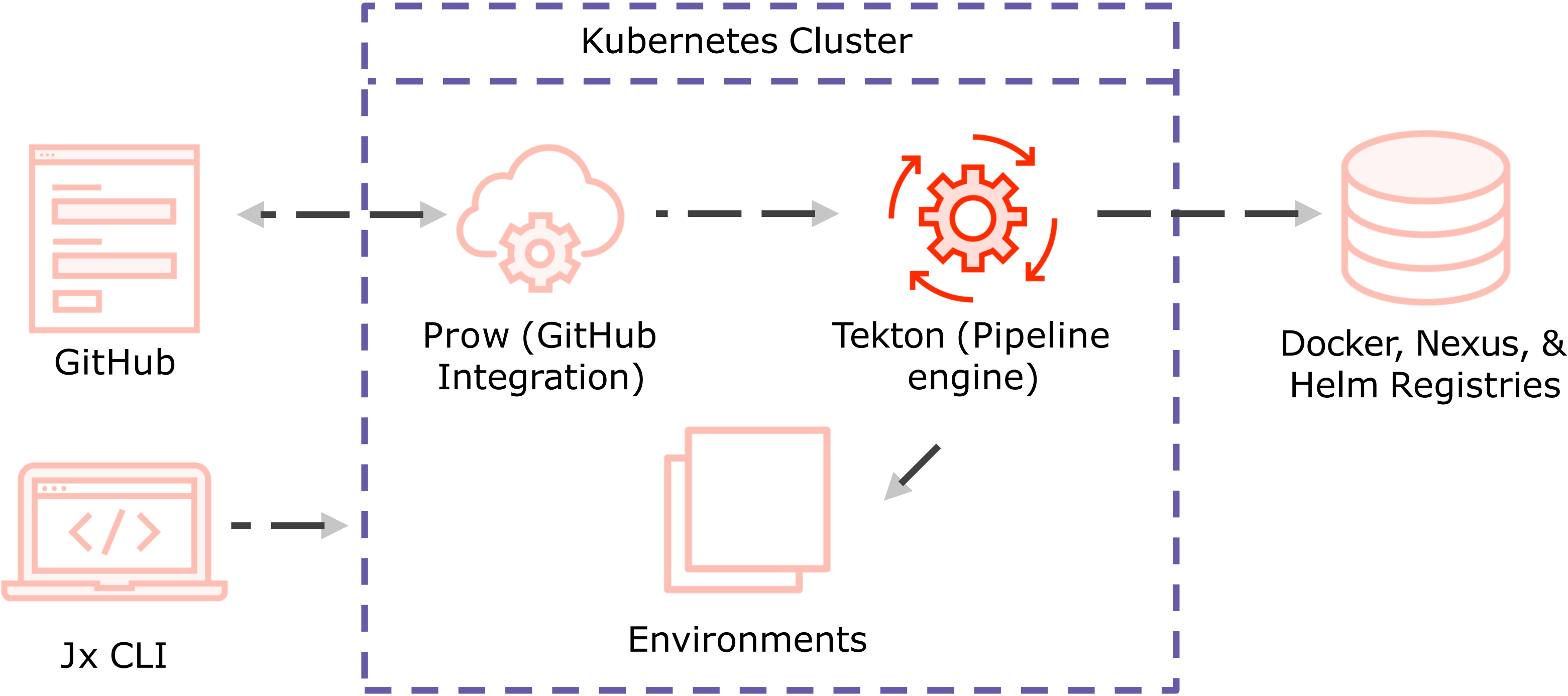
Applications use Helm and Docker for packaging and deployments

Environments up and running by default within Kubernetes

Sensible defaults for projects and pipelines

No longer used by Jenkins X all and replaced with Tekton

The Difference to Classic Jenkins



Summary

Classic CI/CD with Jenkins is not opinionated enough and requires too much customization

Jenkins X is an opinionated ecosystem which gives you an entire end-to-end CI/CD platform out of the box

Classic Jenkins was originally the pipeline engine for Jenkins X, but is now no longer part of it