|  |  |
| --- | --- |
|  | **Overview**  [Why Cloud Native?](#_nmvbjn85po5g)  [Cloud Native Key Characteristics](#_ufoy7clbetek)  [DevOps](#_5tutvfmepjjz)  [Continuous Delivery](#_34xcgreb0rf4)  [Microservices](#_9lon1zx7wny5)  [Containers](#_7djrevllriep) |

### 

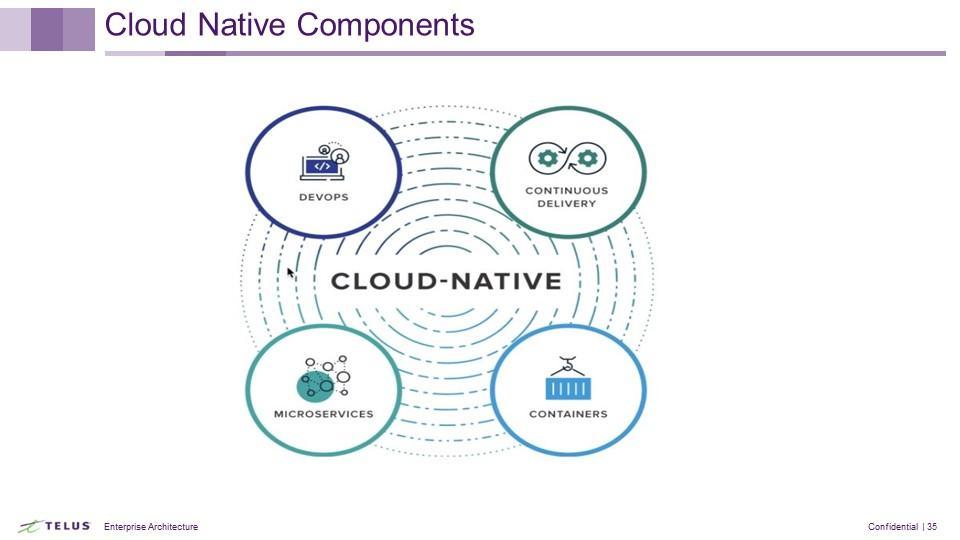
# Why Cloud Native?

* Speed – the ability to innovate, experiment and deliver value more quickly than our competitors
* Safety – the ability to move rapidly but also maintain stability, availability and durability
* Scale – the ability to elastically respond to changes in demand
* Mobility – the ability for our customers to interact with us seamlessly from any location, on any device and at any time.

# Cloud Native Key Characteristics

* Self-Service Agile Infrastructure: Enable development teams to operate at an application and service abstraction level, providing infrastructure-level speed, safety and scale.
* Microservices
  + An architecture pattern that helps us align our units of deployment with business capabilities allowing each capability to move independently and autonomously and in turn faster and safer.
* API-Base Collaboration
  + An architecture pattern that defines service-to-service interaction as automatically verifiable contracts, enabling speed and safety through simplified integration work.
* Twelve-factor applications
  + A collection of patterns for cloud-native application architectures
  + Focused on speed, safety and scale by emphasizing
    - Declarative configuration
    - Stateless/shared nothing processes that horizontally scale
    - Loose coupling to the deployment environment

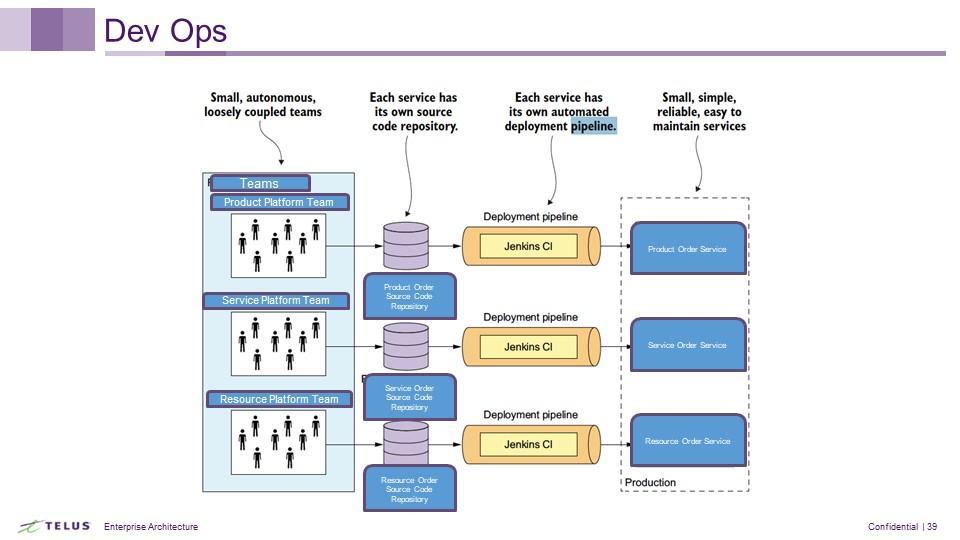
The components of Cloud Native include Dev Ops, Continuous Delivery, Microservice and Containers.



Cloud Native allows for the full adoption of DevOps and realization of its value and still allows for best-of -breed languages and frameworks providing the developer many options, the ability to have highly automated processes, and to be policy driven.

## DevOps

The DevOps processes, best practise and tool recommendations can be found in the API Development Best Practices documents.



## Continuous Delivery

## Microservices

## Containers

## Portability