

# Curso 523 – Aula 04



V18.03

**4LINUX**  
OPEN SOFTWARE SPECIALISTS

## Sumário

Aula 04 – Ferramentas DevOps.....	3
4.1 DevOps não é ferramenta.....	4
4.2 DevOps e Open Source.....	4
4.3 Categorias de Ferramentas.....	6
4.4 Planejamento e colaboração.....	7
4.5 escrita do código.....	7
4.6 Processo de Build.....	8
4.7 Ferramentas de testes.....	9
4.8 O processo de Deploy e Operações.....	10
4.9 Monitoração e métricas.....	11

## Aula 04 – Ferramentas DevOps



# DevOps Essentials

## Ferramentas DevOps

1

Entender a relação do DevOps  
com Open Source

2

Conhecer as principais  
ferramentas DevOps

## 4.1 DevOps não é ferramenta

## DevOps não é Ferramenta!



“A DevOps não é um produto ou ferramenta -  
Consiste uma cultura que evoluiu organicamente para  
atender às necessidades do ritmo mais rápido de TI”

“Até 2020, as iniciativas do DevOps farão com que  
50% das empresas implementem testes contínuos  
usando frameworks e ferramentas de código aberto.”

[Gartner]



## 4.2 DevOps e Open Source

## DevOps e Open Source

## FACEBOOK

“ Facebook is built on Open Source from top to bottom, and could not exist without it. As engineers here, we use, contribute to, and release a lot of open source software, including pieces of our core infrastructure such as HipHop and Thrift.”

Source: <https://goo.gl/DBi1MF>  
Dev Ops: <https://goo.gl/dD2Zun>



## AWS

“We view open source as a companion to AWS's business model,” said Chris Schlaeger, director of software development at Amazon Web Services.  
Linux, in the form of Amazon Linux as well as Xen are fundamental technologies for AWS.

Source: <https://goo.gl/KAHypf>  
Dev Ops: <https://goo.gl/MC8wkk>



## Open Source

As ferramentas DevOps, em sua maioria, operam sob a filosofia Open Source. A união dessas duas culturas está em expansão. Como já conhecemos os pilares do DevOps, vale apenas mencionar a estrutura Open Source que o apoia.

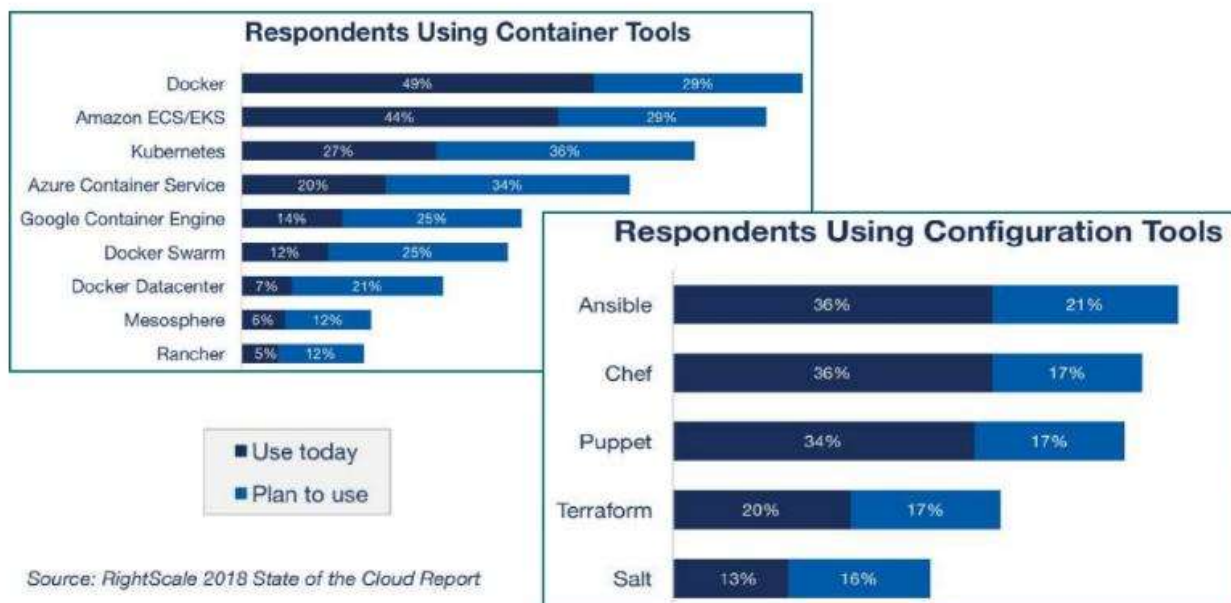
Baseado no texto original da Debian Free Software Guidelines, a Open Source Initiative, define que um programa de código aberto deve garantir:

- distribuição Livre.
- código fonte.
- trabalhos derivados.
- integridade do autor do código fonte.
- não discriminação contra pessoas ou grupos.
- não discriminação contra áreas de atuação.
- distribuição de licença.
- licença não específica a um produto.
- licença não restrinja outros programas.
- licença neutra em relação a tecnologia.

Para maiores detalhes visite: <https://opensource.org/osd>

# Mas qual é a melhor Ferramenta?





### 4.3 Categorias de Ferramentas



- ✓ Em linhas gerais, o ciclo DevOps é composto pelas seguintes fases: Planejamento, Código, Construção, Testes, Deploy, Operação e Monitoração.
- ✓ Para cada etapa, recursos e funcionalidades específicos são demandados, por isso é importante combinar e encaixar as ferramentas mais apropriadas a cada momento do processo.

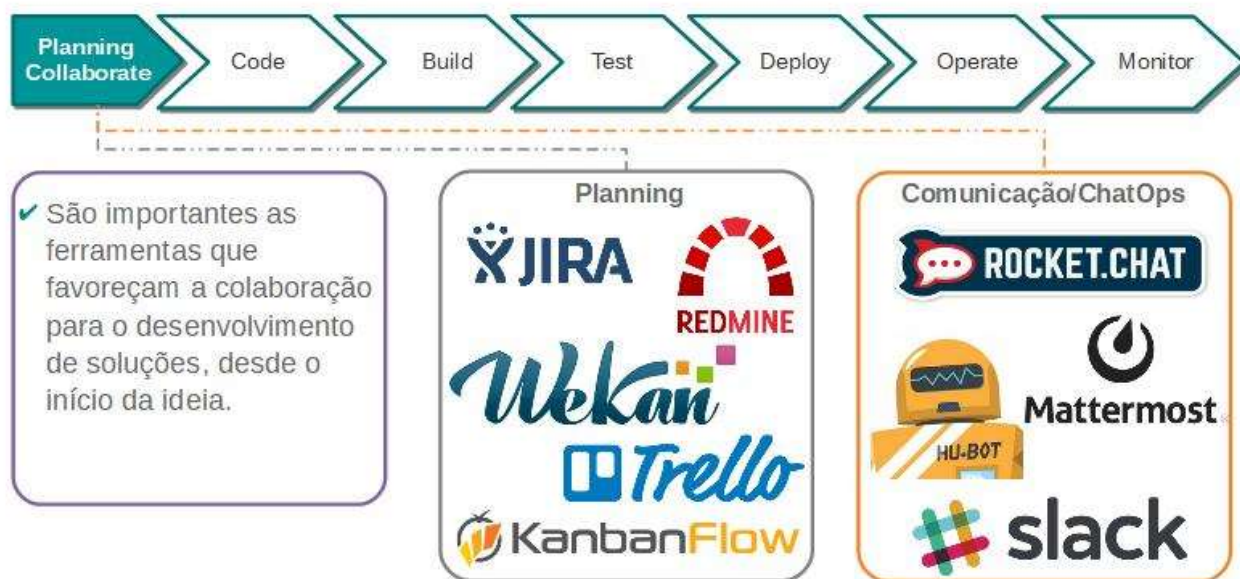


### Ferramentas DevOps

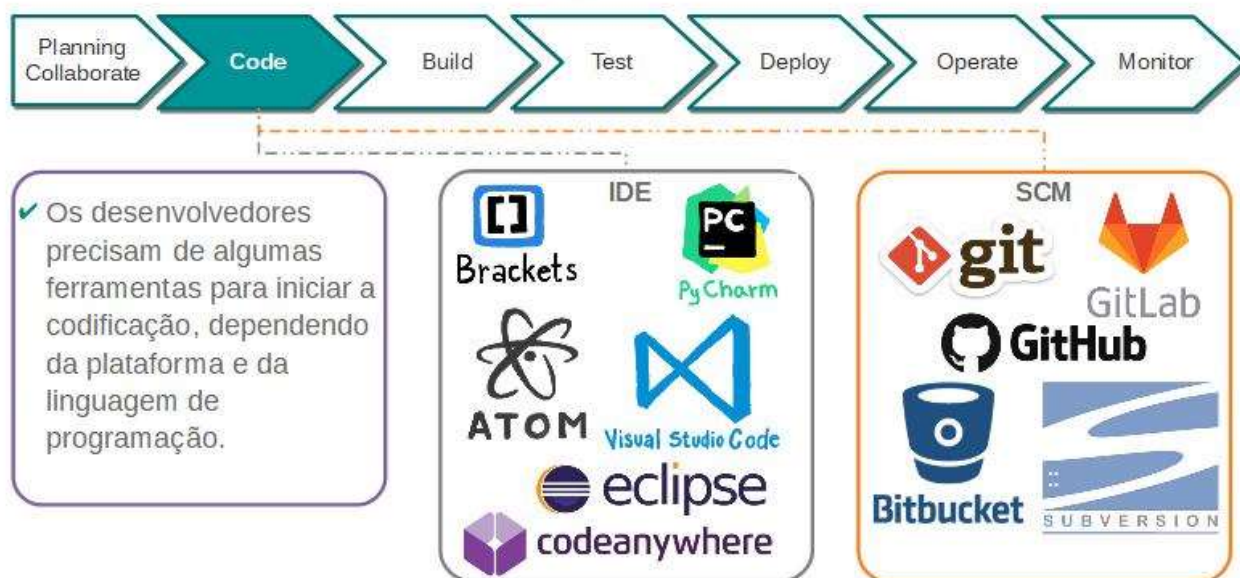
Um dos alicerces do DevOps é a automação dos processos de mudança, configuração e deploy. Isso só é possível graças ao uso de ferramentas que oferecem funcionalidades úteis em cada fase do fluxo.

A ideia é que as ferramentas orquestram todo o ciclo de processos, também controlem instâncias em tempo real, facilitando a administração do ambiente, seja físico ou virtual, desde a automação do desenvolvimento até a infraestrutura.

#### 4.4 Planejamento e colaboração



#### 4.5 escrita do código



## Ambiente de Desenvolvimento

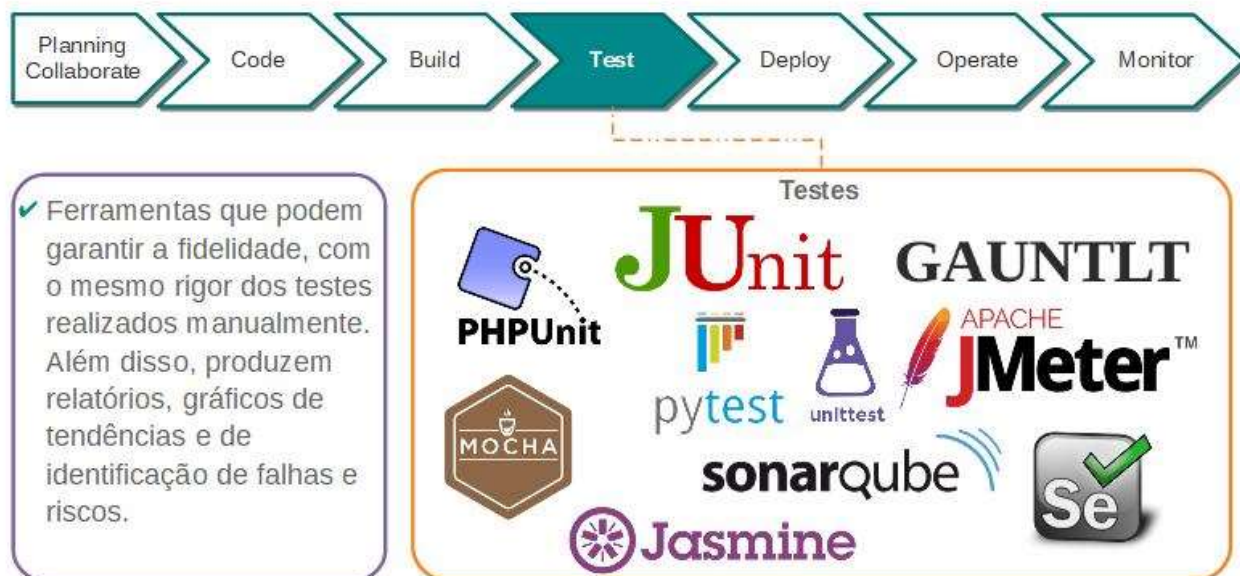
Um IDE reúne diversas características e ferramentas de apoio ao desenvolvimento do software, como por exemplo, compiladores, depuradores, ferramentas para auto-completar o código, entre outras funções.

Por isso não é possível definir o melhor IDE, já que o próprio desenvolvedor deve adapta-se ao IDE para aumentar o seu conforto na criação e, obviamente, agilizar o seu processo de desenvolvimento.

### 4.6 Processo de Build







### Tipos de Testes de Software

Os testes de software são extremamente necessários e trabalhosos. Por esse motivo, é essencial identificar quais testes são indispensáveis para o seu software. Deste modo, recomenda-se fazer a escolha da ferramenta certa, para automatizar o processo que geralmente é cansativo e repetitivo.

### Testes mais comumente utilizados

**Teste Unitário:** avalia pequenas unidades do software, códigos, sub-rotinas, etc.

**Teste de Integração:** garante que um ou mais componentes funcionam. Pode ser entendido como um conjunto de testes unitários.

**Teste de Funcionalidade:** validar as funcionalidades, requerimentos e regras de negócio que o software deve possuir.

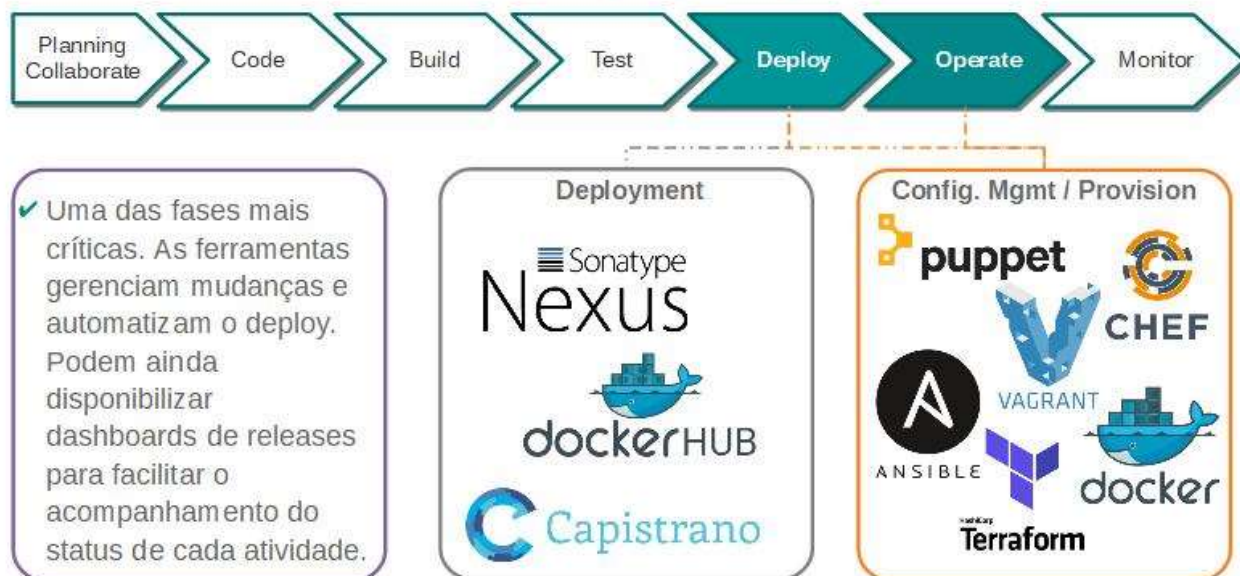
**Teste de Desempenho:** verifica o tempo de resposta e utilização de recursos do software.

**Teste de Segurança:** verifica se o software em questão não apresenta vulnerabilidades.

## 4.8 O processo de Deploy e Operações

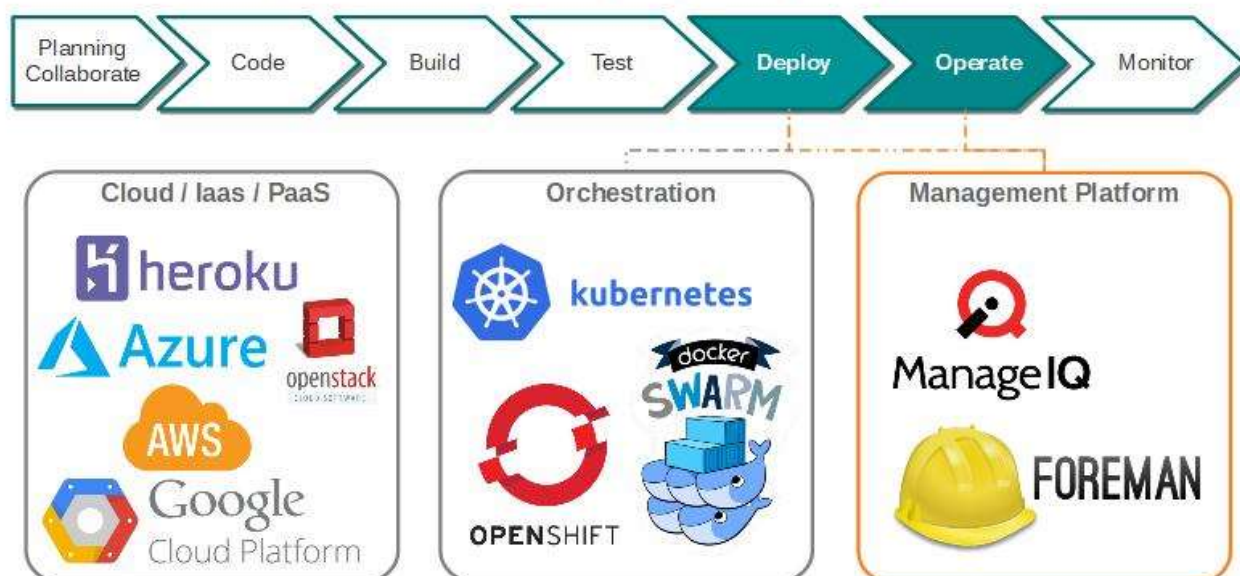
## O processo de Deploy e Operações

14



## O processo de Deploy e Operações

15



## 4.9 Monitoração e métricas



## REVIEW

- ✓ Entendemos a importância das Ferramentas no DevOps.
- ✓ Conhecemos o ciclo do DevOps.
- ✓ Exploramos as principais ferramentas para cada ciclo do DevOps.