



Construindo uma Pipeline DevOps!

Who?



Stone

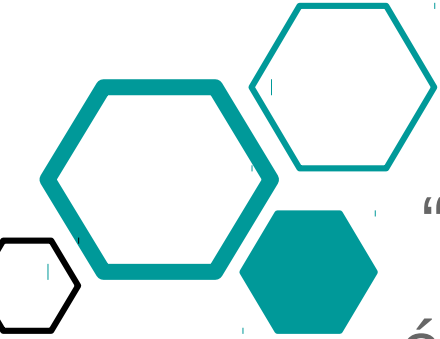
- Desde 2012 construindo um novo mercado.
- Empresa Brasileira
- +3.500 pessoas.
- Stone Co., uma companhia que une todas as atuações do mercado de pagamentos em apenas um ecossistema.



Gabriela Dias

- Gerente de SRE e DevOps
- +17 anos de experiência em Projetos FOSS (Free and Open Source Software)
- DevOps/InfraÁgil desde 2012
- 4Linux, HP Services, Mandic/Rivendel

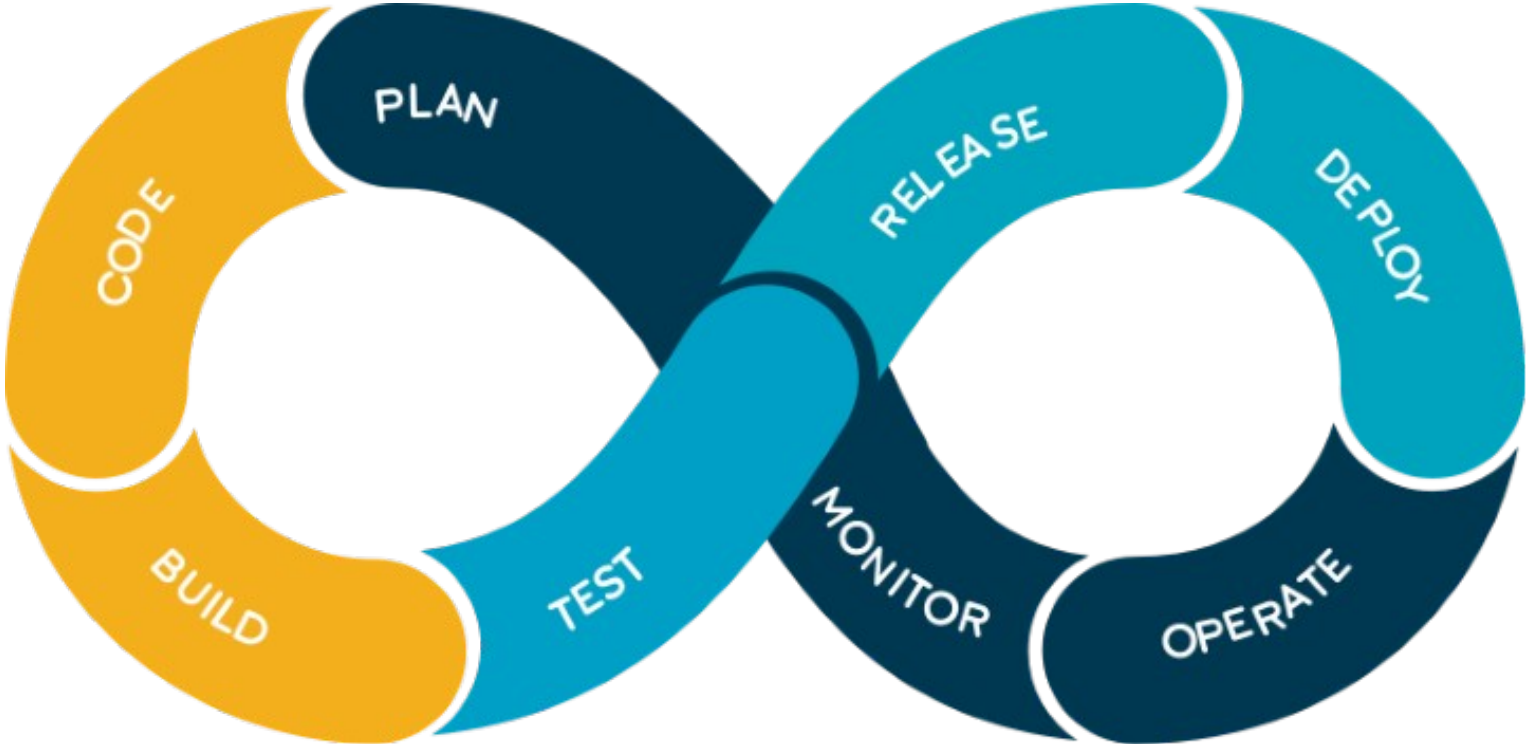
DevOps não é Ferramenta!



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



DevOps






Porque ferramentas são Importantes?

Porque Ferramentas são Importantes?

- ✓ Um dos **alicerces do DevOps** é a **automação** dos processos de mudança, de configuração e de deploy.
- ✓ Isso **só é possível graças ao uso de ferramentas** que oferecem funcionalidades úteis em cada fase do fluxo.
- ✓ As **ferramentas certas são essenciais** para estabelecer os primeiros bons resultados, que convencerão as **pessoas a abraçar uma mudança, ao invés de temê-la**.
- ✓ A ideia é que as **ferramentas orchestrem todo o ciclo** dos processos e controlem instâncias em tempo real, facilitando a administração do ambiente, seja físico ou virtual desde a automação do desenvolvimento até a infraestrutura.



Qual relação de Ferramentas, Pipelines e DevOps?



**DevOps criou uma consciência da
necessidade de automatizar e ser mais
eficiente em termos de entrega de software.**

Pipeline de Desenvolvimento



“Principal foco é a entrega de uma release de Software confiável para entrar em Produção.”

Pipelines DevOps ou Assembly Line



Pipeline DevOps ajudam a automatizar e dimensionar fluxos de trabalhos de ponta a ponta em todas as equipes e ferramentas, permitindo de fato uma entrega contínua.

✓ Pipeline é um reflexo do seu processo!!!

<Processo Torto gera Pipeline Torta>

“Um pipeline de pipelines”

✓ Software/Infra sem Teste é pipeline automatizando “@#\$%” em produção!



Pipeline DevOps



Agile Development

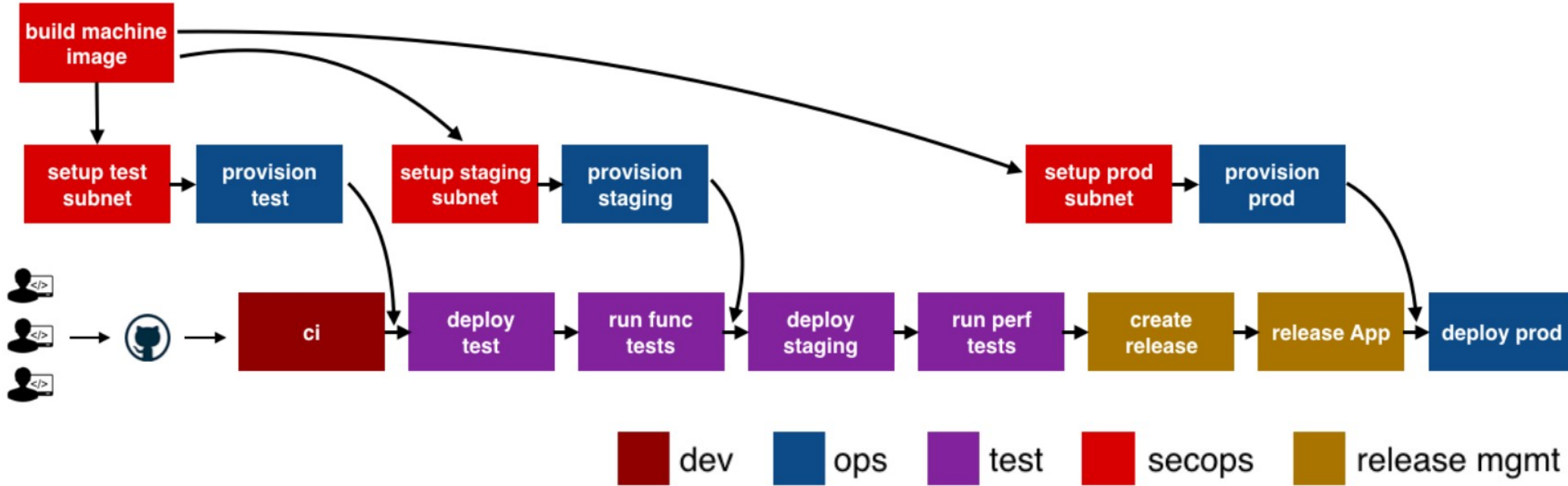
Continuous Integration

Continuous Delivery

Continuous Deployment

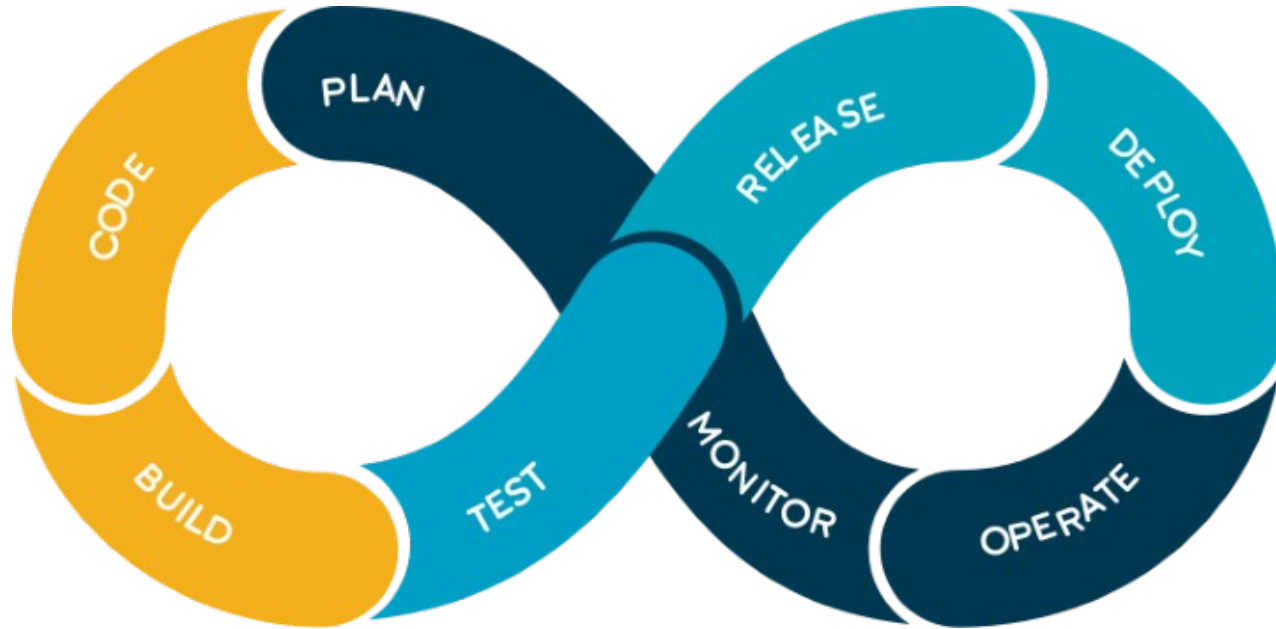
DevOps

Pipeline DevOps

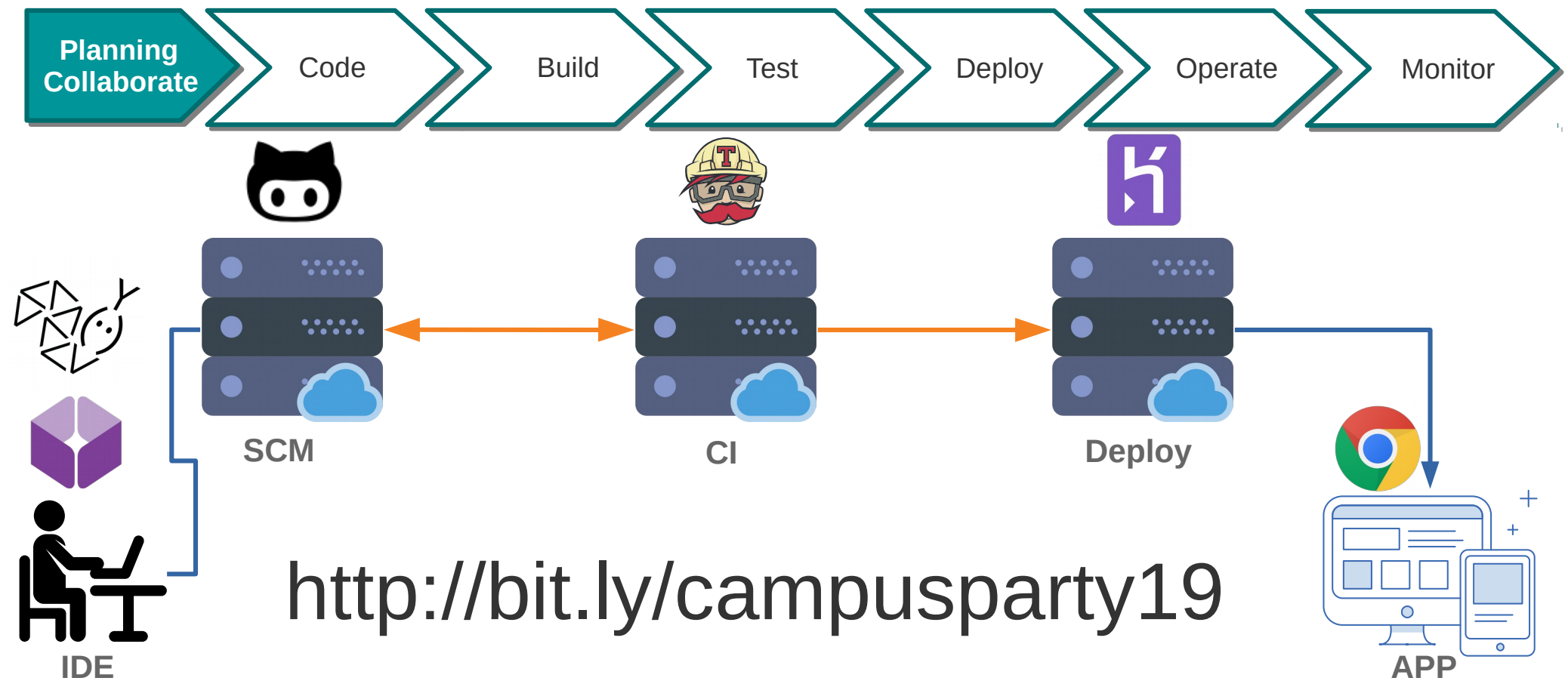


“A cadeia de ferramentas de DevOps é fragmentada e colá-la para conseguir uma entrega contínua é uma tarefa desafiadora.”

Pipeline DevOps



Pipeline DevOps



<http://bit.ly/campusparty19>

Planejamento e Colaboração

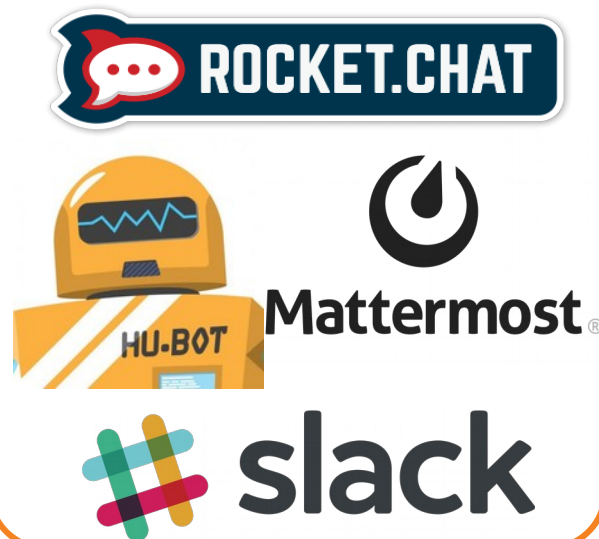


- ✓ São importantes as ferramentas que favoreçam a colaboração para o desenvolvimento de soluções, desde o início da ideia.

Planning



Comunicação/ChatOps



Planejamento e Colaboração



✓ Podemos usar o próprio GitHub para planejar nosso projeto.

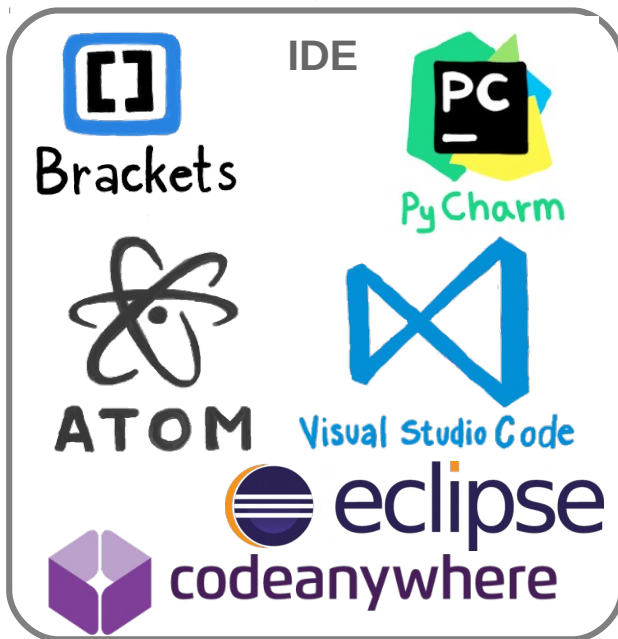
The screenshot shows a GitHub project board for the repository 'gabydias / campusparty19'. The board is titled 'Python App' and has a progress bar. It contains three columns:

- To do (4 items):**
 - ✓ Criar o Teste Unitário (Added by gabydias)
 - ✓ Criar a Pipeline no Travis-CI (Added by gabydias)
 - ✓ Integrar com o Sonar (Added by gabydias)
 - ✓ Fazer Deploy no Heroku (Added by gabydias)
- In progress (1 item):**
 - ✓ Criar a Aplicação (Added by gabydias)
- Done (0 items):**

Escrita do Código



- ✓ Os desenvolvedores precisam de algumas ferramentas para iniciar a codificação, dependendo da plataforma e as linguagem de programação.



Escrita do Código



1

Faça uma alteração no código para enviarmos as atualizações ao GitHub.



2

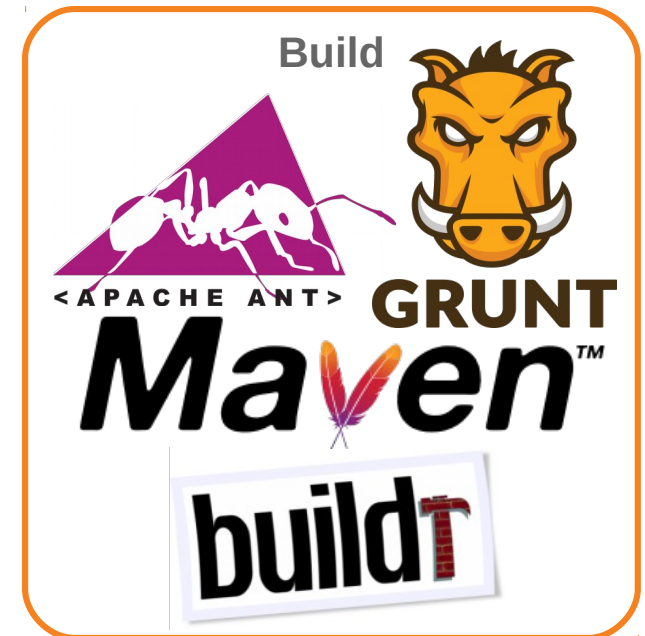
Subindo o código para o Repositório do GitHub:

```
# git add --all  
# git status  
# git commit -m "Update Project"  
# git push origin master
```

Processo de Build



- ✓ A automação de build agiliza tarefas típicas no desenvolvimento de software, como compilação, distribuição, implantação (instalação e configuração) e limpeza.



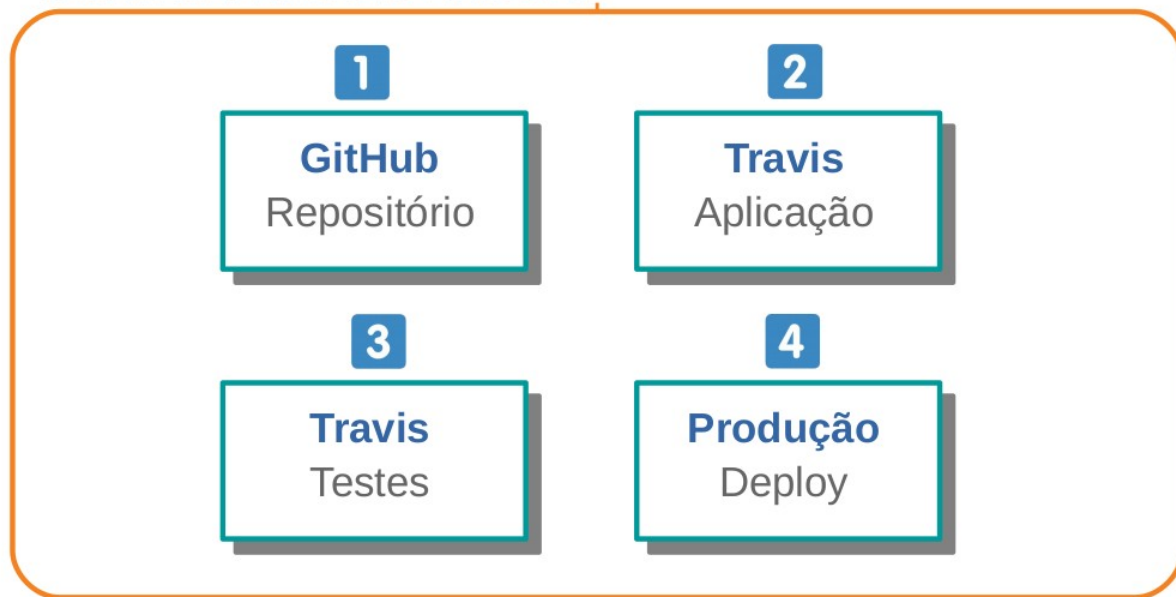
Processo de Build



- ✓ O Travis CI é uma aplicação de Integração Contínua, é totalmente gratuito para projetos de código aberto hospedados no GitHub



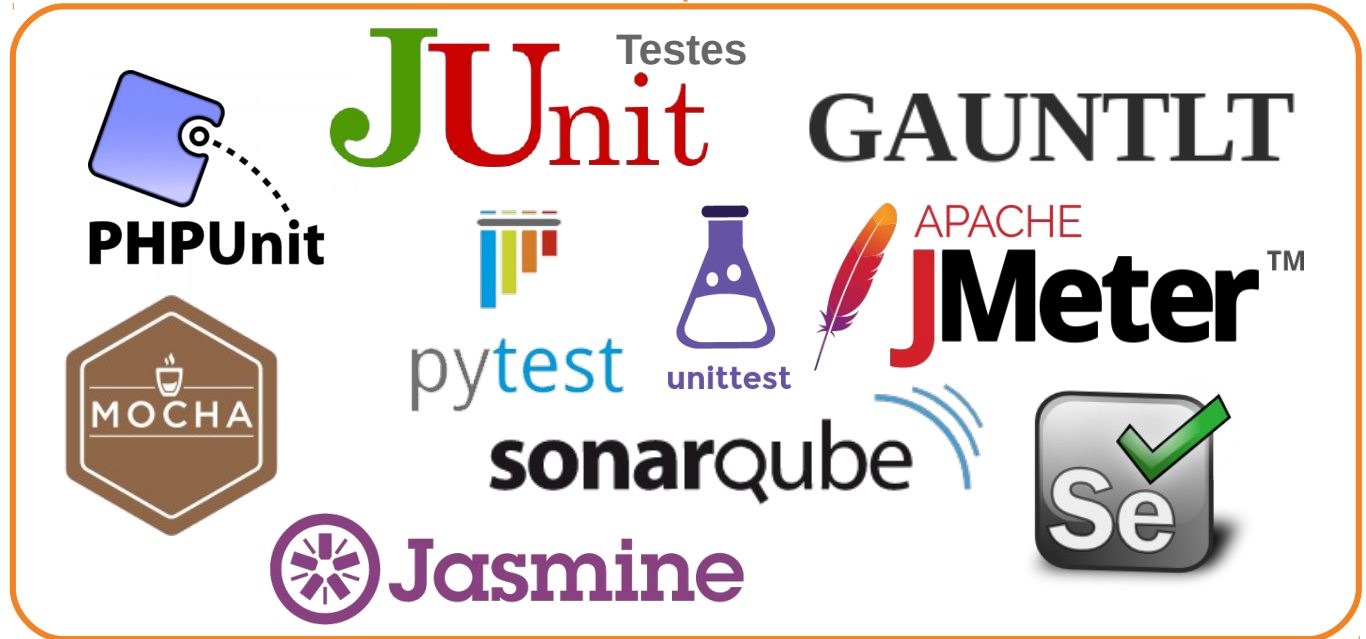
Travis CI



Ferramentas de Testes



- ✓ Ferramentas que podem garantir a fidelidade dos testes, com o mesmo rigor dos realizados manualmente. Além disso, produzem relatórios e gráficos de tendências e de identificação de falhas e riscos.



Ferramentas de Testes



- ✓ O unittest segue a estrutura de estilo xUnit.
- ✓ É um módulo padrão já está no python a partir da versão 2.1.



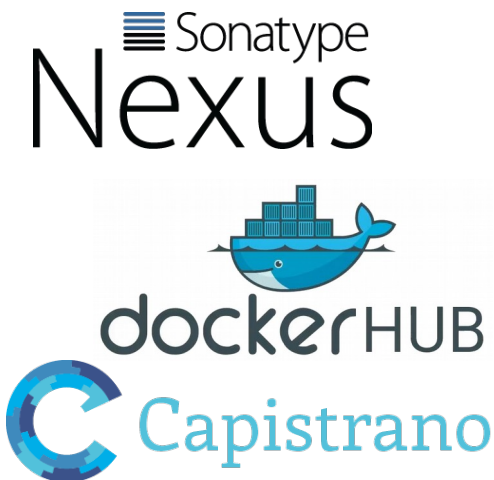
```
test.py
1  from app import app
2  import unittest
3
4  class Test(unittest.TestCase):
5
6      def setUp(self):
7          # cria uma instância do unittest, precisa do nome "setUp"
8          self.app = app.test_client()
9
10     def test_requisicao(self):
11         # envia uma requisicao GET para a URL
12         result = self.app.get('/')
```

O processo de Deploy e Operações

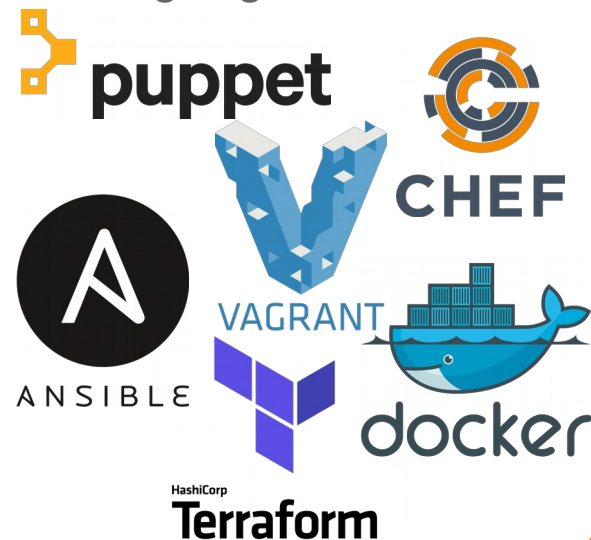


- ✓ Uma das fases mais crítica, as ferramentas gerenciam mudanças e automatizam o deploy. Podem ainda disponibilizar dashboards de releases para facilitar o acompanhamento do status de cada atividade.

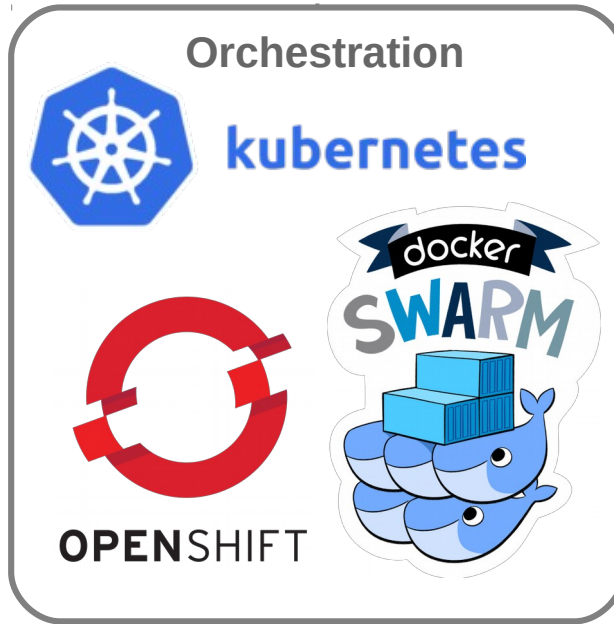
Deployment



Config. Mgmt / Provision



O processo de Deploy e Operações



O processo de Deploy e Operações



- ✓ Heroku é PaaS (Platform as a Service) que suporta diversas linguagens de programação (Java, PHP e Python). Ele usa uma unidade de máquina virtual chamada "Dyno" que são como contêineres que permitem que o desenvolvedor execute a sua aplicação em um ambiente seguro.



Getting Started with Heroku

If you're new to Heroku, choose your language and follow our Getting Started guide to create a new app.



Ruby
Get Started



PHP
Get Started



Node.js
Get Started



Python
Get Started



Java
Get Started



Go
Get Started



Clojure
Get Started



Scala
Get Started

Create New App

Monitoração e Métricas



- ✓ Monitoramento automatizado de infraestrutura e desempenho de aplicações.
- ✓ É importante que as soluções sejam integradas com canais para envio de alertas para que os times se mobilizem prontamente, em caso de incidentes.

Monitor / Logs / Analytics

ZABBIX



Prometheus



Grafana



elastic

graylog

splunk>

kibana



logstash

Monitoração e Métricas



Response Time



Throughput (requests/min)



Desafios de uma Pipeline DevOps

A **entrega de software vai muito além dos desenvolvedores** e envolve várias equipes, como Test, Operations, Security, SecOps e Release Management. Isso significa que qualquer plataforma ou solução que almeje ajudar a obter entrega ou implantação contínua precisa **abordar os fluxos de trabalho de todas essas equipes, além de gerenciar as interações entre elas.**

Precisamos transformar as “ilhas de automação” em soluções holísticas, fluxos de trabalho interconectados para conectar os fluxos de trabalho de ponta a ponta com total visibilidade, rastreabilidade e auditabilidade.

Considerações Finais

- ✓ Dicas para construir uma Pipeline DevOps?
 - ✓ **Não fique apenas no CI**, escolha algumas áreas para adicionar em seu processo;
 - ✓ **Comece por um projeto** de software e mapeie todas as etapas do processo de liberação, desde o design até a produção;
 - ✓ Faça um inventário das ferramentas de DevOps usadas em cada etapa;
 - ✓ Pesquise uma **solução** que, no mínimo, **integre e orquestra todas as suas ferramentas** existentes ou escolhidas;
 - ✓ **Treine seu Time**, promova a colaboração dos times;
 - ✓ Determinar alguns **KPIs principais para medir o sucesso**, como tempo de entrega, frequência de implementação, volume de alterações, taxa de sucesso e tempo médio de recuperação.



PORQUE TUDO EVOLUI.

Obrigada.

Gabriela Dias

gabriela.dias@stone.com.br / gabydias@gmail.com

<https://br.linkedin.com/in/gabydias>

<https://speakerdeck.com/gabydias>