



Estratégia

Concursos

//estratégia tech



ESTRATEGIA
CONCURSOS



<http://anchor.fm/estrategia-tech>

http://t.me/estrategia_ti



<http://instagram.com/estrategiaconcursosti>



INTEGRAÇÃO CONTÍNUA

Prof. Raphael Lacerda

<https://www.instagram.com/p/B74PyOzj5V3/>

<https://www.estategiaconcursos.com.br/blog/prazer-um-eterno-estudante/>



Esse CURSO É PARA VOCÊ?

Prof. Raphael Lacerda



Questão de alinhamento

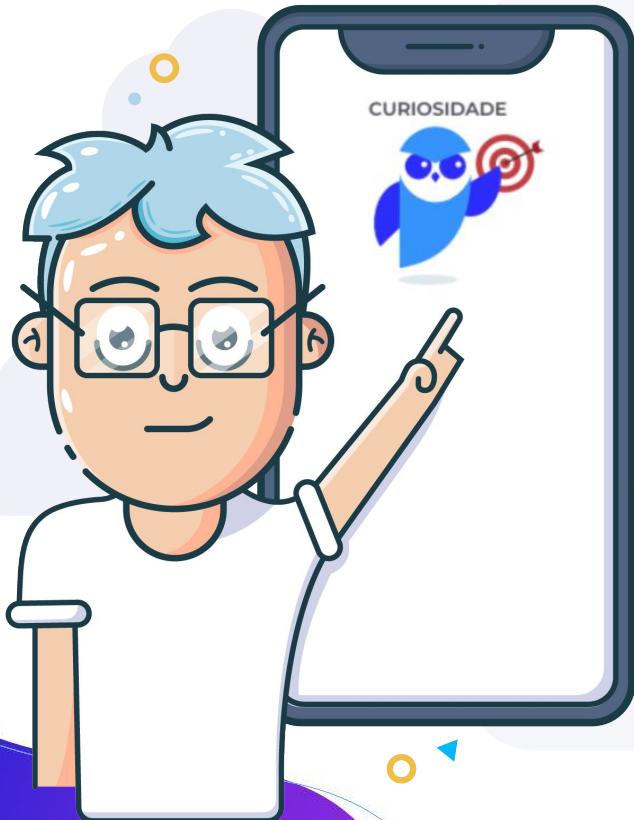
Integração Contínua é um dos pilares das práticas ágeis. Um servidor de Integração Contínua open source, escrito em Java, é o:

- A JServer
- B Integration
- C Symphony
- D Jenkins



SE VOCÊ ERROU...

Prof. Raphael Lacerda



Estratégia

Concursos

SEU LUGAR É AQUI!!!



INTEGRAÇÃO CONTÍNUA

Prof. Raphael Lacerda



AGORA MANDA PARA PRODUÇÃO!

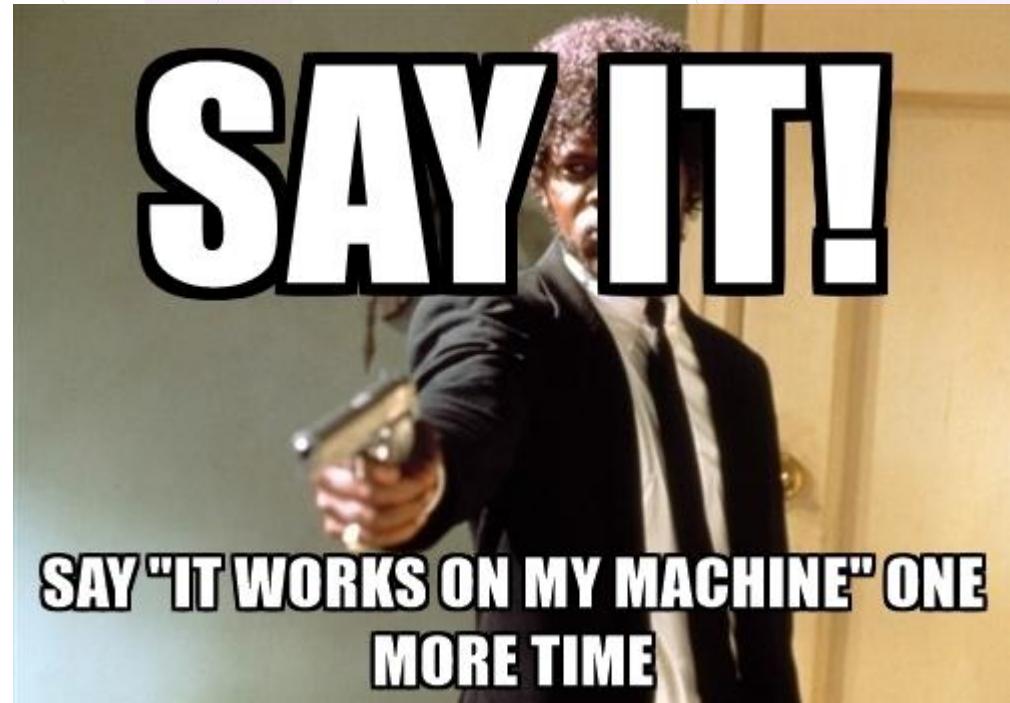
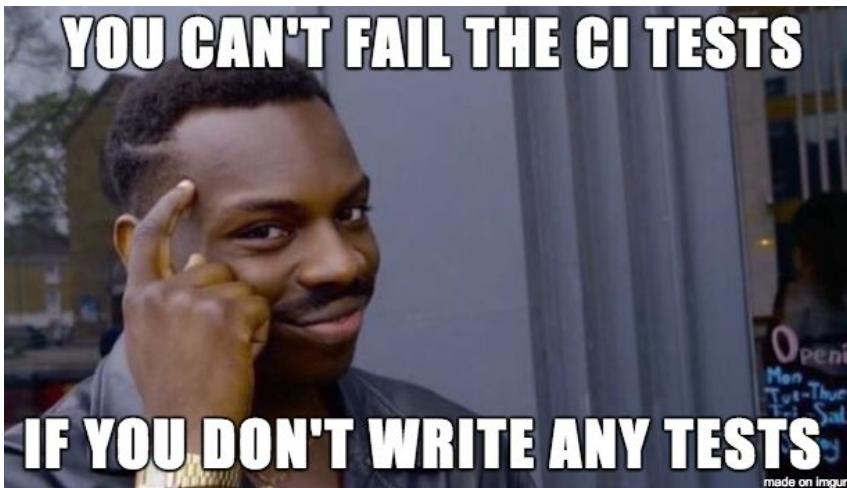


Prof. Raphael Lacerda

Ambiente Integração Contínua

- Jenkins, Circle CI, Gitlab, Team City
- Artifactory, Nexus
- Continuous Deployment x Continuous Delivery
- Feature Toogle (git flow x trunk based development)
- Pipelines
- Questões

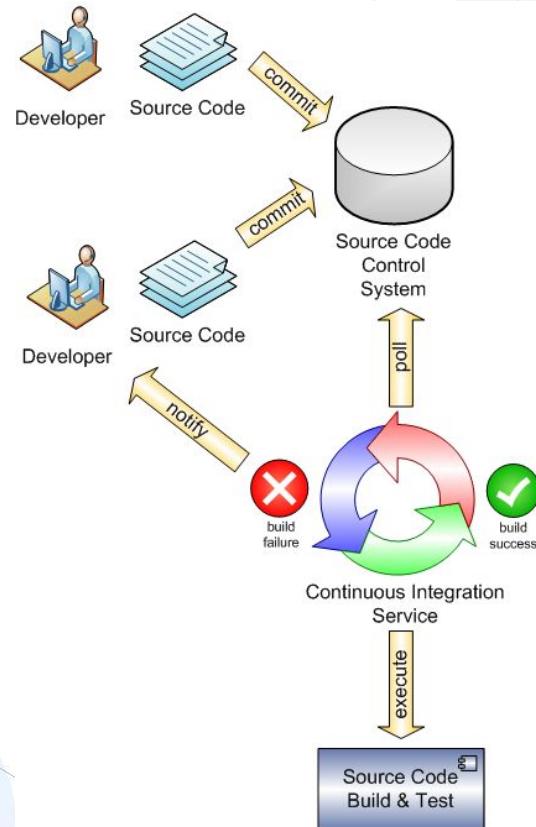
Por que integrar?



Integrando Código



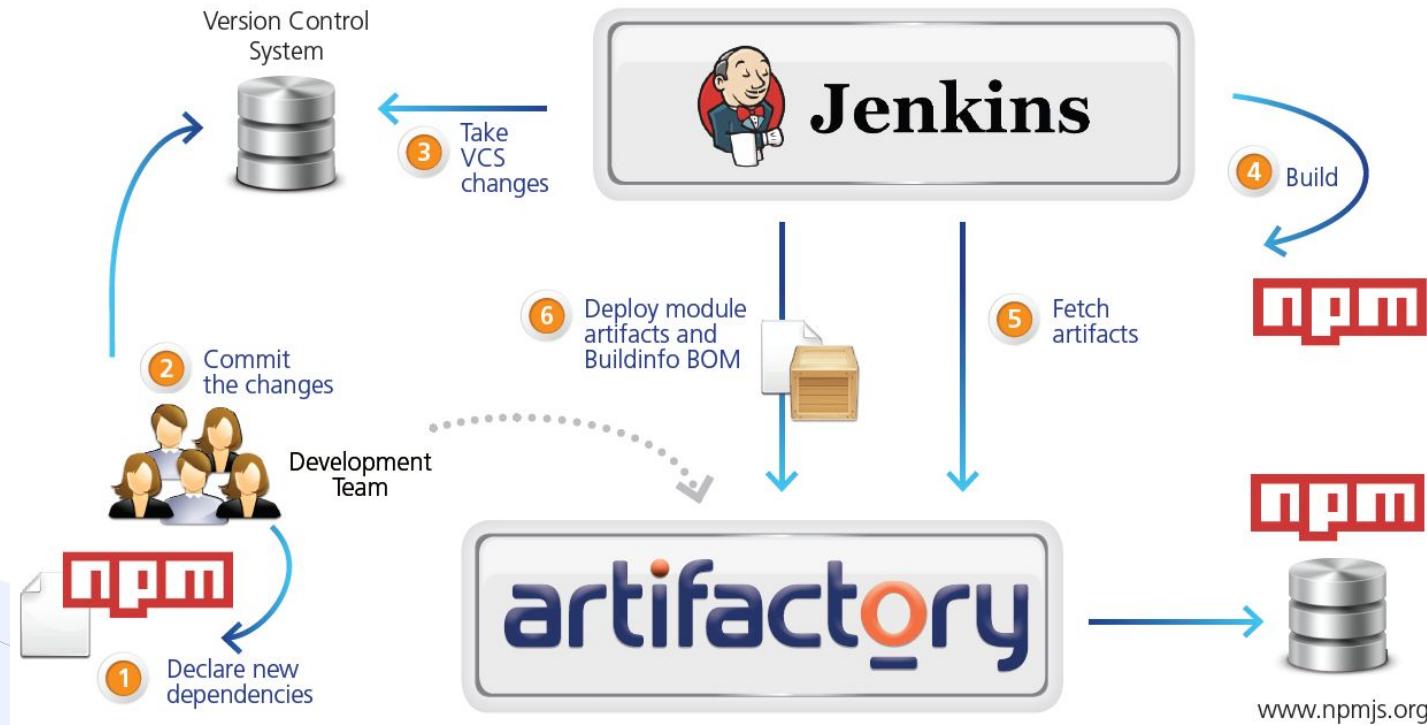
AIC resumido



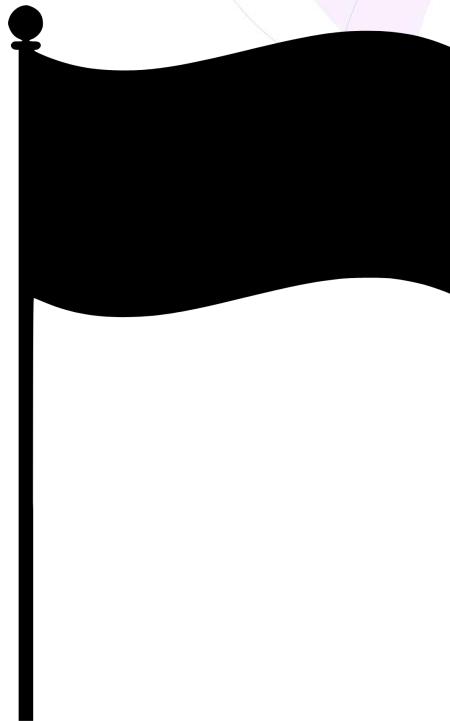
AIC detalhado



AIC detalhado



Feature Toogle/Flag e Pipelines

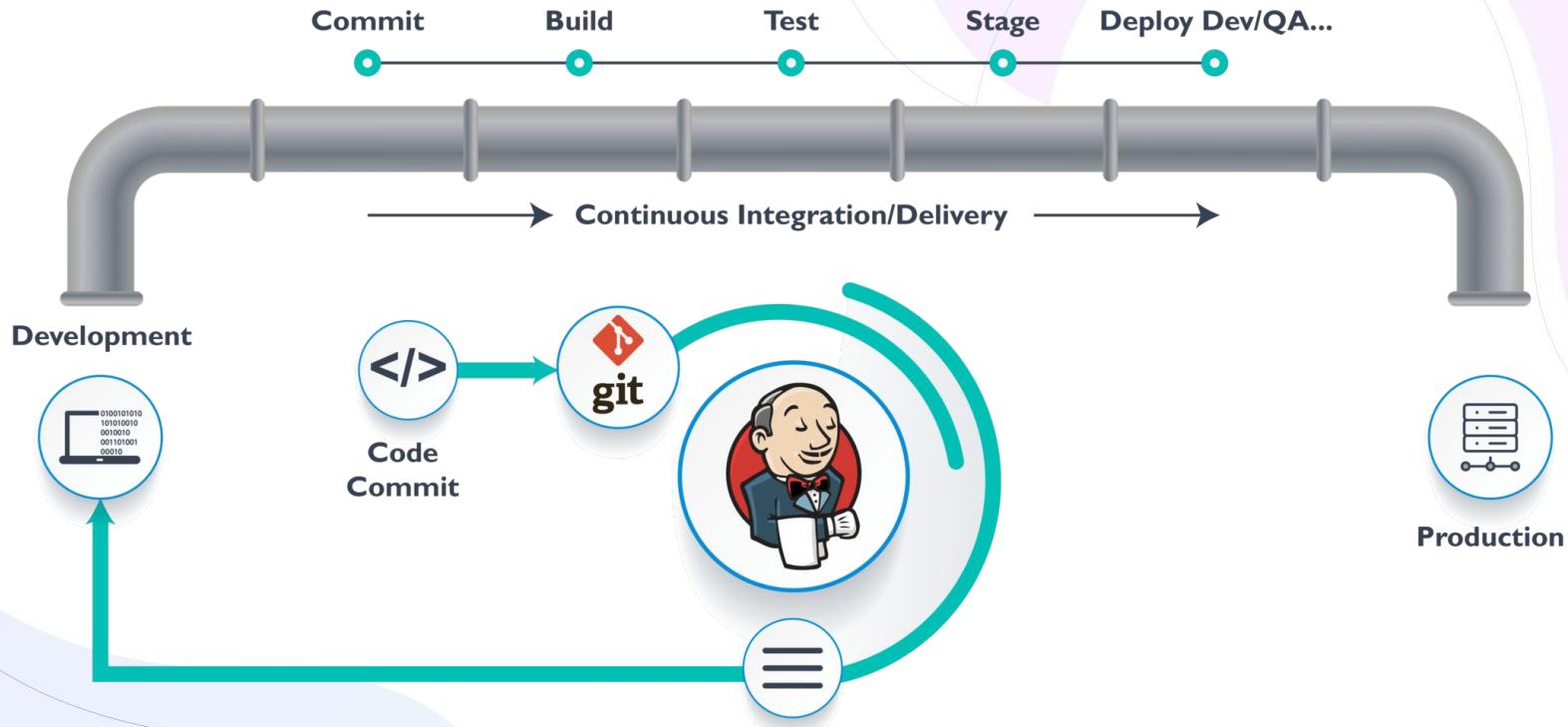


```
function reticulateSplines () {
    if(
featureIsEnabled( "use-new-SR-algo
rithm" )  ){
        return
enhancedSplineReticulation();
    }else{
        return
oldFashionedSplineReticulation();
    }
}
```

Pipelines



Pipelines



<https://medium.com/faun/jenkins-pipeline-tutorial-first-step-guide-to-continuous-delivery-87f74d322ab7>

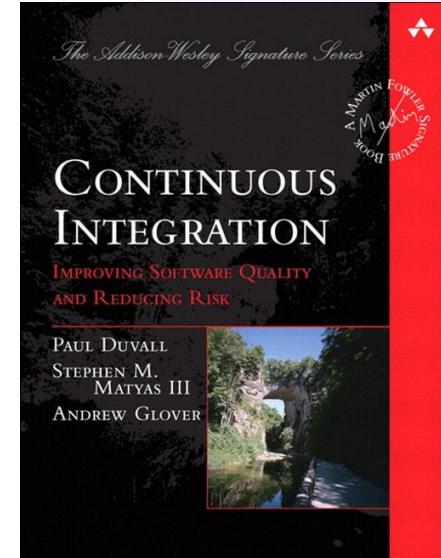
Conceitos!



Integração Contínua
Prof. Raphael Lacerda

<https://martinfowler.com/bliki/ContinuousDelivery.html>

07074192716 - Jefferson Terra



<https://martinfowler.com/delivery.html>

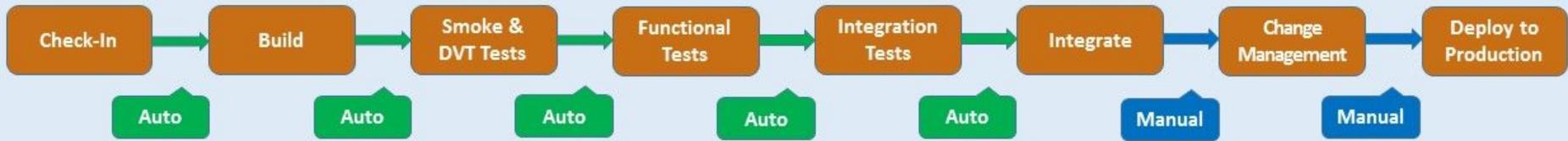
<https://martinfowler.com/articles/continuousIntegration.html>

Continuous Model

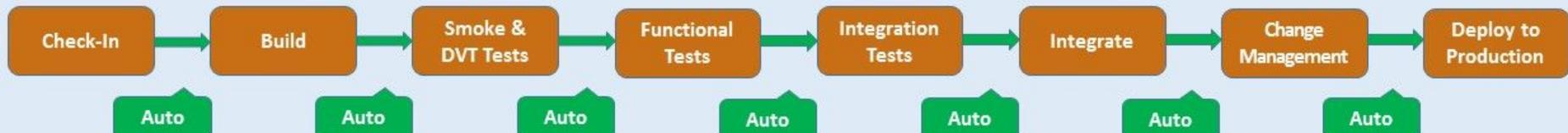
Continuous Integration



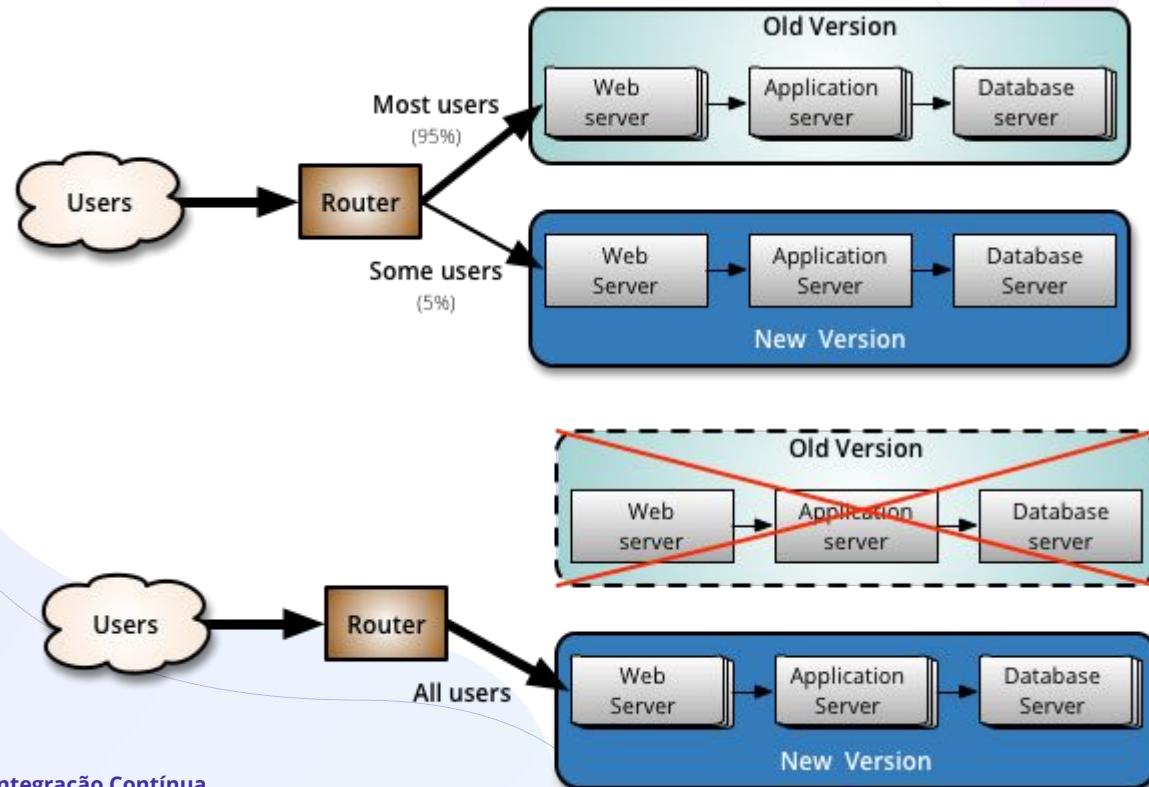
Continuous Delivery



Continuous Deployment



Canary deployment

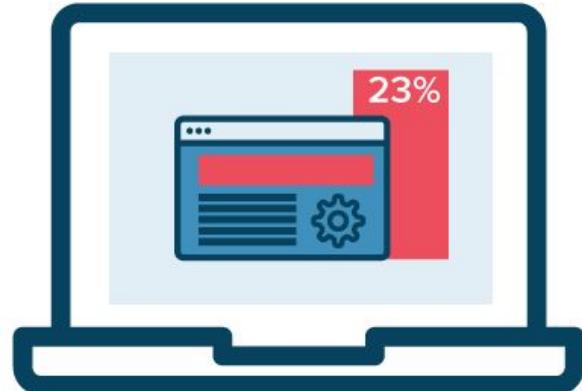


<https://octopus.com/docs/deployment-patterns/canary-deployments>

<https://martinfowler.com/bliki/CanaryRelease.html>

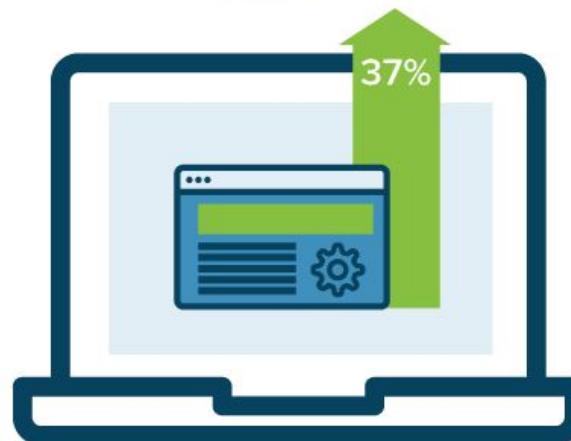
Teste A/B

A



CONTROL

B



VARIATION

Praticando a Integração





INTEGRAÇÃO CONTÍNUA QUESTÕES

Prof. Raphael Lacerda

Analista/MPC-CA/2019

Assinale a opção que apresenta uma das principais características que diferencia o DevOps (desenvolvimento e operações) dos demais métodos da engenharia de software.

A Atribuir as atividades inerentes à equipe de desenvolvimento de software e as atividades da equipe de operações, para que não haja sobreposição de responsabilidades.

B Apresentar processo sistematizado de tarefas sequenciais adotadas para alinhar o que é feito isoladamente pela equipe de desenvolvimento, de modo a garantir que, ao sair da fábrica, o produto esteja de acordo com os artefatos e requisitos solicitados, no início do projeto, pela equipe de produção.

Analista/MPC-CA/2019

- C Ter como referência o conceito de entrega contínua ou CD (continuous delivery), realizada por um processo integrado e automatizado de desenvolvimento, teste de software e operações.
- D Trabalhar, primeiramente, a análise e a validação da documentação de software para, depois, planejar a viabilidade de automação de uma das fases do processo de desenvolvimento.
- E Ter sido o primeiro método a cunhar o termo “integração contínua”, que permite que todas as áreas envolvidas no desenvolvimento de software participem, por meio de reuniões rápidas, do planejamento, desenvolvimento e teste de software.

SLU-DF/2019

Na prática de integração contínua para desenvolvimento de software, vários colaboradores criam e mantêm o código de forma organizada e controlada, utilizando ferramentas como Git (controle de versão), Junit (testes), Hudson e Jenkins (deploys em ambientes de desenvolvimento e produção), o que reduz a geração de erros de integração.

TRF4/2019

Para escrever os testes de aceitação, podemos usar os critérios de aceitação descritos nas histórias de usuário. Estando estes testes implementados, executando e passando com sucesso, obter-se-á uma boa cobertura e garantia de que pelo menos as funcionalidades mais importantes estão sendo de fato cobertas pelos testes. É recomendável que esses testes sejam executados por um tipo de ferramenta (como: Jenkins, TravisCI, Hudson, CruiseControl etc.), que fará o trabalho de executar os testes automatizados de acordo com alguma estratégia predefinida (como: todo dia, a cada modificação feita no código-fonte etc.).

TRF4/2019

O Analista conclui que o tipo de ferramenta mencionada no texto é de

- a) teste unitário.
- b) integração contínua.
- c) teste de usabilidade.
- d) teste cross-platform.
- e) DDT- Database Driven Testing.

UFC/2019

Sobre Jenkins, assinale a alternativa correta.

A Jenkins é uma ferramenta de testes unitários.

B Jenkins é uma IDE de desenvolvimento Java.

C Jenkins é uma ferramenta de automação de compilação.

D Jenkins é uma ferramenta de integração contínua e entrega contínua.

E Jenkins possui suporte apenas aos sistemas de controle de versão SVN, Git e Mercurial.

UFPE/2019

No contexto de integração contínua (IC), assinale a alternativa incorreta.

A

Um dos princípios centrais da IC é integrar cedo e de forma frequente. Os desenvolvedores devem efetuar commits do código com frequência para obter os benefícios da IC.

B

O uso de IC em um projeto reduz riscos, reduz processos manuais repetitivos e permite gerar a qualquer momento uma versão do software pronta para deploy.

UFPE/2019

C Jenkins, Travis e Gitlab são exemplos de ferramentas que dão suporte à IC em projetos de software.

D A IC não é apenas uma implementação técnica, ela também é a implementação cultural e organizacional.

E O uso de testes automatizados é fundamental, pois a automação em um servidor de IC tira a responsabilidade dos desenvolvedores verificarem se há erros no código que escrevem para o projeto e que enviam através de commits para o repositório.

STJ/2018

Em um serviço de integração contínua, testes de unidade são executados automaticamente com a finalidade de detectar erros funcionais.

UFRN/2018

Uma das principais atividades de gerenciamento de configuração de software é o gerenciamento de builds. Sobre essa temática, analise as afirmativas abaixo.

- I Ferramentas de build realizam o controle de versão do código, rotulando as diferentes versões dos arquivos.
- II Ferramentas como Maven e Docker têm o mesmo propósito e a mesma forma de funcionamento.
- III TravisCI e CircleCI são exemplos de serviços online de geração de builds que possuem máquinas virtuais próprias para executar esse processo.

UFRN/2018

IV Uma build pode ser gerada automaticamente após um commit realizado em um repositório remoto.

Estão corretas as afirmativas

- a) I e IV.
- b) I e II.
- c) II e III.
- d) III e IV.

MPE-PI/2018

A infraestrutura como código é uma prática DevOps caracterizada pela infraestrutura provisionada e gerenciada por meio de técnicas de desenvolvimento de código e de software, como, por exemplo, controle de versão e integração contínua.

DPE-AM/2018

Na integração contínua, todo o código desenvolvido pelo time é incorporado em um repositório comum várias vezes ao dia. Isso garante que qualquer problema de integração ao longo do projeto possa ser notado e corrigido rapidamente.

COMPERVE/2018

o Scrum recomenda a adoção das práticas de testes automatizados e integração contínua.

BANRISUL/2018

Suponha uma organização que já está em processo adiantado de práticas ágeis de teste. Ela já tem bem consolidada a prática de criação de testes automatizados para as funcionalidades desenvolvidas, e um processo de integração contínua que inclui a verificação de testes de regressão durante a construção do build. Neste contexto, analise as afirmativas abaixo.

BANRISUL/2018

I - Mesmo que os desenvolvedores se assegurem de que todos os testes sejam executados com sucesso antes de integrar seu código (check-in), ainda assim é possível que testes de regressão falhem quando o novo código for integrado, causando uma “quebra de build”. O desenvolvedor responsável por esse novo código deve consertar o problema para que o build possa ser refeito, avaliando a lista de tarefas que tem a executar, a fim de definir a prioridade desse conserto, ao longo de seu dia de trabalho.

BANRISUL/2018

II - Para assegurar que um build possa dar retorno imediato, é necessário que possa ser gerado em um tempo curto. Se alguns testes de regressão são muito demorados (por exemplo, scripts de interface), podem-se usar estratégias, como ter um processo de build separado que rode os testes mais demorados, ou versões diferentes de construção completa e resumida de build que rodariam durante a noite ou durante as horas de trabalho, respectivamente.

III - Ao longo de uma iteração, novos testes automatizados são construídos. Todos os novos testes automatizados construídos para uma dada história devem ser incluídos na respectiva suíte de regressão.

Embasa/2017

Integração Contínua é um dos pilares das práticas ágeis. Um servidor de Integração Contínua open source, escrito em Java, é o:

- A JServer
- B Integration
- C Symphony
- D Jenkins

TRE-TO/2017

Assinale a opção que apresenta a atividade que consiste na entrega de cópias de uma aplicação para outra(s) máquina(s) a fim de que a aplicação seja executada em um novo ambiente.

- A validação
- B compactação
- C deploy
- D teste
- E Becake

TRE-RS/2015

Quanto às técnicas de entrega contínua e de integração contínua, é correto afirmar que

A o objetivo da integração contínua é manter o software em um estado funcional ininterrupto.

B é evitada a compilação do código da aplicação a cada mudança realizada, na integração contínua.

<http://stackoverflow.com/questions/28608015/continuous-integration-vs-continuous-delivery-vs-continuous-deployment>

TRE-RS/2015

C as duas técnicas são independentes, embora a técnica de integração contínua preceda a de entrega contínua.

D a integração contínua, caso o processo de compilação esteja quebrado, logo identifica a causa do problema e a corrige.

E é recomendável um sistema de controle de versão centralizado, quando se implementa a integração contínua.

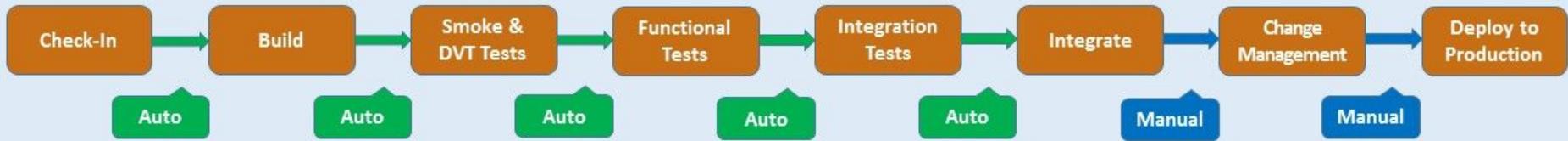
<http://stackoverflow.com/questions/28608015/continuous-integration-vs-continuous-delivery-vs-continuous-deployment>

Continuous Model

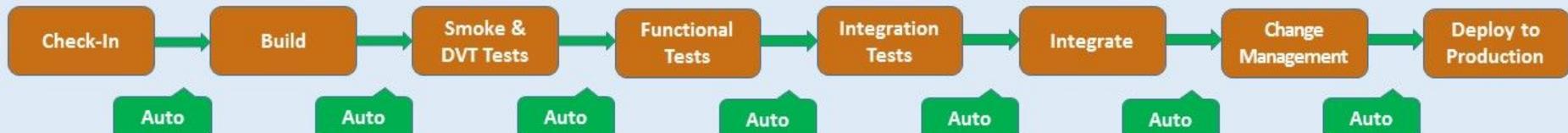
Continuous Integration



Continuous Delivery



Continuous Deployment



TCU/2015

Na gerência de um pipeline de implantação (deployment pipeline), recomenda-se que o código-fonte seja compilado repetidas vezes em contextos diferentes: durante o estágio de commit, nos testes de aceitação, nos testes de capacidade e nos testes exploratórios.

TCU/2015

Para que a prática de integração contínua seja eficiente, é necessário parametrizar e automatizar várias atividades relativas à gerência da configuração, não somente do código-fonte produzido, mas também de bibliotecas e componentes externos.

STJ/2015

A finalidade das ferramentas de integração contínua é a criação de soluções integradas com foco em sistemas fortemente acoplados, com necessidade de criação de uma documentação contínua.

ESAF/2015

Entre as melhores práticas da Integração Contínua citadas por Martin Fowler, no seu famoso artigo "Continuous Integration", não se encontra.

- A automatize o processo de Build.
- B cada modificação salva deve gerar um Build automaticamente.
- C execute os testes finais no ambiente de produção.
- D garanta que o Build seja rápido.
- E mantenha um repositório de fontes unificado.

TRE-MA/2015

No desenvolvimento de software em XP, são empregadas algumas práticas. Avalie as assertivas abaixo.

- I. Programação em pares.
- II. Time coeso.
- III. Integração contínua.
- IV. Desenvolvimento orientado a testes.

Quantas afirmativas são verdadeiras?

ANATEL/2014

A integração contínua constitui um dos pilares das práticas ágeis: a constante integração das respectivas partes deve ocorrer diariamente e ser realizada concomitantemente à atividade de programação em pares.

ANATEL/2014

Por não existir relação direta entre as práticas ágeis de integração contínua e de entrega contínua, pode-se garantir a agilidade do processo pela realização de tais atividades em paralelo, ainda que relacionadas à mesma versão.

STF/2013

São características de um servidor de integração contínua: compilação do código-fonte, realização de operações de checkout do gerenciador de configuração, execução de um conjunto de testes, auditoria do código-fonte e testes funcionais.

EBSERH/2013

Controle de versão: identifica em qual versão está o software e quais as características de cada versão.

Controle de mudanças: identifica quais foram as mudanças, efetuadas na versão.

Integração contínua: tem como características testar as mudanças, assim que são realizadas.

STF/2013

Quanto à gestão ágil de projetos com Scrum e às noções gerais de DevOps, julgue os itens subsecutivos.

Integração contínua, entrega contínua, teste contínuo, monitoramento contínuo e feedback são algumas práticas d

ANAC/2012

A integração contínua é uma técnica de alto impacto no levantamento de requisitos.

TRE-RN/2011

Considere as seguintes características:

- I. Propriedade coletiva.
- II. Integração contínua.
- III. Metáfora.

Dentre as práticas componentes da Extreme Programming, aplica-se o que consta em

TJ-PI/2011

XP (eXtreme Programming) é uma metodologia ágil para equipes pequenas e médias que desenvolvem software com requisitos vagos e em constante mudança. Para isso, adota a estratégia de constante acompanhamento e realização de vários pequenos ajustes durante o desenvolvimento de software. Para aplicar os valores e princípios durante o desenvolvimento de software, a XP propõe uma série de práticas, sendo uma delas: sempre que produzir uma nova funcionalidade, nunca esperar uma semana para integrar à versão atual do sistema a fim de evitar o aumento da possibilidade de conflitos e da possibilidade de erros no código fonte. Tal prática é denominada

- a) Time Coeso.
- b) Refatoração.
- c) Integração Contínua.
- d) Desenvolvimento Orientado a Testes.
- e) Ritmo Sustentável.

Petro/2010

É comum, na Engenharia de Software, o uso de ferramentas de software que auxiliam na realização de diversas atividades do desenvolvimento. Nesse contexto, ferramentas de

- a) refatoração auxiliam a modificação de uma região do código-fonte de um sistema, com o objetivo de alterar comportamento externo deste último.
- b) análise estática do código permitem obter métricas de qualidade de um produto de software, tais como o grau de dependência entre seus componentes.

Petro/2010

- c) integração contínua são destinadas a automatizar a implantação do produto de software no ambiente de produção.
- d) engenharia reversa são destinadas ao controle das diversas versões de um produto de software.
- e) cobertura de código determinam em que grau o código-fonte de um sistema corresponde aos seus requisitos não funcionais.



TÁ PRONTO!
SÓ FALTA
ALINHAR DIVS



Prof. Raphael Lacerda



DevOps DENTRO DO "DEV AO OPS"

Prof. Raphael Lacerda



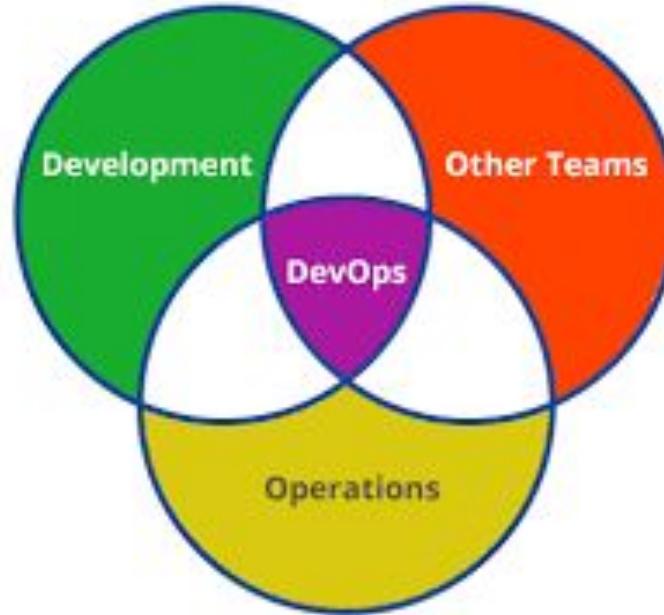
RUMO À AUTOMATIZAÇÃO



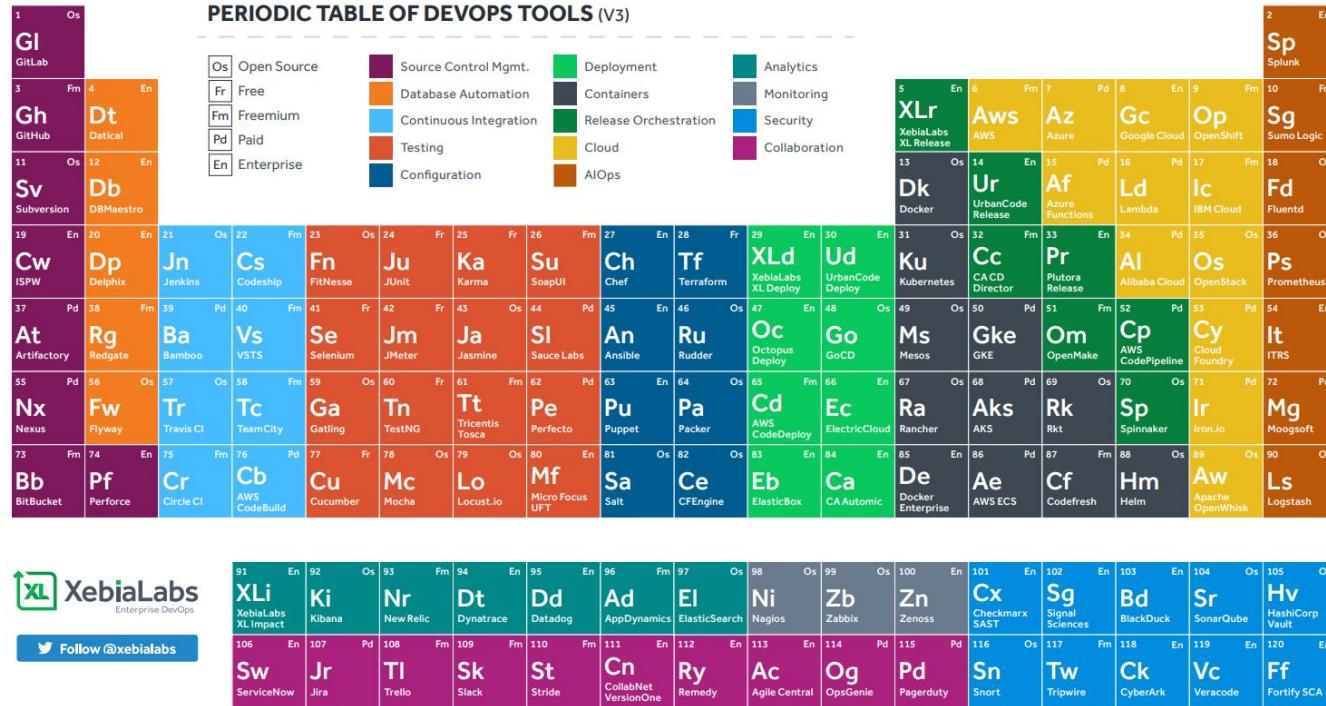
Prof. Raphael Lacerda

ONCE UPON A TIME

Your company in a nutshell



Welcome to your future



<https://xebialabs.com/wp-content/uploads/files/infographics/periodic-table-of-devops-tools-v3-1.pdf>

<https://xebialabs.com/periodic-table-of-devops-tools/>

07074192716 - Jefferson Terra

então devops



é basicamente ferramentas?

no, no, no, nooooo!

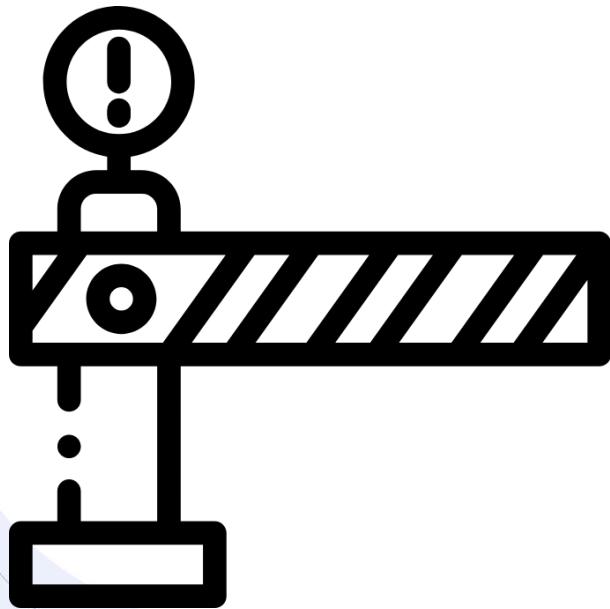
ThoughtWorks®

BUILD YOUR
OWN RADAR



<https://www.thoughtworks.com/radar/techniques>

Como fomentar a cultura?

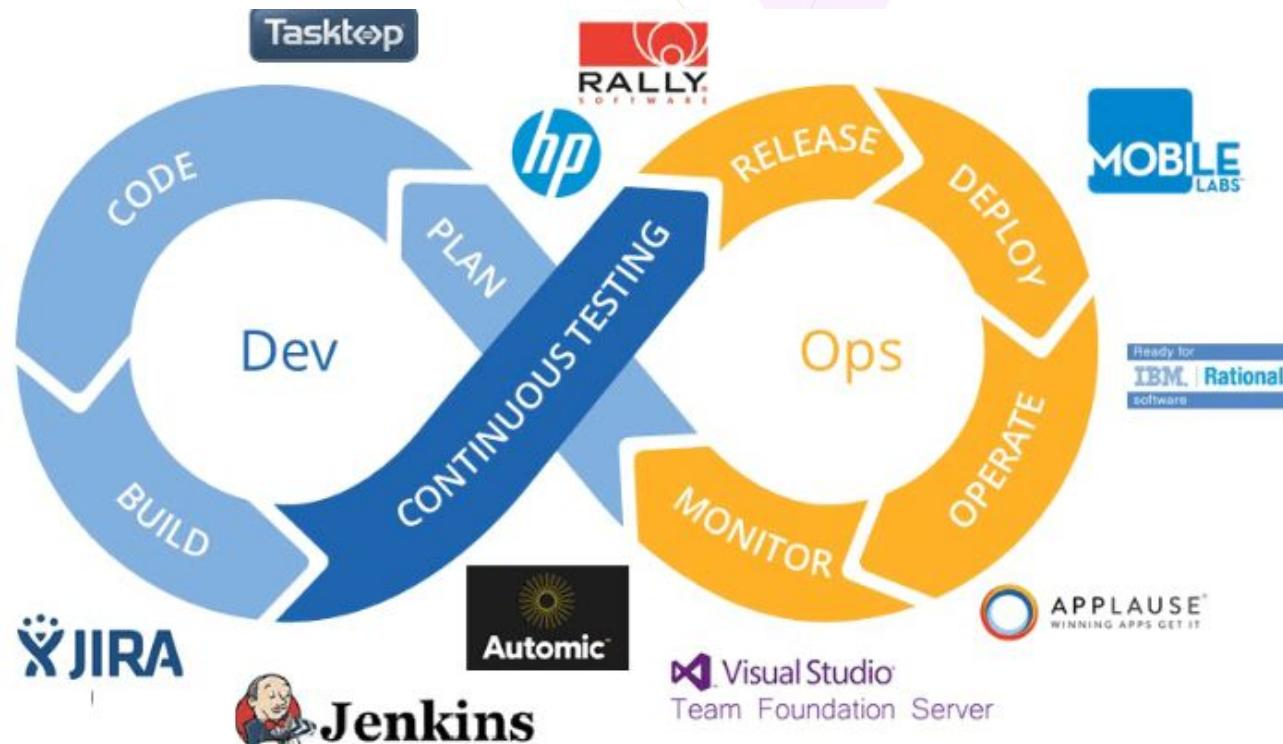


<https://aws.amazon.com/pt/devops/what-is-devops/>

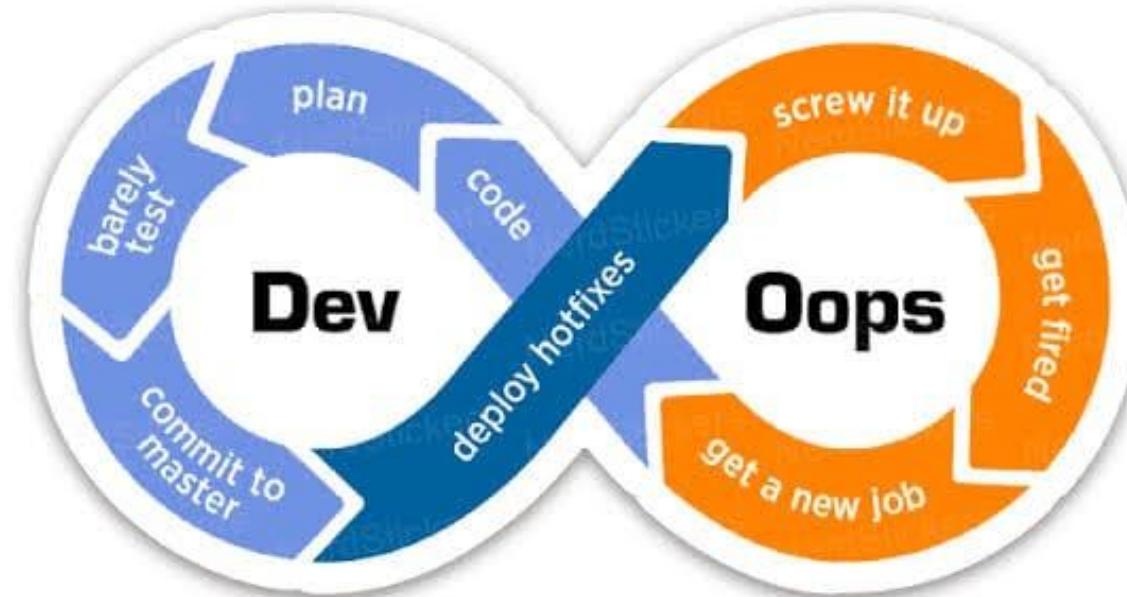
<https://techbeacon.com/devops/2015-state-devops-report-top-10-takeaways-insights>

<https://martinfowler.com/bliki/DevOpsCulture.html>

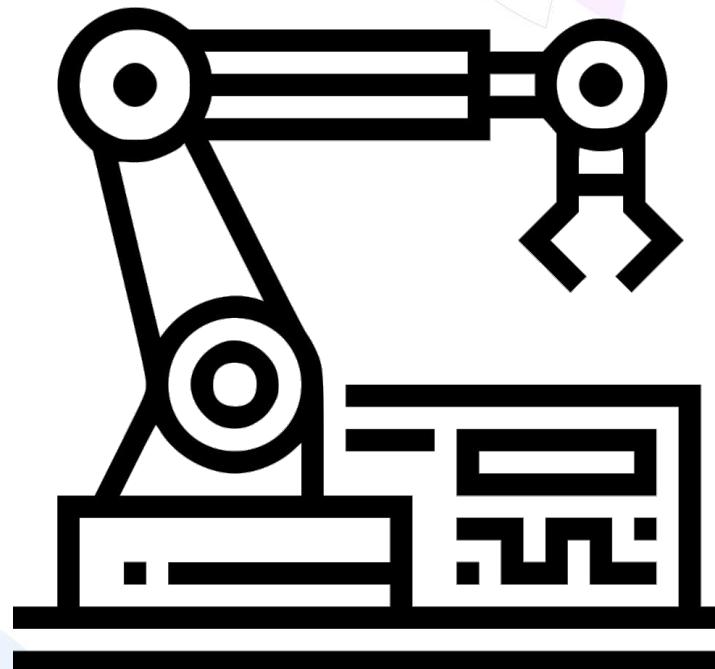
Expectativa



Realidade



mas o que automatizar?



Infra as Code



TERRAFORM

<https://stackshare.io/stackups/ansible-vs-terraform>



SALTSTACK

code na infra

```
1 ----
2 ▼ - hosts: webservers
3   sudo: yes
4
5 ▼ vars:
6   app_name: PleaseDeployMe
7   repo_url: https://github.com/username/repo_name.git
8   repo_remote: origin
9   repo_version: master
10  webapps_dir: /deployed
11  virtualenv_root: /deployed/PleaseDeployMe/mac
12 ▼ tasks:
13
14 ▼   - name: git pull project
15     git: repo={{repo_url}} dest={{webapps_dir}}/{{app_name}} version=master
16
17     notify:
18       - restart app
19
20     - name: install things
21     pip: name=virtualenv
22
23     - name: create virtualenv
24     command: virtualenv /deployed/PleaseDeployMe/venv
25
26     - name: activate virtualenv
27     command: /bin/bash /deployed/PleaseDeployMe/venv/bin/activate
28
29     - pip: requirements=/deployed/{{app_name}}/requirements.txt virtualenv=/deployed/{{app_name}}/mac
30
31     - name: run supervisord
32     command: "supervisord -c /deployed/PleaseDeployMe/supervisord.conf"
33
34     - name: begin flask app
35     supervisorctl: name=flask_app state=started
36
37
38 ▼ handlers:
39 ▼   - name: restart app
40     supervisorctl: name={{app_name}} state=restarted
41
42 07074192716 - Jefferson Terra
```

code na infra

```
variable "base_network_cidr" {
  default = "10.0.0.0/8"
}

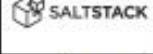
resource "google_compute_network" "example" {
  name          = "test-network"
  auto_create_subnetworks = false
}

resource "google_compute_subnetwork" "example" {
  count = 4

  name          = "test-subnetwork"
  ip_cidr_range = cidrsubnet(var.base_network_cidr, 4, count.index)
  region        = "us-central1"
  network       = google_compute_network.custom-test.self_link
}
```

Infra as Code

The Major (cross-platform) Players

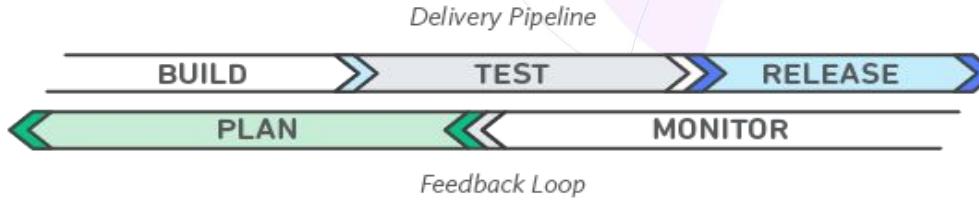
Tool	Tool Type	Infrastructure	Architecture	Approach	Manifest Written Language
 puppet	Configuration Management	Mutable	Pull	Declarative	Domain Specific Language (DSL) & Embedded Ruby (ERB)
 CHEF	Configuration Management	Mutable	Pull	Declarative & Imperative	Ruby
 ANSIBLE	Configuration Management	Mutable	Push	Declarative & Imperative	YAML
 SALTSTACK	Configuration Management	Mutable	Push & Pull	Declarative & Imperative	YAML
 Terraform	Provisioning	Immutable	Push	Declarative	HashiCorp Configuration Language (HCL)



monitorar!



YOUR COMPANY



CUSTOMERS

**YOU CAN'T
IMPROVE
WHAT YOU
DON'T
MEASURE.**

1,000s de mudanças.

por que monitorar?

100,000,000,000s de eventos.

10,000,000,000s de requests.

1,000,000,000s de métricas.

1,000,000s de requests / segundo.

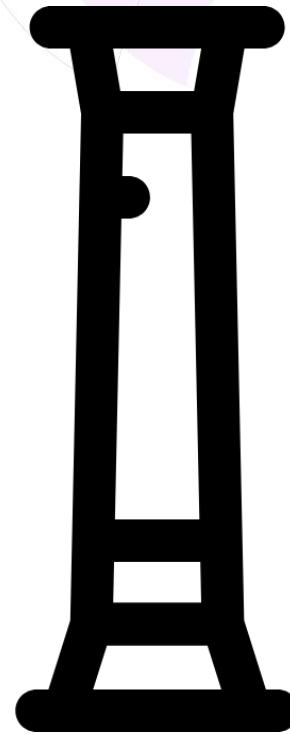
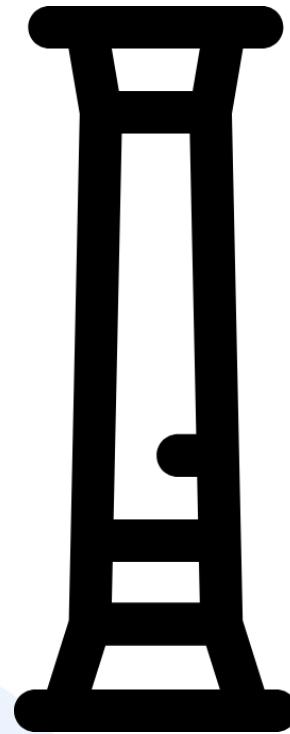
10s de terabits / seg.

O que querem
Quando querem
Dispositivo que querem

100,000,000s de horas de streaming.

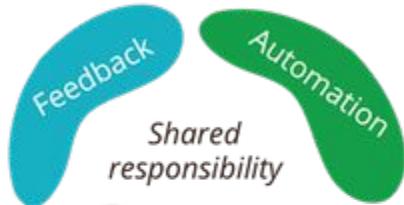
10,000,000s de dispositivos.

pilares



benefícios

Dev & Ops Collaboration

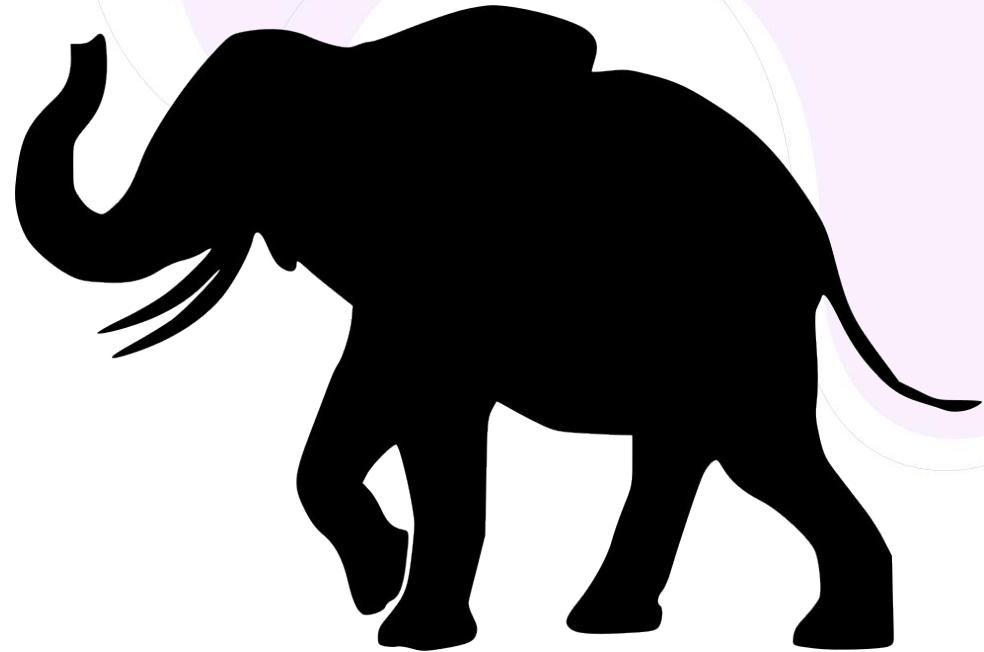


No silos

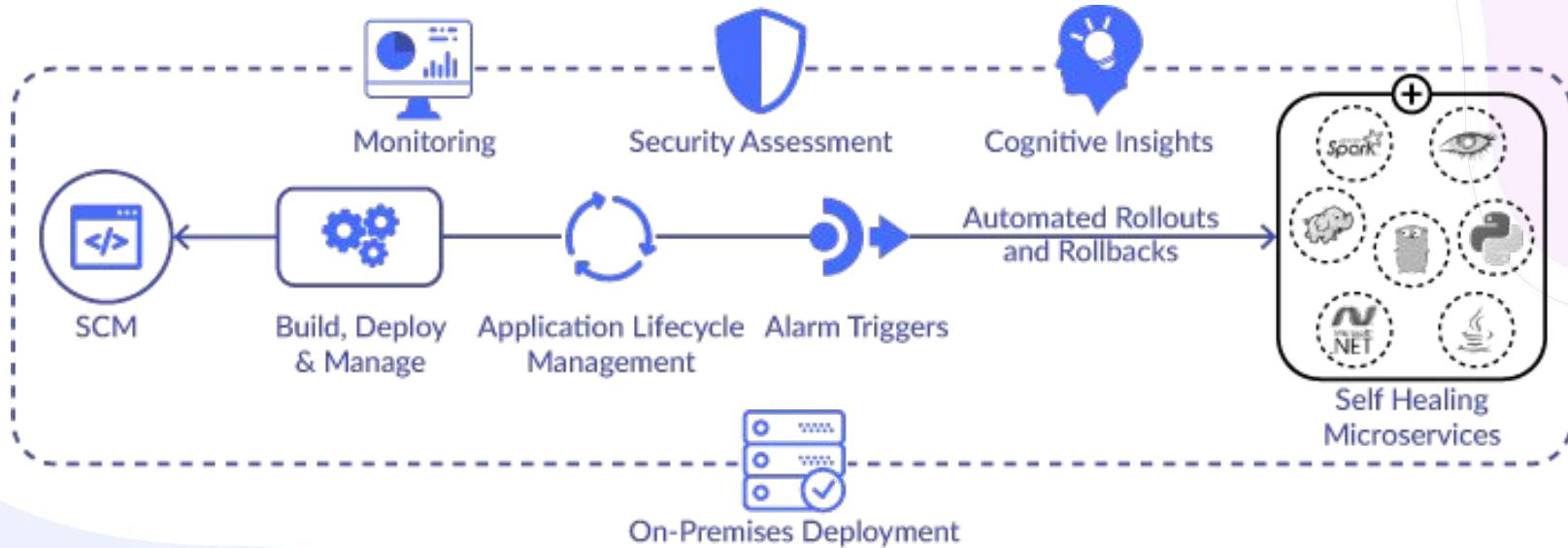
Autonomous teams

Team culture

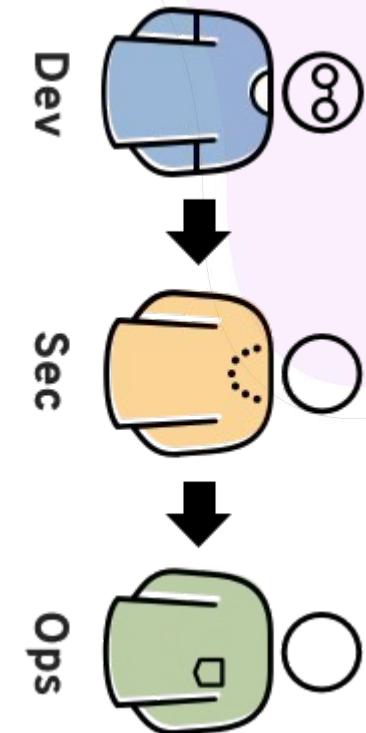
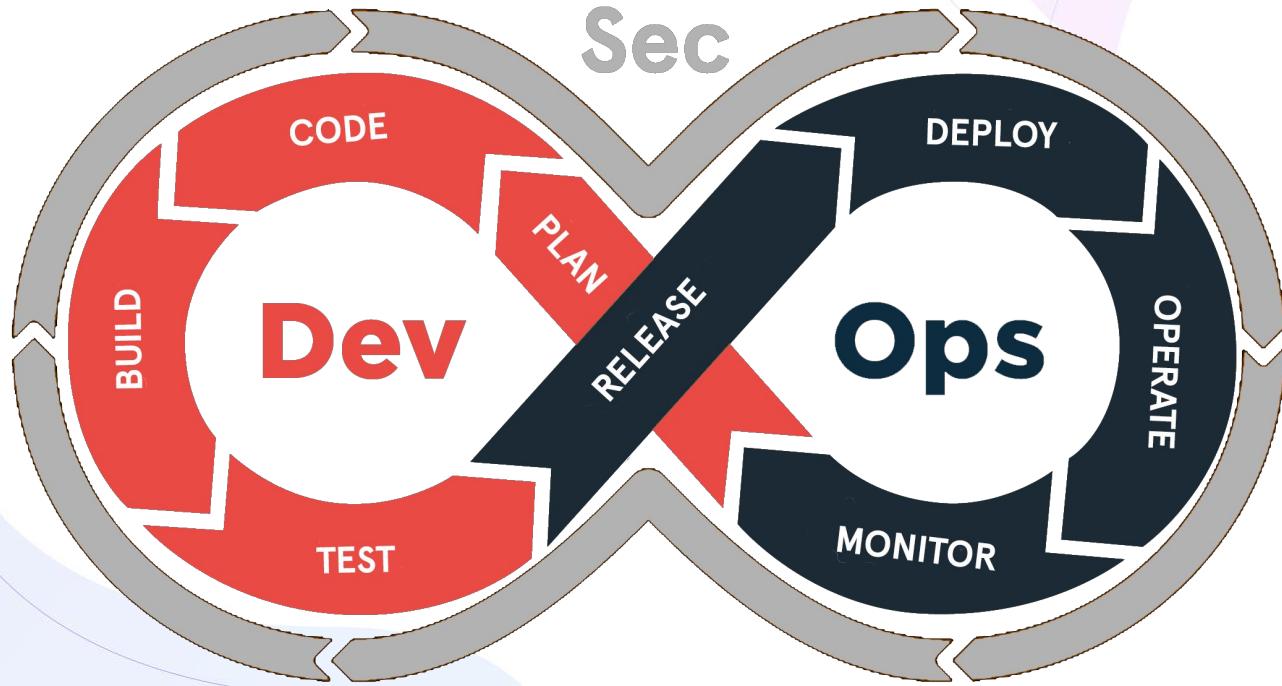
Organizational culture



um mundo ideal

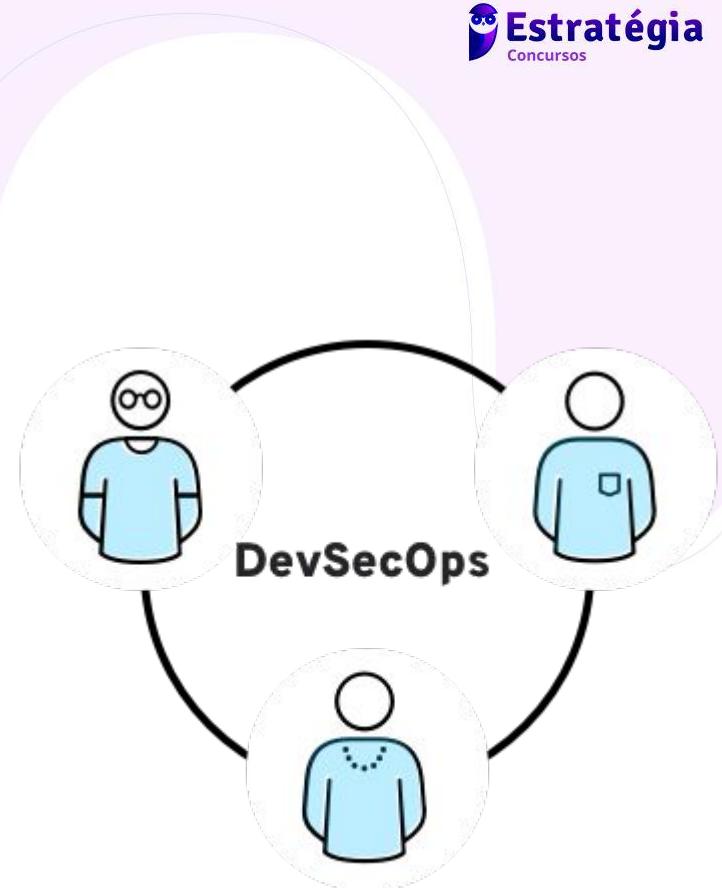


DevSecOps



tudo junto e misturado

DevOps **VS** DevSecOps



automatize tudo!



DevSecOps



OWASP
Open Web Application
Security Project

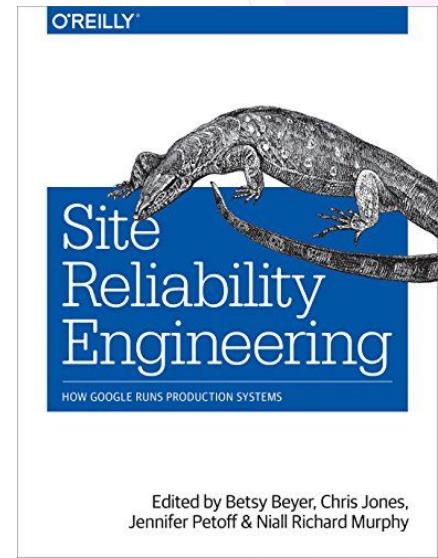
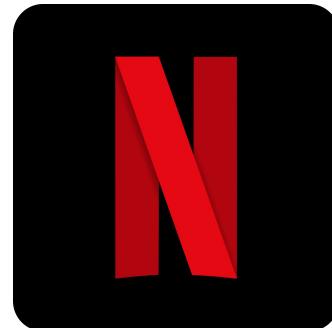
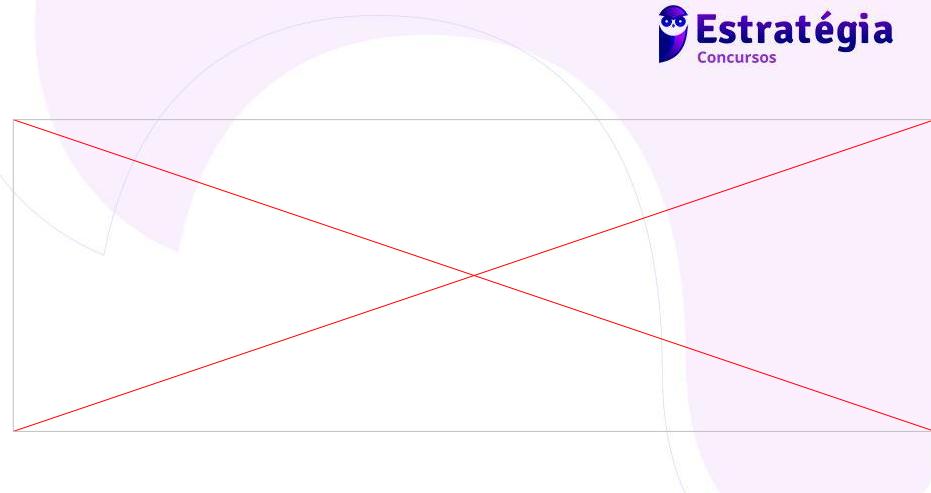
<https://www.sonarqube.org/features/security/>



DevSecOps

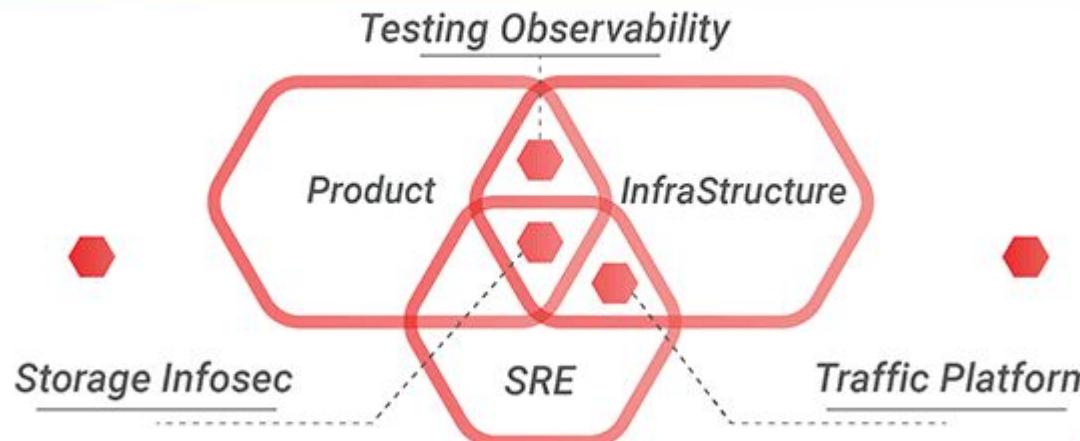


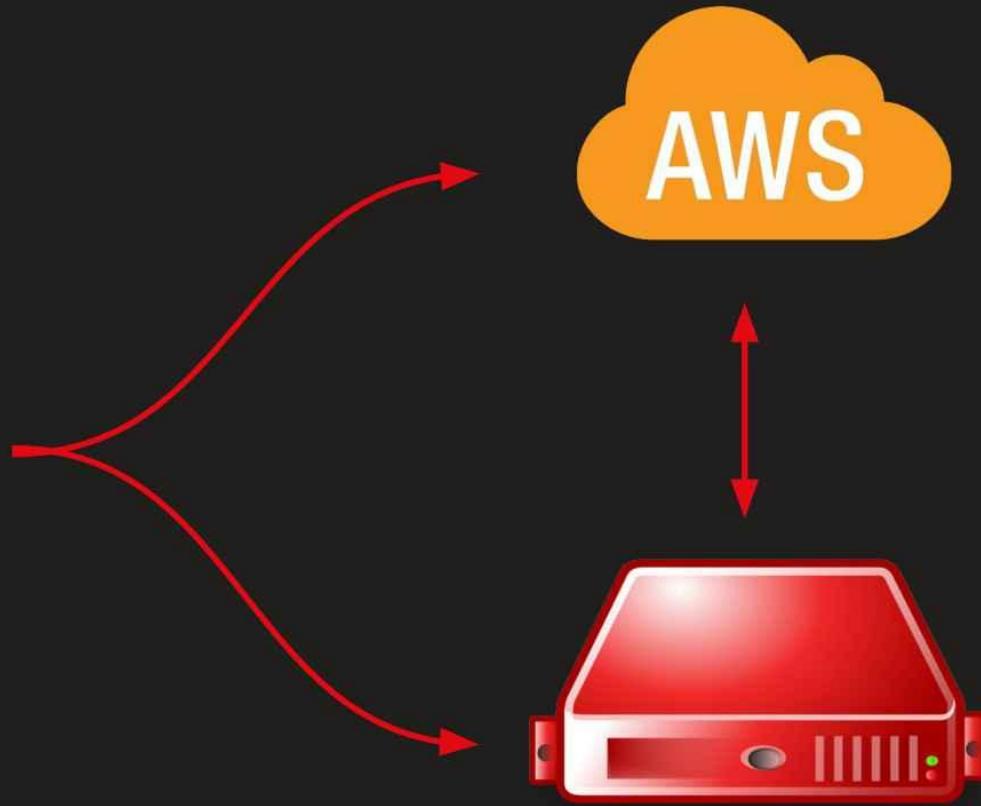
SRE



Quem faz SRE?

— Site Reliability Engineering (SRE) —





NETFLIX



Por que?

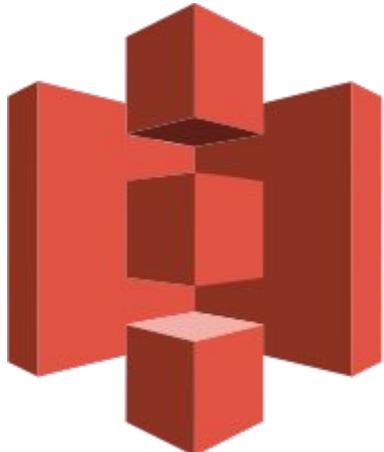
NARCOS

A 2014 report shows that the cost of one hour of downtime for Netflix is \$200,000. It's likely that amount has grown since then.



<https://www.networkworld.com/article/3178076/why-netflix-didnt-sink-when-amazon-s3-went-down.html>

cloud, or not



amazon S3

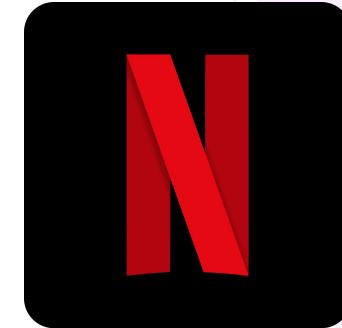
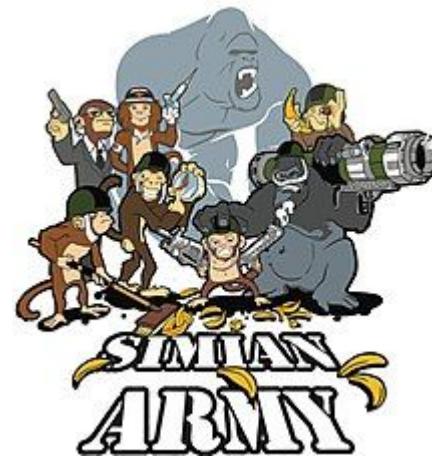
Anything that can go wrong, will go wrong.

<https://aws.amazon.com/pt/message/41926/>





Chaos!



<https://netflix.github.io/>

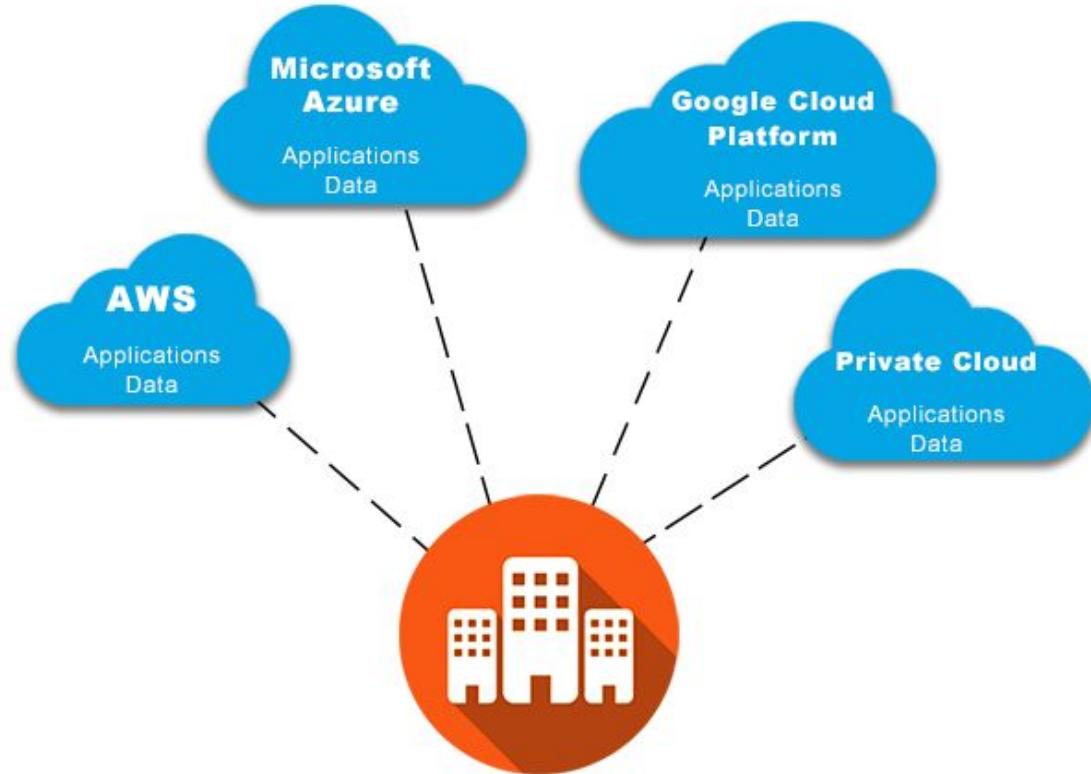


<https://www.gremlin.com/chaos-monkey/>

<https://github.com/Netflix/chaosmonkey>

Engenharia do CAOS!

multicloud





DevOps - QUESTÕES

Prof. Raphael Lacerda

Analista/MPC-CA/2019

Assinale a opção que apresenta uma das principais características que diferencia o DevOps (desenvolvimento e operações) dos demais métodos da engenharia de software.

A Atribuir as atividades inerentes à equipe de desenvolvimento de software e as atividades da equipe de operações, para que não haja sobreposição de responsabilidades.

B Apresentar processo sistematizado de tarefas sequenciais adotadas para alinhar o que é feito isoladamente pela equipe de desenvolvimento, de modo a garantir que, ao sair da fábrica, o produto esteja de acordo com os artefatos e requisitos solicitados, no início do projeto, pela equipe de produção.

Analista/MPC-CA/2019

- C Ter como referência o conceito de entrega contínua ou CD (continuous delivery), realizada por um processo integrado e automatizado de desenvolvimento, teste de software e operações.
- D Trabalhar, primeiramente, a análise e a validação da documentação de software para, depois, planejar a viabilidade de automação de uma das fases do processo de desenvolvimento.
- E Ter sido o primeiro método a cunhar o termo “integração contínua”, que permite que todas as áreas envolvidas no desenvolvimento de software participem, por meio de reuniões rápidas, do planejamento, desenvolvimento e teste de software.

MPE-PI/2018

A infraestrutura como código é uma prática DevOps caracterizada pela infraestrutura provisionada e gerenciada por meio de técnicas de desenvolvimento de código e de software, como, por exemplo, controle de versão e integração contínua.

BNB/2018

DevOPS é um conjunto de ferramentas e práticas de trabalho para integração entre os colaboradores dos grupos de desenvolvimento de código, de operações e de apoio, e pode ser utilizado na produção rápida e segura de aplicações e serviços.

EMAP/2018

O comando a seguir é capaz de executar o comando ping em todos os hosts cadastrados para o Ansible.

```
ansible full -x ping
```

BNB/2018

- I. Colaboração e desenvolvimento de empatia entre integrantes das equipes com foco no projeto e não em interesses pessoais.
- II . Reuniões com a participação de profissionais de várias áreas necessárias para o projeto (desenvolvimento, operações, apoio).
- III . Utilização de metodologias ágeis como RUP, XP e/ou Scrum para permitir entregas rápidas e contínuas.

BNB/2018

- IV. Implementação do gerenciamento de configuração para que mudanças realizadas manualmente nos servidores, sem conhecimento da gerência de configurações, sejam desfeitas.
- V. Estratégias para gestão de incidentes bem definidas, políticas de rollback, backup e ferramentas de monitoração proativas.
- VI. Ambientes necessários para o trabalho da equipe de desenvolvimento providos de forma dinâmica e automatizada, sem necessidade de intervenção da equipe de operações.

BNB/2018

São características corretas e alinhadas às práticas DevOps APENAS os itens

- a) II, IV e VI .
- b) I, III e V.
- c) I, II , IV, V e VI .
- d) III, IV, V e VI .
- e) I, II e III .

TJ-MS/2017

Sistemas virtualizados e containers são conceitos importantes para computação na nuvem. Para gerenciar grande número de servidores físicos, virtualizados e containers, utilizam-se ferramentas especializadas de configuração remota. Indique a afirmativa que descreve de forma CORRETA os conceitos relativos a sistemas virtualizados e containers e as ferramentas de gerenciamento disponíveis.

A Puppet e Ansible são ferramentas que tem a finalidade de simplificar o processo de gerenciamento de servidores remotos. Essas ferramentas funcionam apenas com servidores físicos ou virtualizados. Elas não suportam containers.

TJ-MS/2017

B

Puppet e Ansible podem ser usados para gerenciar serviços virtualizados. Ansible é preferível por ser uma ferramenta multi-plataforma, enquanto Puppet funciona apenas para Linux, pois todos os seus comandos remotos são executados via SSH.

C

Containers e máquinas virtuais são sinônimos, pois ambos são usados para virtualizar o hardware que hospeda um sistema operacional completo, que pode ser diferente do sistema operacional da máquina física.

TJ-MS/2017

D

Containers do tipo Dockers podem ser orquestrados apenas pelo Docker Swarm, que foi desenvolvido especificamente para suportar essa tecnologia de container.

E

Kubernetes é uma ferramenta de orquestração que oferece recursos de gerenciamento para containers, como balanceamento de carga e migração sem perda de dados.

TRT-8/2016

Acerca de DevOps, assinale a opção correta.

- a) O DevOps concentra-se em reunir diferentes processos e executá-los mais rapidamente e com mais frequência, o que gera baixa colaboração entre equipes.
- b) O DevOps tem como princípio produzir, a partir da avaliação dos times de desenvolvimento do serviço, grandes mudanças e farta documentação com valor agregado para os usuários, assemelhando-se, por isso, com objetivos dos métodos iterativos e em cascata.

TRT-8/2016

- c) A infraestrutura de nuvem de provedores internos e externos vem restringindo o uso de DevOps pelas organizações.
- d) O DevOps parte da premissa de adoção de grandes equipes de especialistas, com a menor interação possível, visando à padronização de processos e à mínima automação de atividades.
- e) Atividades típicas em DevOps compreendem teste do código automatizado, automação de fluxos de trabalho e da infraestrutura e requerem ambientes de desenvolvimento e produção idênticos.

TCU/2015

De acordo com a abordagem DevOps (development – operations), os desafios da produção de software de qualidade devem ser vencidos com o envolvimento dos desenvolvedores na operação dos sistemas com os quais colaboraram no desenvolvimento.

STJ/2015

DevOps é um conceito pelo qual se busca entregar sistemas melhores, com menor custo, em menor tempo e com menor risco.

STJ/2015

O DevOps, movimento profissional emergente que defende uma colaboração maior entre desenvolvimento e operações de TI, resulta em um fluxo rápido do trabalho planejado, que aumenta a confiabilidade, a estabilidade e a segurança do ambiente de produção.

STJ/2015

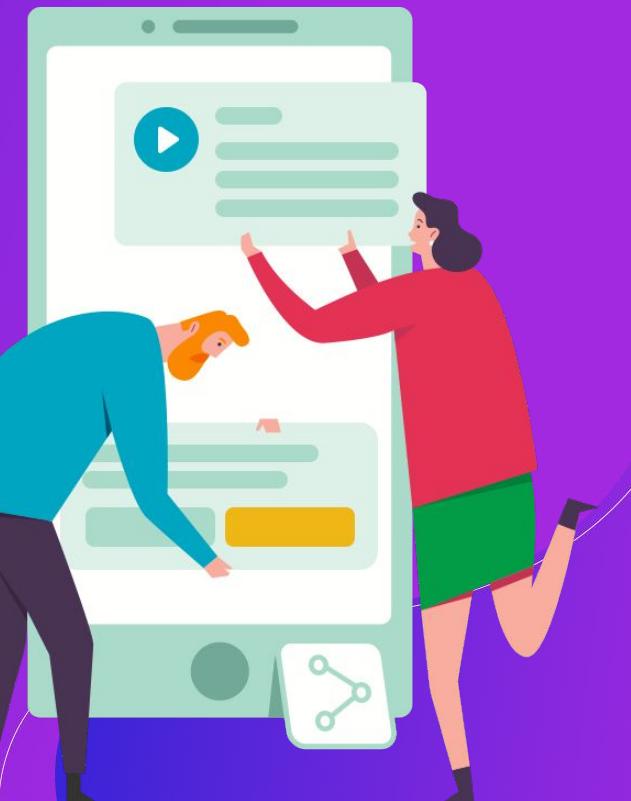
O profissional especialista em DevOps deve atuar e conhecer as áreas de desenvolvimento (engenharia de software), operações e controle de qualidade, além de conhecer, também, de forma ampla, os processos de desenvolvimento ágil.

STF/2013

Teste contínuo é uma prática do DevOps que, além de permitir a diminuição dos custos finais do teste, ajuda as equipes de desenvolvimento a balancear qualidade e velocidade.



DEVOPS TODOS JUNTOS!



Prof. Raphael Lacerda



OBRIGADO



Podcast Estratégia Tech



Prof. Raphael Lacerda



Estratégia

Concursos