



e-Book



EM 3 Fases

4DevOps é a unidade da 4Linux especializada em serviços de consultoria para o mundo DevOps e Cloud, priorizando o uso de software open source.



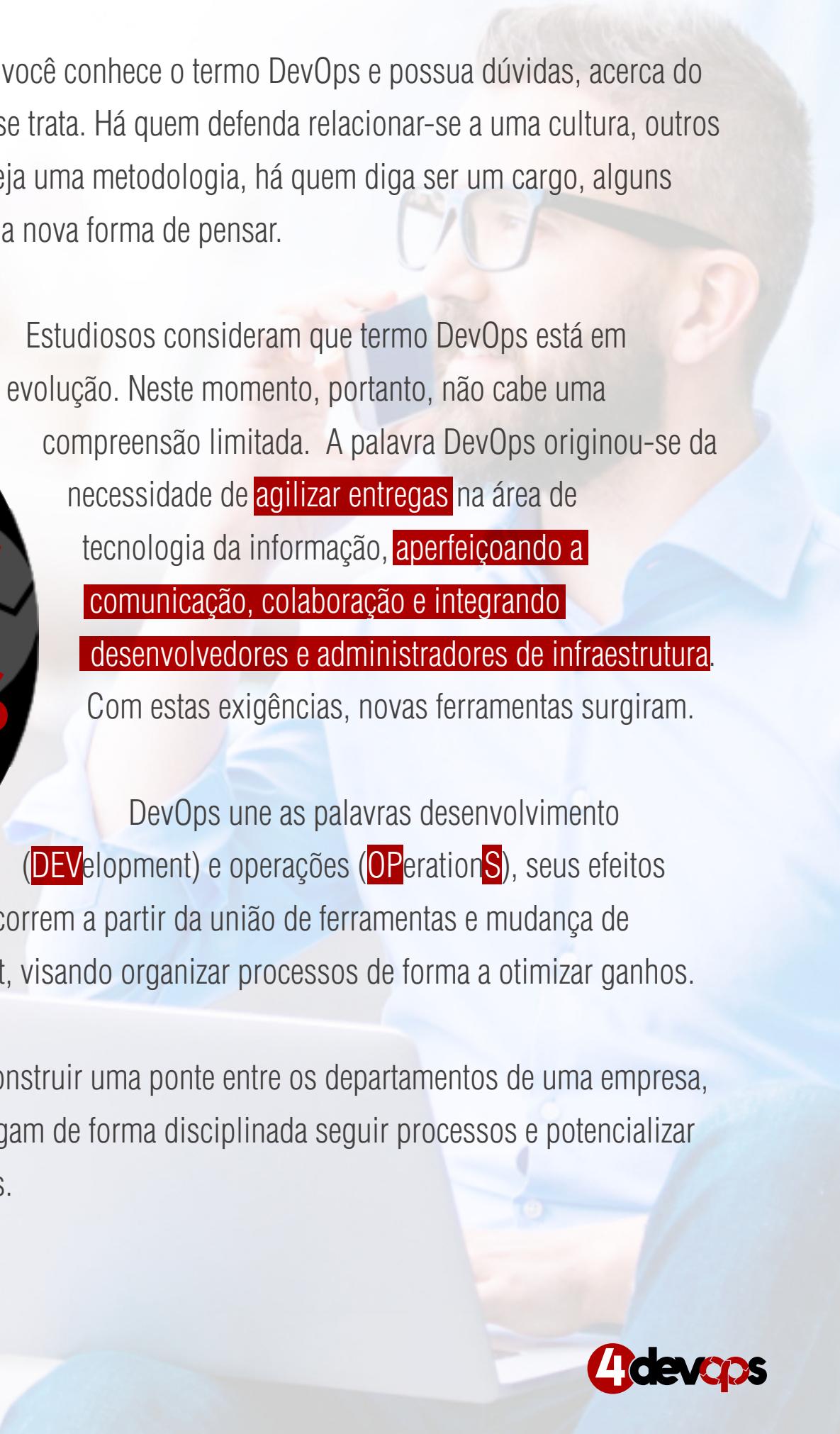
DevOps, assim como tecnologias evolutivas como Cloud, BigData etc, tornou-se a ‘buzzword’ do momento na área de T.I.

Entretanto, muitos interessados no assunto não encontram um caminho seguro ou, um parceiro confiável para fazer esta adoção.

A 4Linux implementou pipeline DevOps usando softwares livres e open source em vários clientes, seja fazendo ‘deploy’ ‘on premise’ ou ‘in cloud’. Também ministra o curso DevOps Master, preparatório para a certificação da EXIN.

Com base em nossa expertise, decidimos redigir este e-book para nortear quem pretende adotar a cultura DevOps em sua área de T.I.

Provavelmente você conhece o termo DevOps e possua dúvidas, acerca do que realmente se trata. Há quem defenda relacionar-se a uma cultura, outros apostam que seja uma metodologia, há quem diga ser um cargo, alguns veem como uma nova forma de pensar.

A large, semi-transparent circular graphic is positioned on the left side of the page. It contains the words "Dev" and "Ops" in red, with a grey gear-like icon between them.

Estudiosos consideram que termo DevOps está em evolução. Neste momento, portanto, não cabe uma compreensão limitada. A palavra DevOps originou-se da necessidade de **agilizar entregas** na área de tecnologia da informação, **aperfeiçoando a comunicação, colaboração e integrando desenvolvedores e administradores de infraestrutura**. Com estas exigências, novas ferramentas surgiram.

DevOps une as palavras desenvolvimento (**DEVelopment**) e operações (**OPERATIONS**), seus efeitos ocorrem a partir da união de ferramentas e mudança de mindset, visando organizar processos de forma a otimizar ganhos.

É necessário construir uma ponte entre os departamentos de uma empresa, para que consigam de forma disciplinada seguir processos e potencializar seus resultados.

Organizações afirmam que os benefícios da cultura DevOps são surpreendentes, podem causar revoluções em todo o processo de trabalho. De modo macro, contribui na alta produtividade, elevada qualidade na entrega de software e menor custo de produção.

Destacamos os benefícios da implementação do DevOps:

Melhoria na **eficiência** de negócio.

Ampliação da **satisfação** de clientes.

Prolongamento do tempo de **permanência da aplicação** no ar.

Incremento de **agilidade** do negócio.

Alta na conversão de clientes.

Aumento do **valor para o negócio**.

Redução de custos de T.I.

Elevação da **produtividade** e na **satisfação** dos funcionários.

Observa-se ainda que a capacidade de pesquisa e inovação da empresa é aprimorada, visto que os colaboradores são incentivados a aperfeiçoar os produtos e serviços, por vezes, saindo de sua zona de conforto na busca de soluções criativas.

ASPECTOS MOTIVACIONAIS PARA IMPLANTAÇÃO DE CULTURA DEVOPS

O primeiro passo para implementar DevOps em sua empresa, elencar as metas que pretende alcançar, contando com ferramentas ideais e eficazes para potencializar o desempenho.

Alguns aspectos merecem atenção durante este processo. Importante que toda a equipe conte com visualização direta e simplificada de produção, preparo e teste, acompanhando de maneira autônoma sua performance.



Aliás, o desempenho deve ser um requisito básico, fator crítico para o sucesso de seus produtos. Valorize-o, desenvolva os times de modo a compreenderem sua relevância. Trata-se de uma característica que deve acompanhar seu produto da ideia inicial até a entrega.

Só há uma maneira de garantir o desempenho total de seu produto: testando-o sempre. Crie um ambiente de teste contínuo, quanto antes receber feedbacks, mais fácil será corrigir erros e melhorar a qualidade.

Ao reforçar a cultura de metas fixas e concretas, se faz necessário partilhar junto aos seus colaboradores. Todos devem estar alinhados, no mesmo ritmo de produção, para unidos conquistarem o resultado esperado.

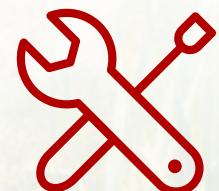
ASPECTOS MOTIVACIONAIS PARA IMPLANTAÇÃO DE CULTURA DEVOPS

O processo de produção deve ser monitorado, especialmente no que tange ao quesito desempenho. Em ambiente completamente automatizado, há cinco categorias que merecem atenção e exame:

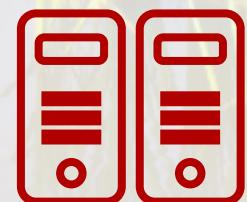


CICLOS DE DESENVOLVIMENTO: nesta etapa verificam-se detalhes técnicos. Por exemplo: quantos recursos serão incorporados, quantos erros foram identificados e corrigidos etc.

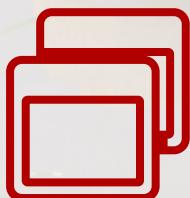
IMPLEMENTAÇÕES: quando o produto falha, é prioridade identificar a causa, por isso torna-se fundamental monitorar todo o percurso.



VULNERABILIDADE: conhecer as vulnerabilidades do seu produto permite que pense e estabeleça ações para solucioná-las e assim, entregar o melhor produto ao cliente.



SAÚDE DO SERVIDOR: há diversas formas de realizar este monitoramento, garantindo que a infraestrutura para receber seu produto esteja em pleno funcionamento.



DESEMPENHO DO APLICATIVO: novamente o desempenho do produto é foco. É necessário garantir que sua produção funcione da maneira esperada por você e, principalmente pelo cliente.

ASPECTOS MOTIVACIONAIS PARA IMPLEMENTAÇÃO DE CULTURA DEVOPS

Concentre-se em proporcionar a melhor experiência ao seu usuário e cliente. Após a implantação do DevOps, é preciso que experimentem melhora na qualidade do produto ou serviço.

A participação é fundamental por tratar-se de uma mudança de processo, não apenas a adoção de ferramentas.

Implementar a cultura DevOps é desafiador, sendo essencial contar com a mobilização de todos no seu negócio, do presidente ao responsável pela entrega.

Para garantir o sucesso em seu processo DevOps, indispensável conhecer muito bem a estratégia, sabendo exatamente como implementá-la.

Há empresas que podem auxiliá-lo neste processo, como é o caso da 4Linux que acumula experiências.

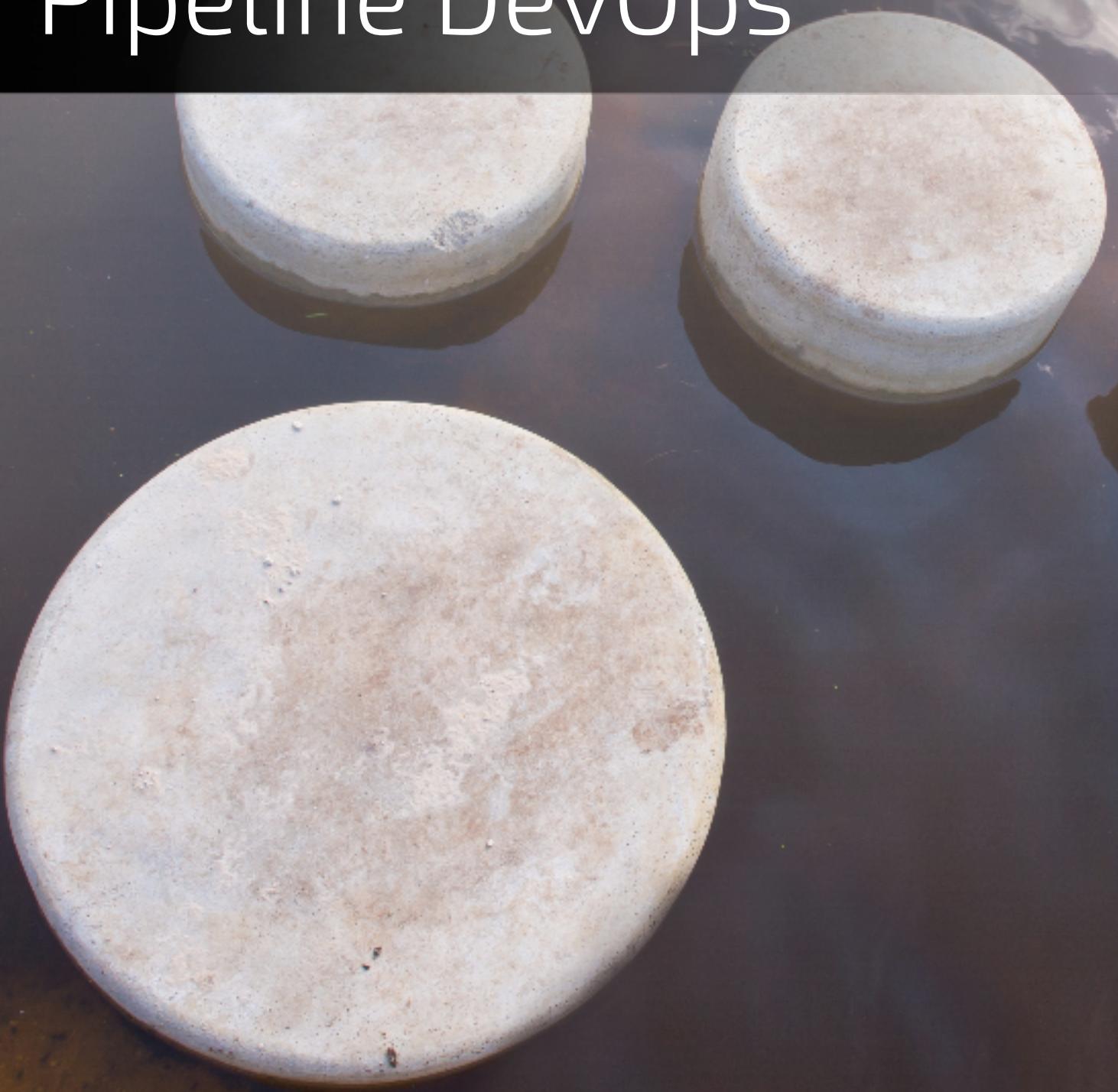
Transformação Digital, representa modificações pelas quais as empresas vêm passando, em seu jeito de fazer negócios, na maneira como operam em meio as mudanças diárias em tecnologia, frente a necessidade de atendimento ao consumidor digital com elevada agilidade.

Infraestruturas escaláveis em Cloud, BigData e Cultura DevOps constituem projetos realizados para suportar as mudanças, gracias a cultura de compartilhamento de código (open source), adotada por organizações como Google, Facebook, NetFlix entre outras.

A área de TI precisa estar preparada. Em nosso entendimento, o que distingue o DevOps, seus benefícios e a cultura de excelência que oferece, o tornam uma poderosa ferramenta para estabelecer base sólida em qualquer negócio no âmbito da Transformação Digital.



3 Fases para a Implantação de uma Pipeline DevOps



3 FASES PARA A IMPLEMENTAÇÃO DE UMA PIPELINE DEVOPS

Nosso e-book está organizado didaticamente, segmentando a implantação de uma pipeline DevOps em 3 fases. As fases foram divididas em etapas, estas por sua vez em tarefas indispensáveis.

Ao final de cada fase indicamos ferramentas para implantação, preferencialmente, soluções baseadas em softwares open source, foco de atuação da 4Linux.

Fases para a implantação de pipeline DevOps

1. Infraestrutura Ágil
2. Integração Contínua
3. Entrega ou Implantação Contínua





CONHECENDO AS ETAPAS DE CADA FASE

10

FASE 1
AGILE INFRAESTRUCTURE

FASE 2
CONTINUOS INTEGRATION (CI)

FASE 3
CONTINUOS DELIVERY DEPLOYMENT CD

	ETAPA 1.1 INFRAESTRUTURA PADRONIZADA	ETAPA 1.2 MUDANÇAS CONFIÁVEIS	ETAPA 1.3 INFRAESTRUTURA EM CONTÊINERES	ETAPA 1.4 MONITORAMENTO FOCO EM NEGÓCIO
	Provisionamento automatizado de VMs	Fluxo automatizado para aplicar mudanças	Implantação de Contêineres Docker	SLAs
	Gestão de configuração	Ambiente de Homologação idêntico a Produção	Red Hat OpenShift (Infra PaaS)	Planejamento de capacidade
	Infraestrutura versionada			Indicador chave de desempenho (KPI)
	Monitoramento de SO	Testes de infraestrutura automatizados	AWS Elastic Cluster Suite (ECS)	Engenharia de Confiabilidade do site (SER)
		Rollbacks automatizados	Escalonamento automático	Integração com Sistema de Tickets
		Monitoramento de serviços	Monitoramento de Contêineres	Integração com Sistema de Inventário
		Monitoramento de Appliance		Dashboards visões Operacionais
		Monitoramento de Aplicações		Dashboards visões de Negócio
FASE 1 AGILE INFRAESTRUCTURE				
	ETAPA 2.1 DESENVOLVIMENTO ÁGIL	ETAPA 2.2 FLUXO DE DESENVOLVIMENTO AUTOMATIZADO	ETAPA 2.3 TESTES DE INTEGRAÇÃO	ETAPA 2.4 MONITORAMENTO CONTÍNUO EM (CI)
	Adoção/Otimização metodologia Scrum	Adoção de método TDD	Testes automatizados de Integração	WIP (Work in Progress)
		Fluxo automatizado para validações de commits		Média de tempo de execução dos testes
		Testes automatizados unitários		Build totais vs Bem sucedidos
		Testes automatizados de qualidade de código		Dashboards Gerenciais
		Build automatizado		
		Provisionamento automatizado em DEV		
FASE 2 CONTINUOS INTEGRATION (CI)				
	ETAPA 3.1 FLUXO DE ENTREGA AUTOMATIZADO	ETAPA 3.2 GESTÃO DE DADOS	ETAPA 3.3 IMPLANTAÇÃO CONTÍNUA	ETAPA 3.4 MONITORAMENTO CONTÍNUO EM CD
	Fluxo automatizado para validações de realeases	Adoção de Schemas	Testes de carga automatizados	Frequências dos deployments
	Testes de aceitação automatizados	Versionamento de Migrations	Implantações Blue/Green	Tempo médio com alterações
	Integração com Sistemas de Gestão de Mudanças		Implantações Canário	Mean time to recover (MTTR)
	Provisionamento automático de HOM			Taxa de recuperação de falhas
	Versionamento de artefatos			Dashboards Gerenciais
	Provisionamento em PROD após aprovação			
FASE 3 CONTINUOS DELIVERY DEPLOYMENT CD				



Etapas de adoção



Tarefas envolvidas



Testes Contínuos



Monitoramento Contínuo

Fase 1

Infraestrutura

Ágil

- Algo criado antes mesmo do termo DevOps, está presente em todos os processos durante uma pipeline de deploy de aplicações, garantindo alta performance no provisionamento e configuração de servidores, a começar dos ambientes de desenvolvimento até a produção.

Nesta fase:

1. crie servidores automaticamente em poucos minutos assegurando que funcionem como esperado.
2. garanta a padronização, evitando incidentes causados por erros humanos.
3. versione sua infraestrutura como se faz com um software.
4. aplique validações automatizadas.
5. audite mudanças, recupere versões anteriores de seus servidores sempre que preciso.

Fase
1

Todas configurações existentes, serão externalizadas em ferramentas de gestão de configurações, garantindo que os servidores estejam sempre em compliance com o padrão definido pela empresa. Mudanças em servidores ou mesmo a criação de outros novos, deverão ser implementadas de modo automatizado com poucos clicks, em poucos minutos. Toda alteração pode ser auditada e, os servidores serão auto documentados.

Tarefas:

1. implementação de ferramenta de SCM (Source Control Management) como o GITLAB.
2. instalação de ferramentas de gestão de configuração como: Puppet, Ansible e/ou Chef.
3. conversão das configurações da infraestrutura vigentes em arquivos usados pelas ferramentas de gestão de configuração.
4. versionamento das configurações, possibilitando restaurar versões anteriores da infraestrutura (roll back) Infraestrutura como código, com o GIT.
5. instalação de soluções, como Foreman e o Puppet Dashboard, para a administração visual das ferramentas de gestão de configurações.
6. Provisionamento automatizado de VMs, empregando ferramentas como Terraform, Foreman e Openstack.
7. Definição e implementação de métricas essenciais, com Zabbix e Prometheus, relacionadas a performance e ao estado de execução dos servidores.

Os arquivos utilizados pelas ferramentas de configurações implantadas, serão executados através de um fluxo automatizado de mudanças (pipeline), permitindo que testes automatizados sejam efetuados, validações manuais, complementares, sejam feitas em ambientes pré-produção, para então, aplicação automática em produção.

Tarefas:

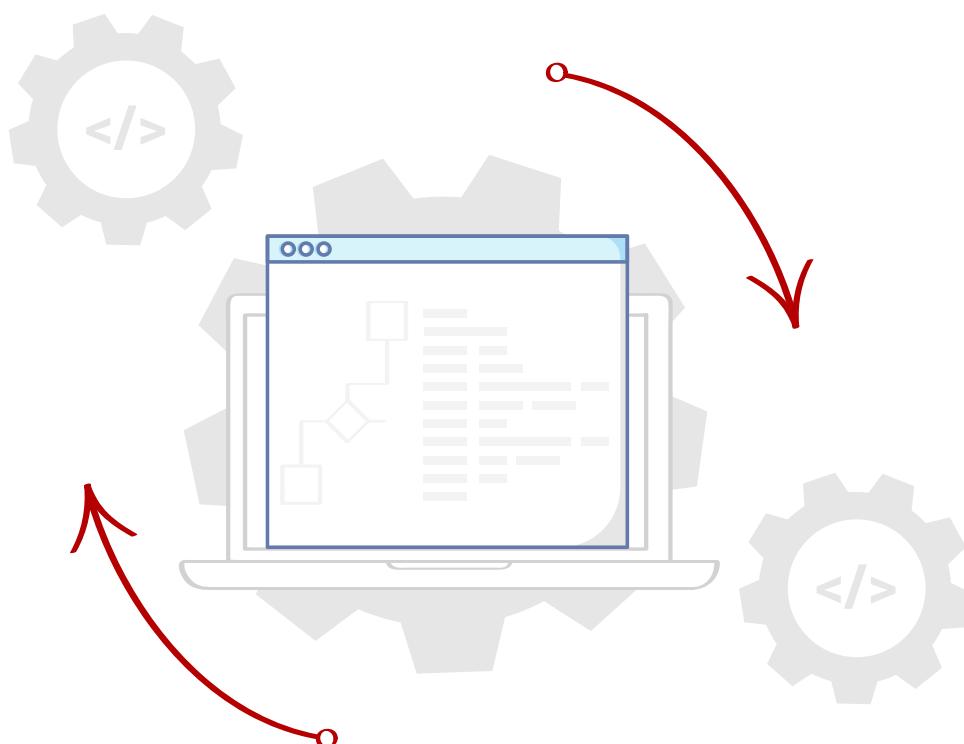
1. desenvolvimento de uma pipeline de mudanças com a ferramenta de orquestração Jenkins.
2. criação de fluxo, com iniciação automática, sempre que for gerada uma nova versão da infraestrutura no sistema de versionamento.
3. aplicação automática de mudanças em ambiente de Homologação. Este ambiente pode ser criado sob demanda e será idêntico ao de Produção.
4. criação e implantação de testes, poderão ser aplicados automaticamente, para validar se a infraestrutura segue funcional após uma mudança.
5. emprego automático das alterações na infraestrutura de produção, após aprovação manual.
6. definição e criação de ‘Botão rollback’, para reverter qualquer aplicação de mudança em ambiente de Produção.
7. monitoramento estendido, com Zabbix e Prometheus, tanto dos servidores como de serviços em execução.



Infraestrutura baseada em micro serviços, adotando contêineres Docker auto escalados conforme a demanda, balanceados, em alta disponibilidade para maior flexibilidade e melhor uso de recursos. Infraestruturas completas poderão ser geradas com um apertar de botão, equipes de desenvolvimento poderão ser autônomas na criação de seus ambientes.

Tarefas:

1. implantação de contêineres Docker.
2. definição e implantação de uma Infraestrutura baseada em PaaS (Platform as Service), com adoção das ferramentas RedHat: Openshift, AWS, Elastic, Cluster Suit (ECS), Google Kubernetes Engine (GKE).
3. monitoramento de contêineres com a ferramenta Prometheus.



Métricas consolidadas para tomadas de decisões estratégicas.

Dashboards personalizados para as equipes de operações e negócios.

Monitoramento integrado a gestão ITSM.

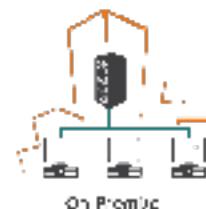
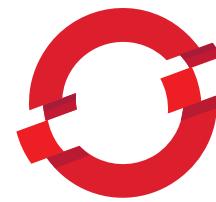
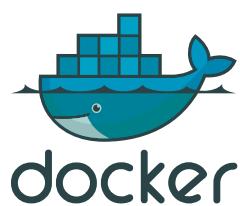
Tarefas:

1. integração do Zabbix e Prometheus aos principais sistemas de gestão de tickets e inventários do mercado.
2. monitoramento para sustentar ações de Capacity Planning.
3. desenvolvimento de dashboards nativos no Zabbix ou no Grafana.
4. desenvolvimento de painéis web com rico visual, elaborados em Python e Javascript.
5. implantação de SLAs e KPIs - Key Performance Indicator.

Etapa
1.4



TECNOLOGIAS RELACIONADAS A FASE 1 – INFRAESTRUTURA ÁGIL



Maven

Gradle





Fase 2

Integração Contínua (CI)



Desenvolvimento do software em um trabalho contínuo e conjunto.

Nesta fase:

1. encontre e investigue bugs mais rapidamente.
2. melhore a qualidade de códigos, reduza o tempo necessário para validar novas versões de software.
3. garanta que os códigos dos desenvolvedores sejam integrados sem bugs a cada iteração para um trabalho contínuo e conjunto.
4. crie ciclos curtos e recorrentes de feedbacks.



As metodologias de desenvolvimento ágil são a base da prática DevOps, sua adoção e uso disciplinado são imprescindíveis. A 4Linux há anos, pratica a metodologia SCRUM no desenvolvimento de seus produtos web, iremos ajudá-lo nessa transformação.

Tarefas:

1. consultoria para adoção de metodologia ágil, SCRUM, para empresas que utilizam métodos tradicionais de desenvolvimento de software.
2. consultoria para uso disciplinado e otimizado da metodologia ágil, SCRUM, para organizações que buscam maior eficiência em suas reuniões e elaboração de histórias.
3. implantação de ferramentas de gestão de processos como Kanban e Jira.

Etapa
2.1

Implantação de uma pipeline Continuous Integration (CI), para compilação e validações automatizadas em cada nova iteração, para garantir ritmo fluído ao desenvolvimento ágil de softwares. Esta etapa é compatível com qualquer linguagem de programação, por exemplo: Java, .NET, PHP, Python etc

Tarefas:

1. definição de estratégias para versionamento usando o GIT. Baseadas no modelo Gitflow ou, de acordo com as estratégias de lançamento que a empresa espera obter.
2. criação de uma pipeline CI na ferramenta de orquestração Jenkins.
3. aplicação de testes unitários automaticamente a cada novo commit, assegurando que não haja bug, certificando que o trabalho do desenvolvedor poderá ser considerado pronto, do ponto de vista funcional (Definition of Done).
4. 100% na incorporação de testes unitários no desenvolvimento dos softwares, adotando a metodologia TDD (Test Driven Development, ou, Desenvolvimento Orientado a Testes).
5. geração de métricas e validação automatizada da qualidade dos códigos empregando o Sonar.
6. ‘build’ automatizado e relação de dependências com as ferramentas Maven, Gradle e Ant.
7. automação no envio de feedbacks e criação de dashboards visuais sobre o estado de execução de cada etapa da pipeline CI.
8. automatização do Deployment em ambiente de desenvolvimento.



Valide se um novo código irá funcionar adequadamente quando combinado com outras partes do sistema desenvolvidas em paralelo, otimize seus processos de mesclagem. Garanta um trabalho mais fluído e colaborativo. Encontre bugs mais rapidamente.

Tarefas:

1. criação de testes de integração.
2. incorporação dos Testes de Integração ao pipeline CI para que sejam executados automaticamente a cada novo commit

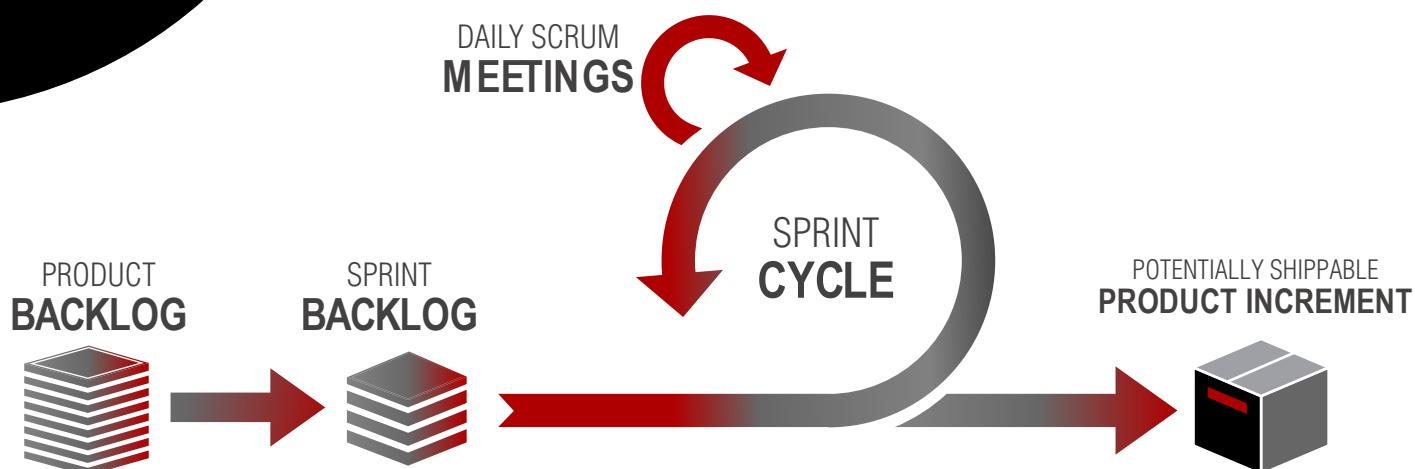
Etapa
2.3



Monitore continuamente seu processo de CI. Receba métricas para sustentar suas ações visando a melhora contínua dos processos de desenvolvimento e validação.

Tarefas:

1. monitoramento sobre todas as etapas da pipeline CI.
2. definição e implantação de métricas acerca do tempo gasto em cada tarefa da pipeline CI (WIP - Work in Progress), assim, monitorando possíveis gargalos nos processos.
3. acompanhamento do tempo de execução de testes automatizados, tendo em vista garantir a fluidez da pipeline CI.
4. controle para atentar sobre quantos builds, dos totais, submetidos a pipeline CI foram bem-sucedidos.
5. desenvolvimento de dashboards web com alertas visuais.

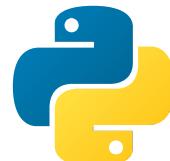


TECNOLOGIAS RELACIONADAS

A FASE 2 – INTEGRAÇÃO CONTÍNUA



Agile



Python

JUnit
JavaPHPUnit
PHPNUnit
Net CoreJasmine
JavaScript

Protractor



Spring Test



Codeception



Selenium



PhantomJS

Maven
JavaAnt Java
e PHPGoBuild
GODotNet Build
Dotnet

Gradle



MemCache





Fase 3

Entrega e Implantação

Contínua (CD)

Na cultura DevOps software em desenvolvimento não entregue, é considerado desperdício.

Nesta fase:

1. adote uma pipeline CD para garantir que novas versões dos softwares já validadas pelo processo de CI, possam ser liberadas com segurança em Produção.
2. crie um ciclo de lançamento recorrente, para entrega de valor constante a área de negócios e aos usuários.



Fase
3

A pipeline CI (continuous integration) criada na fase anterior, será ampliada para adoção de CD (continuous deployment). Novos testes serão incorporados para validações de aspectos não funcionais: acessibilidade, performance, conformidade... para provisionamento automatizado em ambiente de Homologação. Após validações finais com aprovação, seguirão para Produção.

Tarefas:

1. desenvolvimento de testes de aceitação automatizados com a ferramenta Selenium.
2. criação de testes de performance automatizados, utilizando ferramentas como o Jmeter.
3. validações finais manuais.
4. integração da pipeline CD com sistemas de gestão de mudanças, visando auditoria sobre as novas versões aprovadas para entrar em Produção.
5. provisionamento automatizado em ambiente de Homologação. Este, pode ser criado sob demanda.
6. versionamento de artefatos e criação de relação de dependências, usando ferramentas como Nexus e Artifactory.
7. prover automatização em ambiente de produção após aprovação.
8. rollbacks e feedbacks automatizados.

Etapa
3.1

Gerir mudanças em base de dados é particularmente desafiador.

Diferentemente do código da aplicação, os dados precisam ser preservados entre os lançamentos, possuem um volume muito maior de informações e não raramente possuem dados confidenciais. Mas é possível incorporar mudanças em banco de dados como parte da pipeline CD, com o versionamento e indexação das alterações e o uso de schemas.

Tarefas:

1. adoção de schemas, para agrupar objetos do banco de dados e aplicar mudanças mais facilmente.
2. adoção de recurso migration, afim de versionar mudanças em banco de dados, como é feito com o código da aplicação, permitindo assim, a aplicação e reversão de mudanças em banco de dados, de modo automatizado como parte da pipeline.



Etapa
3.2

Se as validações automatizadas, previstas pela pipeline, forem bem-sucedidas, com Implantação Contínua, novos releases serão colocados diretamente em Produção. Para minimizar os riscos envolvidos, pode-se contar com dois ambientes de Produção idênticos, comutando os acessos de usuários entre eles, de uma vez ou gradativamente. Estas técnicas, também podem ser adotadas em uma pipeline de Entrega Contínua aumentando sua confiabilidade.

Ações envolvidas:

1. adoção da técnica de implantação Blue/Green ou Red/Black. Caso seja utilizada a ferramenta Spinnaker, um novo release é provisionado, primeiramente em um ambiente de Produção secundário, idêntico ao empregado para validações finais. Uma vez aprovado o novo release, todos os acessos de usuários são comutados para o novo ambiente. Se for preciso reverter, basta redirecionar os acessos para o ambiente anterior, contendo a versão antiga da aplicação. Os ambientes se alternam a cada novo lançamento..
2. escolha da técnica de implantação Canary. Semelhante ao anterior, porém, ao invés de comutar todos acessos de usuários entre os dois ambientes de Produção, apenas parte dos acessos serão direcionados. Uma vez certificado de que não há problemas, o restante dos usuários também será comutado.
3. adesão a deploys utilizando a técnica Rolling Deployment. Essa técnica é muito utilizada quando falamos de contêineres. Pode ser definida uma quantidade de contêineres fora do ar, garantindo assim deploys zero downtime.



Monitore continuamente seu processo de CD, receba métricas para sustentar suas ações objetivando melhora contínua dos processos de entrega de novos releases.

Ações envolvidas:

1. monitoramento sobre todas as etapas da pipeline CD.
2. medida das frequências de lançamento de novos releases.
3. aferição do tempo consumido com correções e testes automatizados,
4. mean time to recovery (MTTR) – Tempo gasto e taxas para se restabelecer de eventuais falhas.
5. desenvolvimento de dashboard web com alertas visuais.

Etapa
3.4



TECNOLOGIAS RELACIONADAS A FASE 3 - ENTREGA E IMPLANTAÇÃO CONTÍNUA

Como a etapa 3 engloba as etapas 1 e 2 muitas tecnologias são as mesmas indicadas anteriormente somadas as citadas a seguir:





4DevOps em 3 Fases - V1.0

Este e-book é licenciado sob regras da licença:
Creative Commons CC BY - NC 4.0

Compartilhe, adapte, dê os crédito
e não explore comercialmente.

<https://creativecommons.org/licenses/by-nc/4.0/>

Autor: Rodrigo Dias
EXIN DevOps Master Certified

Colaboradores: Alisson Menezes
Gustavo Lira
Rodolfo Gobbi

Para dúvidas, críticas ou sugestões:
contato@4devops.com.br

Sobre Nós: 4devops é a unidade da 4linux (www.4linux.com.br) especializada em serviços de consultoria DevOps e Cloud priorizando o uso de Software Open Source.

www.4devops.com.br