Devops

Capítulo 5. Ciclos contínuos

Prof. Thiago Chierici

Aula 5.1. Diretivas de compilação e empacotamento

Nesta aula

- Algumas considerações importantes para seu processo de compilação e empacotamento.

Compilação, empacotamento e implantação

- Quando o deploy não é feito corretamente, todo o esforço anterior é jogado fora!

Compilação

- Simples.
- Rápida.
- Boa gestão de dependências externas.

Empacotamento

- Automatizar todo o processo.
- Um pacote e vários ambientes.
- Clean build.

Implantação

- Deploy:
  - Automatizar.
  - Gestão de configurações com ferramentas.
  - Atenção com a segurança.

Compilação, empacotamento e implantação

- Todo cuidado é pouco na hora da implantação.

Conclusão

☑ Automatização e velocidade são importantes em todas etapas do processo.

Próxima aula

- Algumas diretivas de integração contínua.

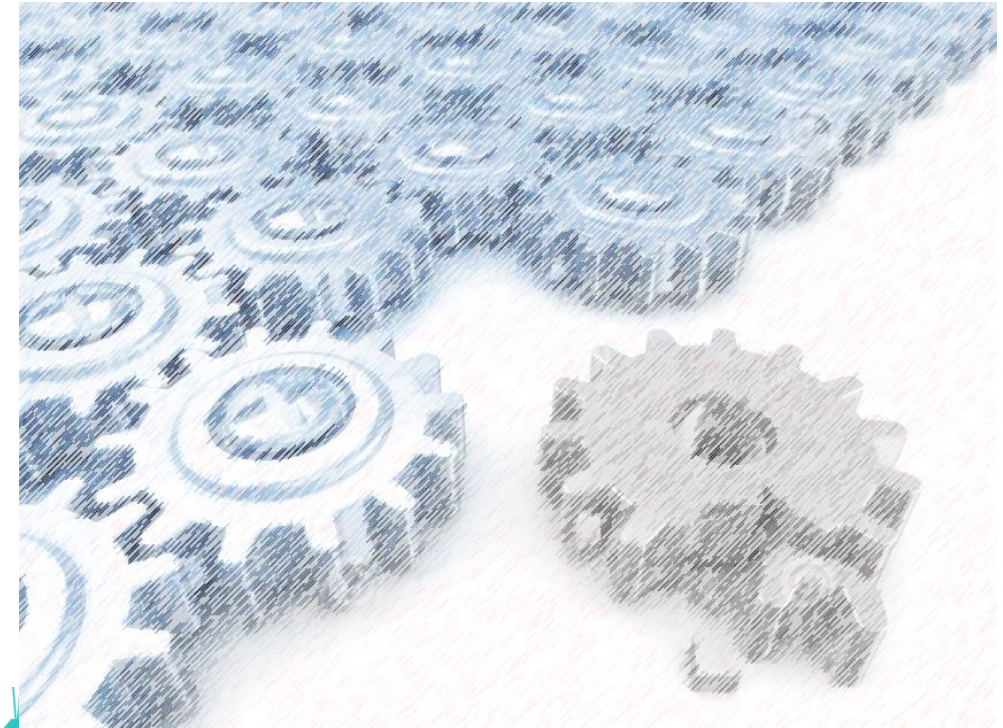
Aula 5.2. Diretivas de integração contínua

Nesta aula

- Algumas considerações importantes para o seu processo de integração contínua em uma abordagem Devops.

Diretivas de CI – Pré-integração

- Não dê check-in (commit) em códigos que quebrem o build.



Diretivas de CI – Pré-integração

- Sempre executar todos os testes automatizados no ambiente local, antes de enviar para o servidor.

Diretivas de CI – Pré-integração

- Nunca vá para casa se o build automatizado estiver quebrado.

Pós-entrega – Agir rápido

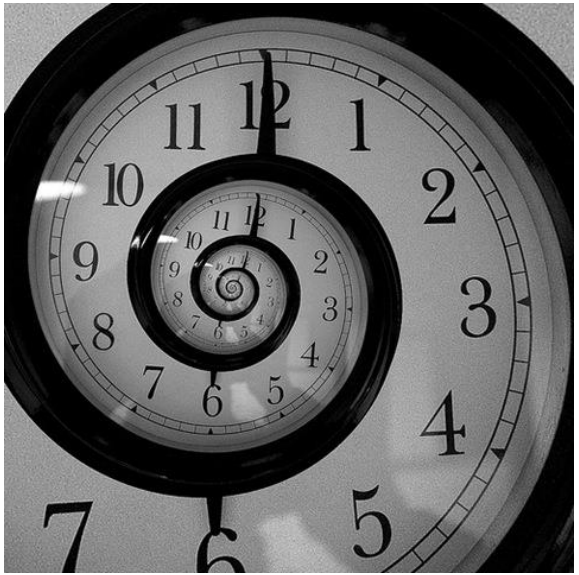
- “Você não se afoga por cair na água, você se afoga por ficar lá.”
  - Edwin Louis Cole.

Pós-entrega – Agir rápido

- Esteja sempre preparado para voltar para a versão anterior.

Pós-entrega – Agir rápido

- Defina um tempo limite para correção do build antes de voltar para a versão anterior.



CI – Outras práticas sugeridas

- Quebrar o build quando as diretivas arquiteturais não forem seguidas corretamente.

CI – Outras práticas sugeridas

- Quebrar o build se algum teste estiver muito lento.

CI – Outras práticas sugeridas

- Quebrar o build para alertas ou problemas de padrões de codificação.

Conclusão

☑ Pequenas práticas na rotina de cada membro do time podem ajudar bastante na eficiência do processo e mecanismos Devops.

Próxima aula

- Técnicas (bem interessantes) de implantação.
- Como garantir disponibilidade, velocidade e confiança no processo de deploy.

Aula 5.3. As 7 técnicas de implantação mais importantes para o seu sucesso com Devops

Nesta aula

- Como utilizar a infraestrutura, ferramentas e processos para fazer entregas contínuas com maior agilidade e menor risco.

Agilidade

- How long would it take your organization to deploy a change that involves just one single line of code?
  - Mary Poppendieck.

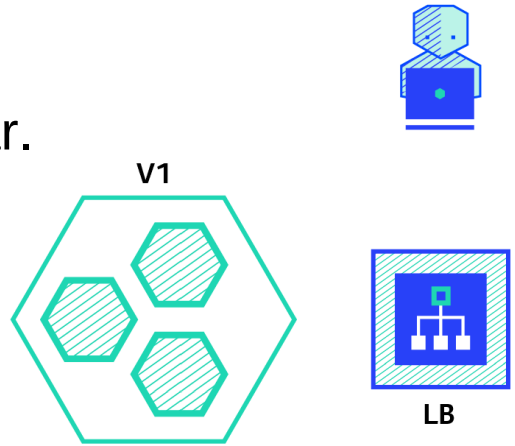
Fontes:
https://thenewstack.io/deployment-strategies/
https://medium.com/@cgrant/deployment-strategies-release-best-practices-6e557c3f39b4

Environment separation

- Uma das formas mais simples de oferecer uma versão diferente pra um grupo de usuários sem riscos, ou impactos, na versão live.
- Manualmente podemos sugerir um link alternativo que use a nova versão:
  - https://beta.meusite.com.br

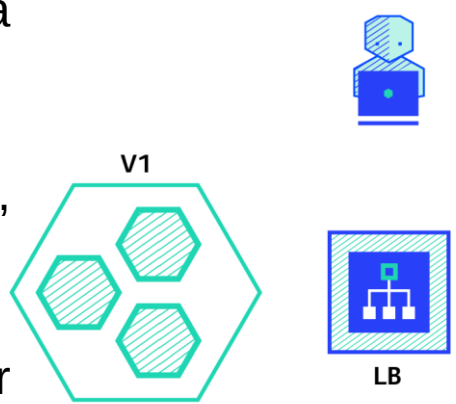
Recreate

- Método simplista.
- Desligamos a infra V1.
- Recriamos a infra V2.
- Significa um tempo fora do ar.



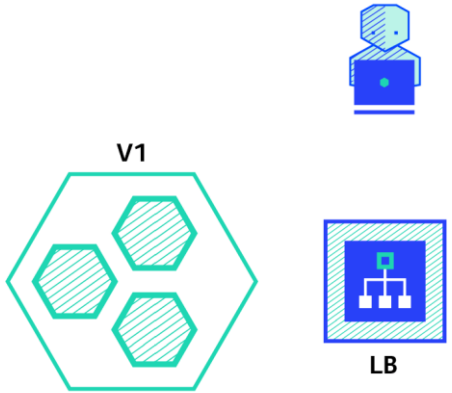
Shadow

- Método um pouco mais elaborado.
- O ambiente V2 é criado enquanto o V1 ainda funciona.
- Todo o acesso vai para o novo ambiente, assim como para o antigo.
- Depois da validação, o servidor V1 pode ser removido.
- Atenção especial ao estado da aplicação durante a transição.



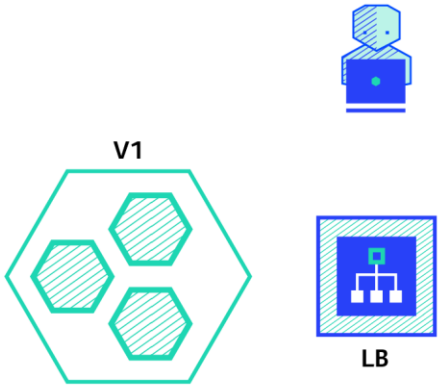
Ramped

- Viável em ambientes com clusters de vários servidores.
- Os servidores com a V1 vão sendo substituídos aos poucos.
- Pode ou não haver paralelismo a cada substituição.



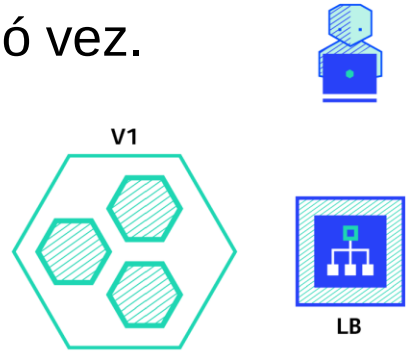
Blue/green

- Se parece com a Ramped, porém todos os nós são criados de uma vez.
- Se tudo estiver ok, a carga é direcionada para o novo cluster.
- Depois o cluster antigo pode ser a



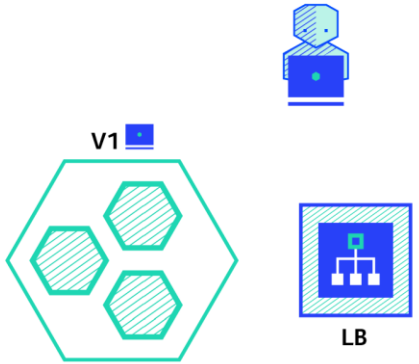
Canary

- Um novo cluster é criado assim como na Blue/Green.
- A diferença da Canary para as anteriores é que a migração da carga é feita aos poucos (90/10).
- Em especial quando não temos testes automatizados suficientes para confiarmos na migração de uma só vez.

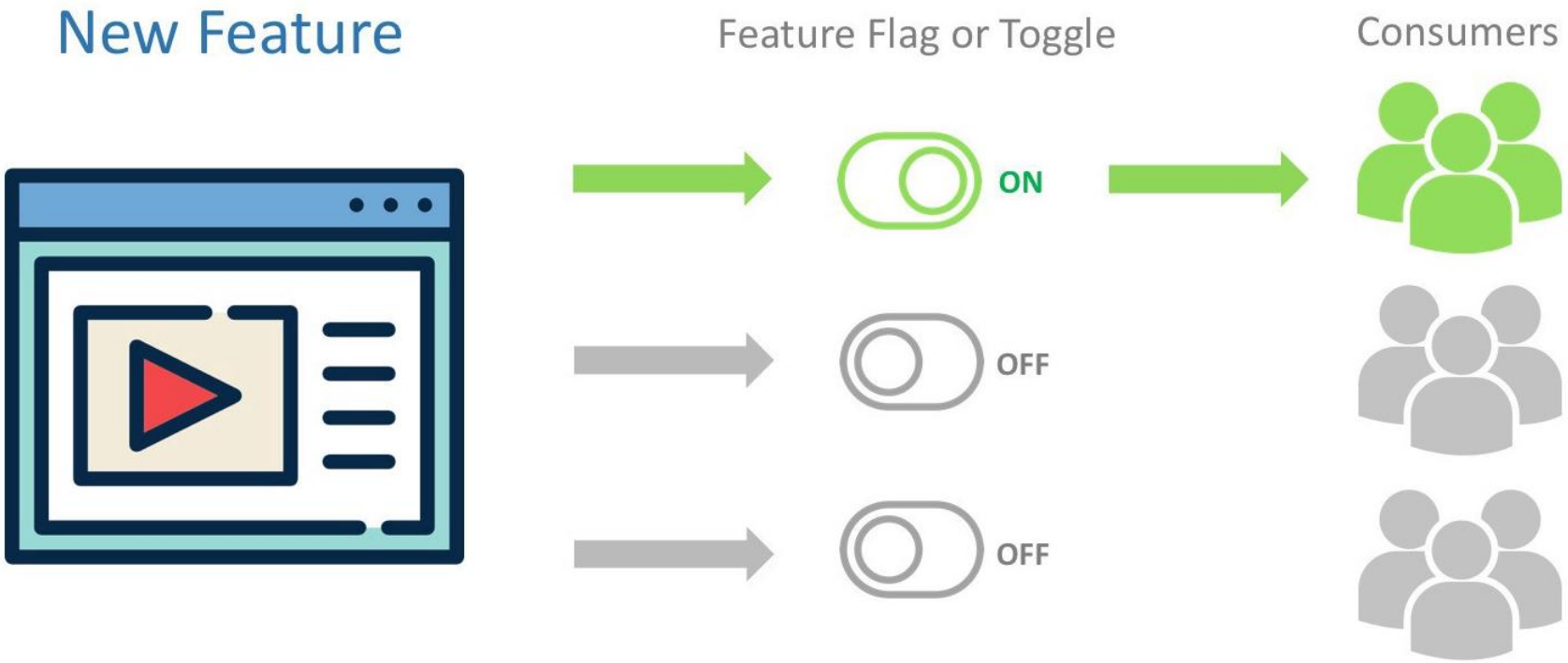


A/B testing

- Usar qualquer atributo técnico ou do negócio para segmentar os usuários.
- Cada grupo acessa uma versão diferente.



Bônus: Feature toggles



Fonte: https://launchdarkly.com/blog/author/justinucd/

Conclusão

☑ Não há uma única forma certa de fazer deploys.

☑ Técnicas de deploy podem acelerar e dar mais robustez ao processo de entrega.

Próxima aula

- Depois de ver algumas formas certas de se fazer deploys, vamos ver algumas formas conhecidas de fazer os deploys darem errado. Os antipadrões de entrega de software.

Aula 5.4. Como fazer sua entrega dar errado

Nesta aula

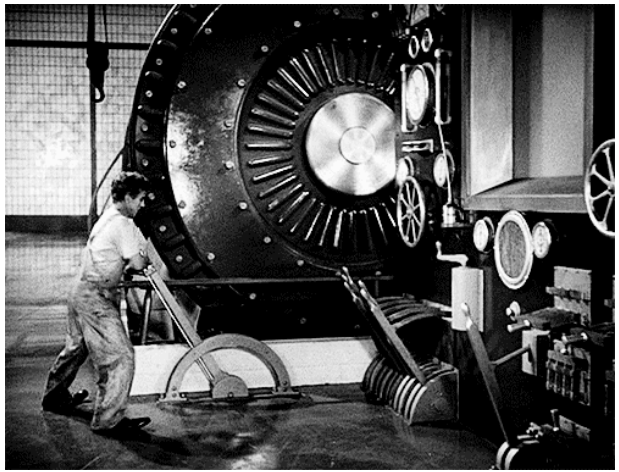
- Conheça alguns antipadrões de entrega de software.

Antipadrões de entrega de software

- “O tolo nunca aprende, o inteligente aprende com sua própria experiência e o sábio aprende com a experiência dos outros.”
  - Provérbio chinês.

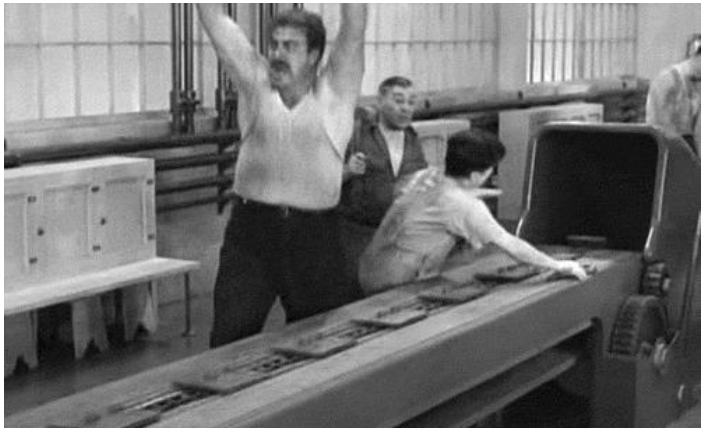
Implantação manual de software

- A implantação deve ser automatizada e disparada por um simples comando.
- Ferramental já está altamente evoluído.
- Não é feito por costume ou falta de conhecimento.
- Processo manual é demorado e falho.



produção ao final do desenvolvimento

- Diferença de ambientes pode causar grandes transtornos.
- Validar comportamento no ambiente de produção ou similar.
- PoCs.
- Deixar para a última hora pode invalidar todo o desenvolvimento.



Gestão de configuração de produção manual

- Popularização de várias ferramentas nos últimos anos.
- Muitas equipes insistem no argumento da segurança para manter processos manuais.
- Processos manuais são muito mais inseguros.

Arquitetura monolítica x microservices

- Arquiteturas monolíticas chegaram a ser consideradas fortemente desaconselháveis.
  - Já vemos grandes cases do caminho oposto:
    - Ex: https://segment.com/blog/goodbye-microservices/
- Toda decisão arquitetural pode trazer algum efeito colateral:
  - Versionamento, dependências, merges, deploys, permissões, performance, produtividade, gestão, etc.

Conclusão

- ☑ Podemos aprender pelos exemplos e pelos contraexemplos.
- ☑ O mercado pode nos trazer aprendizados para evitar de cometermos erros que são comuns em várias empresas.
- ☑ Algumas vezes, um consenso de mercado pode mudar ao longo do tempo, graças, principalmente, às novas experiências e ferramentas.

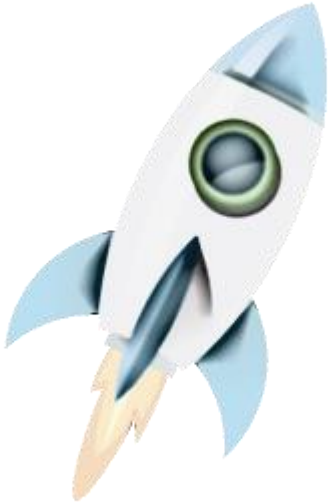
Próxima aula

- Veremos como a arquitetura pode favorecer as entregas quando utilizamos unidades de desenvolvimento e implantação menores.

Aula 5.5. Inspiração - Flickr

Nesta aula

- Conheça as lições compartilhadas pela plataforma Flickr em seu case de entrega contínua (2009):
  - 10 implantações por dia.
  - 3 bilhões de fotos.
  - 40k fotos por segundo.



https://www.slideshare.net/jallspaw/10-deploys-per-day-dev-and-ops-cooperation-at-flickr/23-Role_congurationmanagementOS_imaging

Inspiração

- "Se você pensa que pode ou se você pensa que não pode, de qualquer forma, você tem toda a razão."
  - Henry Ford.

Se tiver que escolher uma ação

- Automatize toda a infraestrutura:
  - Muda a escala da agilidade de entrega.

Controle de versão

- Compartilhado entre times:
  - Infra.
  - Security.
  - Development.
- Única fonte:
  - Todos sabem encontrar o que precisarem.

Compilação e implantação

- Um passo:
  - Dispositivo físico pode motivar o time.
  - Simplicidade.

http://deploybybutton.com/
https://blog.github.com/2018-08-16-how-to-deploy-to-production-with-an-actual-button/

Feature flags

- Menos branches:
  - Menos merges.

Métricas

- Compartilhadas:
  - Valores: comunicação, feedback, visibilidade e confiança.

Monitoração

- Logs.
- Alertas.
- Robôs.
- Buscas.

Cultura

- Respeito:
  - Diferenças e opiniões.
- Confiança:
  - Confiar em permissão para todos do time.

- Falhar é saudável (e inevitável):
  - Procedimentos/mecanismos para restaurar a ordem.
- Evitar a culpa:
  - Pular fases de tentar achar culpados, mas não pular a compreensão do problema (causa raiz).

Conclusão

☑ IaC = Agilidade.

☑ Simplicidade = Eficiência.

☑ Cultura é a base de tudo.

Próxima aula

- Um novo capítulo focado em tudo que envolve informações coletadas da operação, como monitoramento, logs e métricas sem as quais não conseguimos evoluir.