



igti

—

Devops

Capítulo 6. Produção

Prof. Thiago Chierici



Aula 6.1. Sete métricas importantes para avaliar o sucesso do Devops



- ❑ Importância das métricas na melhoria contínua.
- ❑ Principais métricas pra avaliar o Devops.

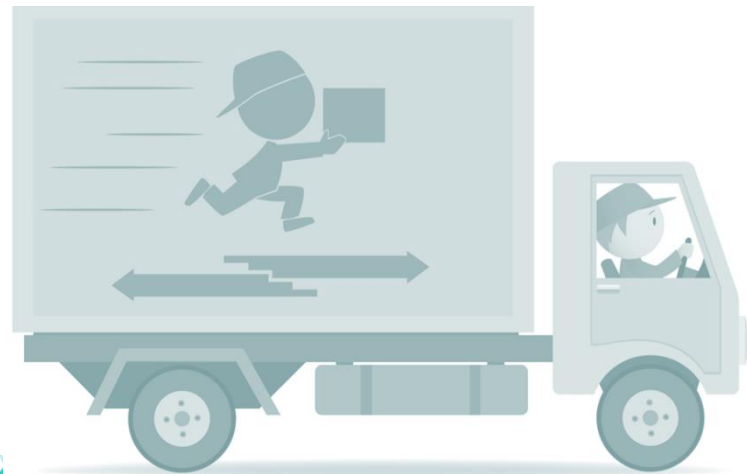


- "Aquilo que não se pode medir, não se pode melhorar."
 - William Thomson.

Fonte: <https://stackify.com/15-metrics-for-devops-success/#post-14669-gi64o3v428nm>
<https://www.riverbed.com/blogs/seven-metrics-matter-measuring-devops-success.html>



- Como evoluir fazendo um deploy por ano?
- É crucial ter alta frequência.
- Evolução:
 - 1/ano -> 1/mês -> 1/semana -> 1/dia.



- Se o deploy demorar, não conseguiremos fazer muitos.
 - Relação com complexidade.
 - Relação com passos manuais.
- Evolução:
 - Dias -> horas -> minutos -> segundos.



Taxa de falha em deploys



- Falhas são esperadas.
- Precisamos reduzir a ocorrência.
- Quando começamos a implantar Devops, pode haver um aumento inicial nas falhas.
 - Automatizar cada solução.
 - Um problema não pode reaparecer.



- Se falhas acontecem, precisamos ser rápidos na recuperação.
- Automatizar rollback.
- Precisamos ser muito rápidos.



- Saber o que é mais utilizado no sistema ajuda em vários aspectos:
 - Estratégia de testes.
 - Disponibilidade.
 - Foco em melhorias.
 - Descontinuidade.



- Pode inviabilizar automação completa do ciclo.
- Pode tirar o efeito psicológico de descobrir uma falha.
- Tipos:
 - Falsos positivos.
 - Falsos negativos.



- Alguns fatores podem causar indisponibilidade:
 - Deploys.
 - Manutenções programadas.
 - Falhas não previstas.
 - Bugs.
 - Invasões.
- 99,8% -> 17 horas fora do ar por ano:
 - Pouco mais de uma hora por mês.



- ☑ Sem métricas não evoluímos.
- ☑ O uso de métricas que já foram avaliadas com sucesso por outras empresas pode nos ajudar a sermos assertivos.



Próxima aula



- ❑ Como monitorar seu ambiente de produção.





Aula 6.2. Continuous monitoring



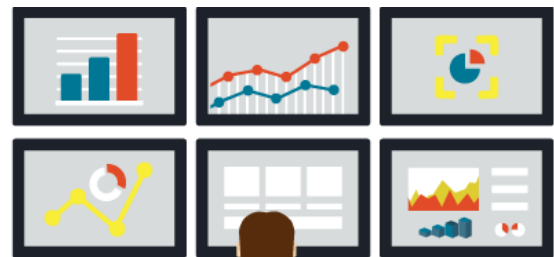
- ❑ Os principais tipos de monitoramento necessários para a sustentação da nossa estratégia Devops.
- ❑ Sugestão de ferramentas relacionadas.



- Infraestrutura:
 - É importante conhecer o perfil de uso de recursos de seu sistema.
 - Gargalos comuns:
 - Rede.
 - Memória.
 - Disco.
 - CPU.
 - Escolha ferramentas que se adequem à dinâmica de criação da infra.



- Clássicas:
 - Exemplos: [Nagios](#) / [Zabbix](#).
 - Problemas.
- Devops:
 - Exemplos: [Sensu](#), [Prometheus](#), [SysDig](#) e [New Relic Infrastructure](#).
- Nuvem:
 - Exemplos: [AWS CloudWatch](#) / [StackDriver](#).



Tipos de monitoramento



- Aplicação:
 - Consumo de recursos da aplicação.
 - Profiling de memória.
 - Análise de acordo com a tecnologia utilizada.



- Exemplos:
 - [New Relic.](#)
 - [AppDynamics.](#)
 - [Compuware APM.](#)
 - [Boundary.](#)



- Disponibilidade:
 - Muitas ferramentas online disponíveis.
 - Pode ser um indicador importantíssimo de que qualquer problema aconteceu.
 - Ferramentas:
 - Pingdom.
 - Statuscake.
 - Updown.
 - StatusOK.



- ☑ As principais categorias de monitoramento são de aplicação e de infraestrutura.
- ☑ Vimos algumas estratégias e ferramentas para ter um bom monitoramento da sua operação Devops.



- ❑ Outros mecanismos importantes para saber o que acontece em produção: alertas e logs.





Aula 6.3. Alerting / logging



- ❑ Como garantir a visibilidade das informações nos ambientes produtivos.



- “O que os olhos não veem, o coração não sente.”
 - Ditado popular.



- Importante ter algum tipo de mecanismo de alerta para problemas em produção:
 - Agilidade na identificação contribui com a agilidade na resolução.
 - Ferramentas de captura de exceções não tratadas, testes de aceitação e monitoramento de disponibilidade.



- Ambiente distribuído e heterogêneo gera outras dificuldades.
- Ferramentas específicas também podem ajudar:
 - Crashlytics.



- Não precisa ser ativo.
- Sempre que um problema é encontrado, precisamos de pistas – o log.
- Mecanismo ativo com alguma inteligência pode ser um grande diferencial.



- ☑ Visibilidade das informações e métricas pode garantir a melhoria contínua do processo Devops.



Próxima aula



- ❑ Mais um case de inspiração Devops.





Aula 6.4. Inspiração - Telemetria na Netflix



- ❑ Case Netflix sobre monitoramento de ambiente de produção.



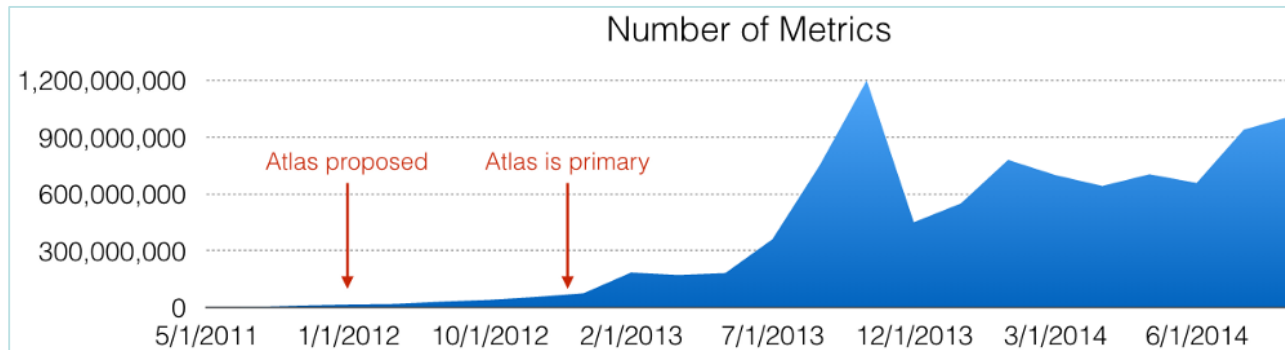


- Abordagens:
 - Ferramenta comercial.
 - Scripts próprios:
 - Dados relacionados à linha do tempo.
 - Dificuldade de escalar para milhões de time series.
- Em 2012 iniciaram a criação da Atlas, a nova geração do ferramental próprio para monitoração.

Objetivos da plataforma



- Api comum.
- Escala:



- Dimensões:
 - `com.netflix.eds.nccp.successful.requests.uiversion.nccprt-authorization.devtypid-101.clver-PHL_0AB.uiver-UI_169_mid.geo-US`

Fonte: <https://medium.com/netflix-techblog/introducing-atlas-netflixs-primary-telemetry-platform-bd31f4d8ed9a>

O que foi construído



- Camada de busca.
- Deep links.
- Dados em memória.
- Foco em performance, resiliência e custos.

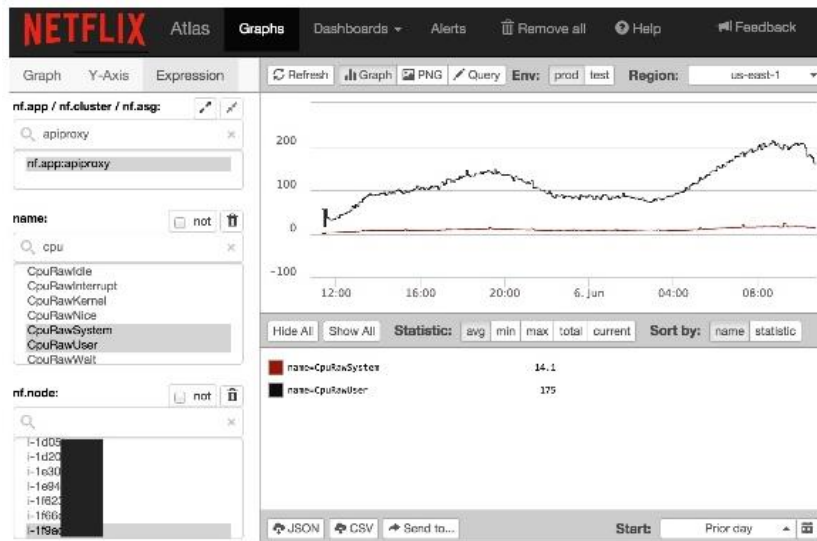


Fonte: <https://medium.com/netflix-techblog/introducing-atlas-netflixs-primary-telemetry-platform-bd31f4d8ed9a>

O sistema

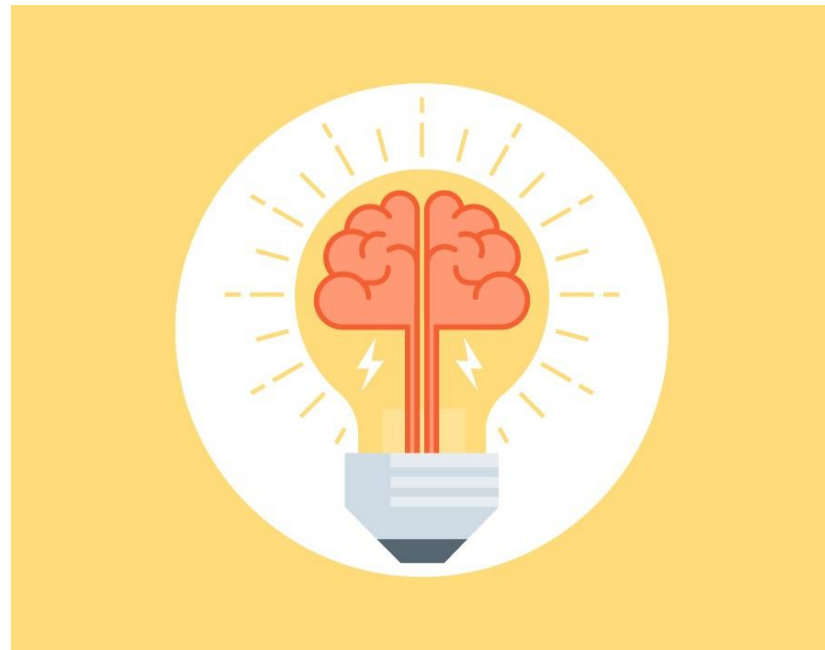


- Interface:
 - Navegação, queries, dashboards e alertas.
- Tempo real:
 - Volume de métricas.
 - Análise de implantação.
 - Detecção de anomalias.
 - Ações.



O que aproveitar

- Mentalidade.
- Código-fonte.
- Uso do ferramental.



- ☑ A Netflix é um dos grandes cases em Devops.
- ☑ Várias ferramentas utilizadas pela Netflix estão disponíveis e podem nos ajudar a implementar nossos próprios casos de sucesso.



Próxima aula



- ❑ Um novo capítulo focado em segurança.

