DevOps – Atividade 2

***Thiago Carvalho Carneiro da Fontoura***

***0030482421026***

1. **Introdução**

Um composto de Dev (desenvolvimento) e Ops (operações), o DevOps é a união de pessoas, processos e tecnologias para fornecer continuamente valor aos clientes.

O DevOps permite que funções anteriormente isoladas, como desenvolvimento, operações de TI, engenharia da qualidade e segurança, agora atuem de forma coordenada e colaborativa para gerar produtos melhores e mais confiáveis. Ao adotar uma cultura de DevOps em conjunto com as práticas e ferramentas de DevOps, as equipes ganham a capacidade de responder melhor às necessidades dos clientes, aumentar a confiança nos aplicativos que constroem e cumprir as metas empresariais mais rapidamente.

Equipes que adotam a cultura, as práticas e as ferramentas de DevOps apresentam alto desempenho, criando produtos melhores, com mais rapidez, para maior satisfação do cliente. Esse aumento na colaboração e na produtividade também é essencial para cumprir metas empresariais como as seguintes:

O DevOps influencia o ciclo de vida do aplicativo em todas as fases do planejamento, do desenvolvimento, da entrega e da operação. Cada fase depende das demais e elas não são específicas da função. Em uma verdadeira cultura de DevOps, cada função está envolvida de alguma forma em cada fase.

1. **Ciclo de Vida**

* **Planejamento**

O DevOps influencia o ciclo de vida do aplicativo em todas as fases do planejamento, do desenvolvimento, da entrega e da operação. Cada fase depende das demais e elas não são específicas da função. Em uma verdadeira cultura de DevOps, cada função está envolvida de alguma forma em cada fase.

* **Desenvolvimento**

A fase de desenvolvimento inclui todos os aspectos da codificação – gravação, teste, revisão e integração do código pelos membros da equipe – bem como a compilação do código em artefatos de compilação, que podem ser implementados em vários ambientes. As equipes de DevOps buscam inovar rapidamente sem sacrificar a qualidade, a estabilidade e a produtividade. Para fazer isso, elas usam ferramentas extremamente produtivas, automatizam etapas elementares e manuais e iteram em pequenos incrementos por meio de testes automatizados e integração contínua.

* **Entrega**

A entrega é o processo de implantação de aplicativos nos ambientes de produção de maneira consistente e confiável. A fase de entrega também inclui a implantação e a configuração da infraestrutura fundamental totalmente governada que compõe esses ambientes.

Na fase de entrega, as equipes definem um processo de gerenciamento de versão com estágios claros de aprovação manual. Elas também definem portões automatizados que movem os aplicativos entre os estágios, até que sejam disponibilizados aos clientes. A automação desses processos os torna escalonáveis, repetíveis e controlados. Dessa forma, as equipes que praticam o DevOps podem frequentemente atuar e entregar com facilidade, confiança e tranquilidade.

* **Operação**

A fase de operação envolve manter, monitorar e solucionar problemas de aplicativos em ambientes de produção. Ao adotar as práticas de DevOps, as equipes trabalham para garantir a confiabilidade do sistema, a alta disponibilidade e o objetivo de tempo de inatividade igual a zero, reforçando a segurança e a governança. As equipes de DevOps buscam identificar os problemas antes que eles afetem a experiência do cliente e mitigar os problemas rapidamente quando ocorrem. Manter esse nível de vigilância requer telemetria avançada, alertas acionáveis e visibilidade total sobre os aplicativos e o sistema subjacente.

1. **Cultura do DevOps**

* **Colaboração, visibilidade e alinhamento**

O DevOps influencia o ciclo de vida do aplicativo em todas as fases do planejamento, do desenvolvimento, da entrega e da operação. Cada fase depende das demais e elas não são específicas da função. Em uma verdadeira cultura de DevOps, cada função está envolvida de alguma forma em cada fase.

* **Mudanças de escopo e de responsabilidade**

À medida que as equipes se alinham, elas assumem a responsabilidade e se envolvem em outras fases do ciclo de vida, não apenas naquelas que são essenciais para suas funções. Por exemplo, os desenvolvedores tornam-se responsáveis não apenas pela inovação e a qualidade estabelecidas na fase de desenvolvimento, mas também pelo desempenho e estabilidade que as alterações trazem para a fase de operação. Ao mesmo tempo, os operadores de TI certamente incluirão governança, segurança e conformidade na fase de planejamento e desenvolvimento.

* **Ciclos de versão mais curtos**

As equipes de DevOps se mantêm ágeis lançando versões de software em ciclos curtos. Ciclos de versão mais curtos facilitam o planejamento e o gerenciamento de riscos, pois o progresso é incremental, reduzindo também o impacto sobre a estabilidade do sistema. A redução do ciclo de versão também permite que as organizações se adaptem melhor e reajam de maneira mais ágil à evolução das necessidades dos clientes e à pressão da concorrência.

* **Aprendizado contínuo**

As equipes de alto desempenho de DevOps estabelecem uma mentalidade de crescimento. Elas falham rapidamente e incorporam aprendizados aos seus processos, melhorando continuamente, aumentando a satisfação do cliente e acelerando a inovação e a capacidade de adaptação ao mercado. O DevOps é uma jornada, portanto, há sempre espaço para crescer.

1. **Práticas do DevOps**

* **CI/CD (Integração Contínua e Entrega Contínua)**

O gerenciamento de configuração refere-se ao gerenciamento do estado dos recursos em um sistema, incluindo servidores, máquinas virtuais e bancos de dados. Usando ferramentas de gerenciamento de configuração, as equipes podem implementar alterações de maneira controlada e sistemática, reduzindo os riscos de modificar a configuração do sistema. As equipes usam ferramentas de gerenciamento de configuração para acompanhar o estado do sistema e ajudar a evitar desvios de configuração, já que é assim que a configuração de um recurso de sistema se desvia do estado desejado ao longo do tempo.

Praticadas em conjunto com a infraestrutura como código, a definição e a configuração do sistema são fáceis de padronizar e automatizar, o que ajuda as equipes a operar ambientes complexos em escala.

* **Controle de Versão**

O controle de versão é a prática de gerenciar código por meio de versões, acompanhando as revisões e o histórico de alterações para facilitar a revisão e a recuperação do código. Essa prática geralmente é implementada usando sistemas de controle de versão, como o Git, que permitem que vários desenvolvedores colaborem na criação do código. Esses sistemas fornecem um processo claro para mesclar alterações de código ocorridas no mesmo arquivo, gerenciar conflitos e reverter alterações para estados anteriores.

O uso do controle de versão é uma prática fundamental de DevOps que ajuda as equipes de desenvolvimento a trabalhar juntas, dividir as tarefas de codificação entre os membros da equipe e armazenar o código completo para fácil recuperação, se necessário.

O controle de versão também é um elemento necessário em outras práticas, como a integração contínua e a infraestrutura como código.

* **Desenvolvimento de software Agile**

O Agile é uma abordagem de desenvolvimento de software que enfatiza a colaboração da equipe, os comentários do cliente e do usuário e a alta adaptabilidade às mudanças por meio de ciclos de versão curtos. As equipes que praticam o Agile fornecem mudanças e aprimoramentos contínuos aos clientes, coletam seus comentários, aprendem e se ajustam com base nos desejos e nas necessidades dos clientes. O Agile é substancialmente diferente de outras estruturas mais tradicionais, como a cascata, que inclui longos ciclos de versão definidos por fases sequenciais. O Kanban e o Scrum são duas estruturas populares associadas ao Agile.

* **Infraesturura como código**

A infraestrutura como código define os recursos e as topologias do sistema de uma maneira descritiva que permite às equipes gerenciar esses recursos da maneira como codificariam. Essas definições também podem ser armazenadas e versionadas em sistemas de controle de versão, nos quais podem ser revisadas e revertidas – novamente como código.

A prática de infraestrutura como código ajuda as equipes a implantar os recursos do sistema de maneira confiável, repetível e controlada. A infraestrutura como código também ajuda a automatizar a implantação e reduz o risco de erro humano, especialmente para ambientes grandes e complexos. Essa solução confiável e repetível para a implantação de ambientes permite que as equipes mantenham ambientes de desenvolvimento e teste idênticos ao ambiente de produção. Da mesma maneira, a duplicação de ambientes para diferentes data centers e plataformas em nuvem também se torna mais simples e eficiente.

* **Gerenciamento de configuração**

O gerenciamento de configuração refere-se ao gerenciamento do estado dos recursos em um sistema, incluindo servidores, máquinas virtuais e bancos de dados. Usando ferramentas de gerenciamento de configuração, as equipes podem implementar alterações de maneira controlada e sistemática, reduzindo os riscos de modificar a configuração do sistema. As equipes usam ferramentas de gerenciamento de configuração para acompanhar o estado do sistema e ajudar a evitar desvios de configuração, já que é assim que a configuração de um recurso de sistema se desvia do estado desejado ao longo do tempo.

Praticadas em conjunto com a infraestrutura como código, a definição e a configuração do sistema são fáceis de padronizar e automatizar, o que ajuda as equipes a operar ambientes complexos em escala.

* **Monitoramento contínuo**

O monitoramento contínuo significa ter visibilidade total e em tempo real sobre o desempenho e a integridade de toda a pilha de aplicativos, desde a infraestrutura subjacente que está executando o aplicativo até os componentes de software de nível superior. Essa visibilidade consiste na coleta de telemetria e metadados, bem como na definição de alertas para condições predefinidas que merecem a atenção de um operador. A telemetria compreende dados de eventos e logs coletados de várias partes do sistema e armazenados em um local onde possam ser analisados e consultados.

As equipes de alto desempenho de DevOps garantem a definição de alertas acionáveis e significativos e a coleta de uma telemetria abundante para que seja possível extrair insights de grandes quantidades de dados. Esses insights ajudam a equipe a mitigar problemas em tempo real e a descobrir como melhorar o aplicativo em ciclos de desenvolvimento futuros.

1. **Conclusão**

Em resumo, DevOps acelera os processos necessários para levar uma ideia do desenvolvimento à implantação. Essa abordagem exige que as equipes de desenvolvimento e operações se comuniquem com frequência e conduzam seu trabalho com empatia pelos colegas. Também são necessários provisionamento flexível e escalabilidade. As equipes podem obter os recursos de que precisarem por meio da automação e do self-service. Os desenvolvedores, que normalmente criam códigos em um ambiente de desenvolvimento padrão, trabalham em estreita colaboração com a equipe de operações de TI para acelerar compilações, testes e lançamentos, sem perder a confiabilidade. Com esse conjunto de práticas, são oferecidas soluções mais práticas e rápidas, permitindo entregar serviços valor de negócio com mais qualidade.

1. **Referências**

* <https://azure.microsoft.com/pt-br/resources/cloud-computing-dictionary/what-is-devops#tabx7a30a21035a04006a4836916b3d057c7>
* <https://www.redhat.com/pt-br/topics/devops>