FIAP

MBA

MBA - Cyber Security
Governance & Management

Cloud Computing Security, DevOps e DevSecOps

Turma: 67SEG

Prof. Me. Nivaldo T. Marcusso

Prof. Me. Nivaldo Tadeu Marcusso

- 59 anos, formado em Engenharia Eletrônica, MBA em Conhecimento, Tecnologia e Inovação pela FIA/USP e Pós-MBA em Gestão avançada pela FIA / USP, Membro da Wharton Fellows.
- Certificação Executiva em Estratégia e Inovação pela MIT Sloan e Especialização em: Gestão Estratégica de TI (SUIT) pela Universidade de Stanford, Estratégia e Serviços de TI (DIS) pela Universidade de Harvard, Gestão Internacional pela Universidade Euromed de Marseille/FEA-USP e Gestão de Conhecimento pela Universidade de Lyon e FEA/USP.
- Professor de Pós-Graduação Lato Sensu (cursos MBA) em Estratégia, Inovação, Gestão do Conhecimento, Tecnologia da Informação e Educação a distância pela UNISAL, USP, FIA, FIAP, FATEC e e palestrante em conferências nacionais e internacionais de TI, Inovação, Tecnologia Educacional e Educação à distância.
- Eleito em 2010, 2009, 2008, 2007, CIO do ano no segmento de educação no Brasil, pela HITEC, revista Computerworld e 1º lugar entre os CIOs, das 100+ Empresas Inovadoras em TI na categoria de serviços diversos, pela revista Information Week.
- Experiência de negociação e liderança no desenvolvimento de parcerias internacionais com empresas e Universidades, para a transferência de tecnologias aplicadas a educação e banking, tendo visitado mais de 15 países nos últimos 13 anos, como os EUA, China, Inglaterra, França, Alemanha, Irlanda, Tunísia, Espanha, Chile entre outros.
- Membro de Comunidades, Associações e Sociedades focadas em gestão da tecnologia, da estratégia, da inovação e da educação a distância, como o ISPIM (Noruega), IBGC, Praxis (Brasil), ABED (Brasil), e-learning Brasil, Educause (EUA), FIRST (EUA) e BDRA (Inglaterra).
- Coautor e coordenador da coleção “Tecnologia e Educação”, com os livros eletrônicos (eBooks): Tecnologia e Aprendizagem e a Tecnologia transformando a Educação.
- Experiência de mais de 25 anos na gestão da TI, EAD, planejamento estratégico, Inovação, RH, Finanças em empresas como Digilab, Fundação Bradesco, Bradesco, Anhembi Morumbi.
- Atualmente além de Professor da USP, FIA, FDC, FIPE, FIAP, UNISAL e FACAMP, além de Consultor da FIA (TI, Tecnologia Educacional e EAD) , FDC (Processos de Negócios) , 4Strategis (Planejamento Estratégico, Modelagem de Negócios e Inovação) e da MARCX (TI, e-Learning e Mobile Learning).

Agenda

Aula 1 : Arquitetura e Segurança em Cloud

Aula 2: DevOps, DevSecOps e DevRiskOps

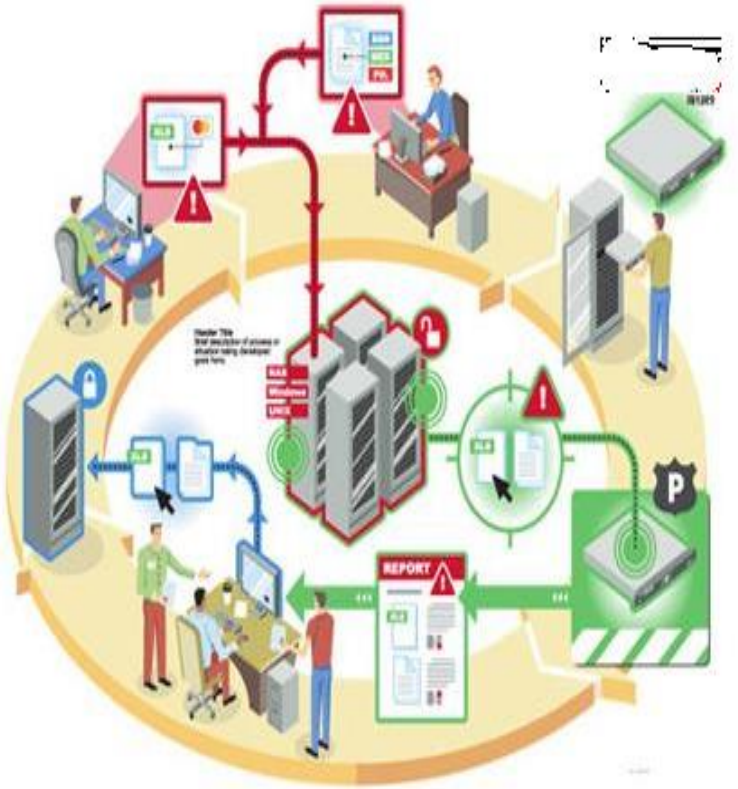
Aula 3: Segurança Adaptativa e SIEM 3.0

Aula 4: Segurança em Serviços em Cloud de IoT, IA, Blockchain e outros

Aula 5: Workshop das Atividades

Agenda

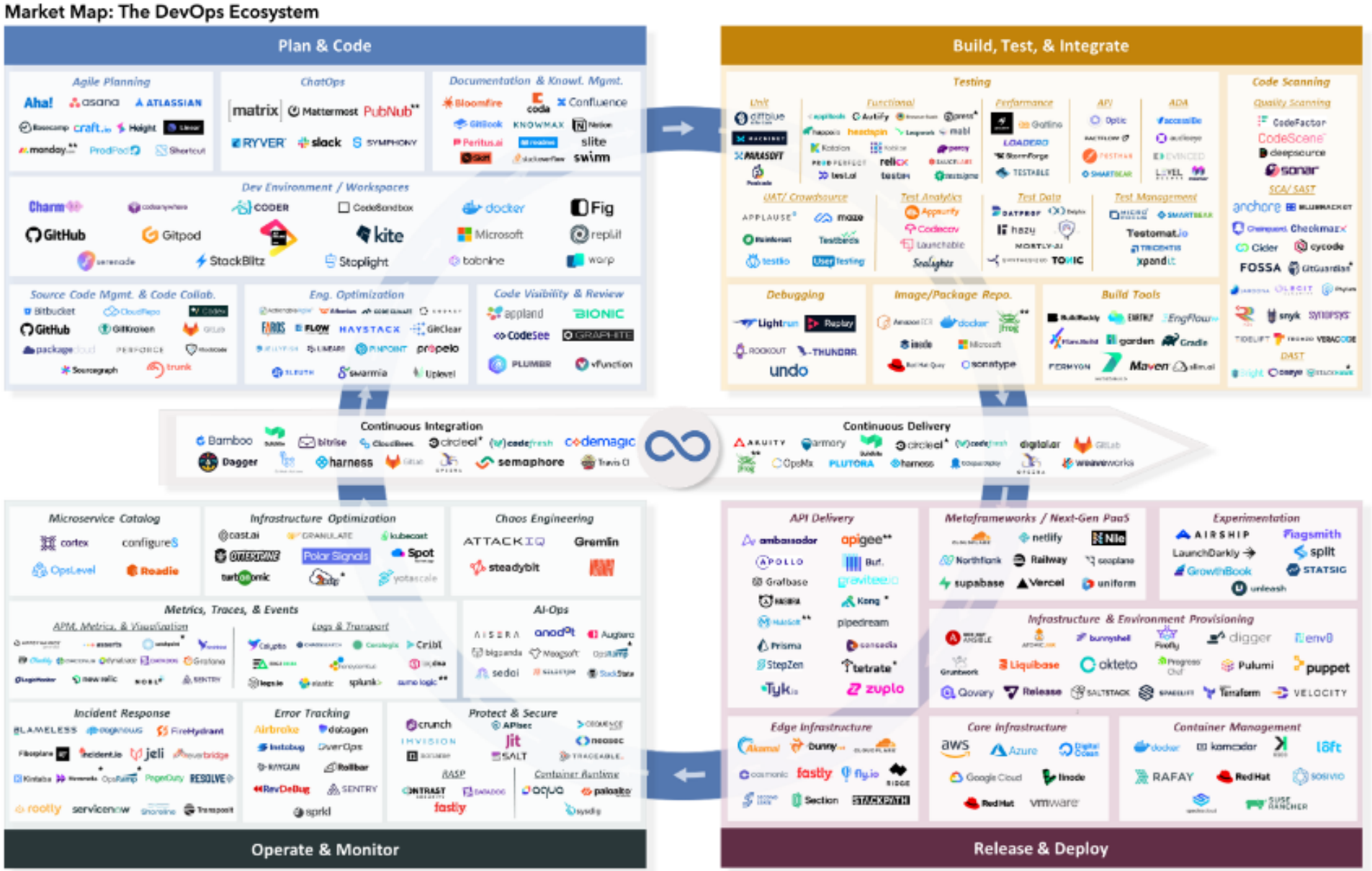
- Aula 1: 25/09/23
- Aula 2: 27/09/23
- Aula 3: 30/10/23 (M)
- Aula 4: 30/10/23 (T)
- Aula 5: 09/10/23



Acesso ao material de apoio:
https://www.dropbox.com/sh/br6gkgfg1301viw/AACL5onRJi5SRF5qRaIwnU6Ya?dl=0

- **Avaliação:**
  - Atividades:
    1. Tutoriais dos Serviços de Cloud das principais plataformas do mercado
    2. Tutoriais de DevSecOps
    3. Tutoriais de Segurança adaptativa e SIEM
    4. Apresentação dos principais fornecedores de SIEM para Cloud
  - Grupo: até 6 alunos

DevOps Landscape



June 2022 - Casber Wang (@casberw), Adam Liu (@adamdliu), David Carter. Note: Sapphire internal mapping, sorted by primary focus area. * = current portfolio company. ** = prior portfolio company.

SAPPHIRE

https://news.sapphireventures.com/hubfs/Sapphire%20Market%20Maps/7.19.22_Update_DEVOPS-MarketMap.png

Três Maneiras do Fluxo DevOps

- Fluxo: Fluxo de trabalho que vai do Desenvolvimento às Operações até o cliente. Maximizar esse fluxo é uma das chaves para o sucesso. As práticas incluídas no Flow são integração e implantação contínuas, limitação do trabalho em andamento, criação de ambientes sob demanda, automação etc.
- Feedback: Este é o fluxo de feedback rápido. Identificar problemas o mais rápido possível. Gerando qualidade e conhecimento. Práticas que ajudam no feedback são conjuntos de testes automatizados, compilações falhando no pipeline de implantação, monitoramento, etc.
- Experimentação e Aprendizagem Contínuas: Trata-se de criar uma cultura que promova a experimentação e aprendizagem contínuas. Isso é necessário para obter Flow e Feedback, mas também para mantê-los. As atividades que podem ajudar aqui são: criar uma cultura de inovação, construir confiança, alocar pelo menos 20% do tempo de Dev e Ops para requisitos não funcionais, etc.

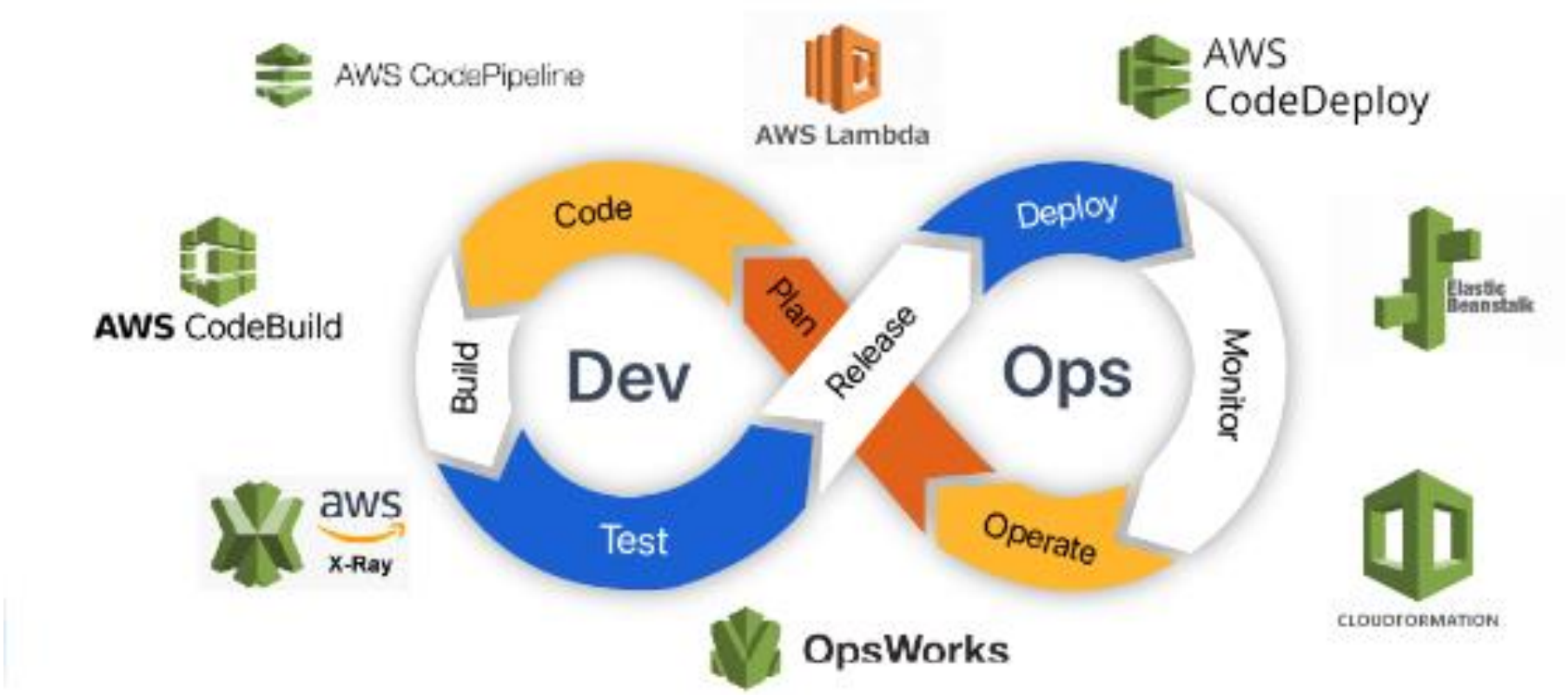
Contribuição nos Tipos de Trabalho de TI - DevOps

Tornar o trabalho visível é muito importante. Sem transparência, é difícil entender o que realmente está acontecendo e onde o tempo é gasto.

Os quatro tipos de trabalho são:

- Projetos de Negócios – Iniciativas de negócios, a maior parte do trabalho de desenvolvimento.
- Projetos internos de TI – Infraestrutura e Operações de TI. Criando novos ambientes, automatizando coisas, etc. Frequentemente não rastreados adequadamente. Isso cria problemas quando as operações já estão sob estresse.
- Atualizações e Alterações – Frequentemente geradas a partir dos dois tipos de trabalho anteriores. Atualizando e alterando diferentes sistemas.
- Trabalho não planejado ou trabalho de recuperação – Incidentes e problemas gerados por outros trabalhos. Isso dificulta a realização do trabalho planejado.

AWS DevOps


AWS CodePipeline
AWS Lambda
AWS CodeDeploy
AWS CodeBuild
Code
Plan
Build
Dev
Release
Deploy
Ops
Monitor
Test
Operate
aws X-Ray
Elastic Beanstalk
CLOUDFORMATION
OpsWorks

DevOps na AWS

DevOps na AWS:

- A AWS apresenta a mais ampla variedade de ofertas de serviços nas categorias PaaS, SaaS e IaaS, incluindo computação, gerenciamento de identidade e acesso (IAM), rede e armazenamento.
- AWS ofereça nuvens públicas, privadas e híbridas, seu foco está mais na nuvem pública.

Modelos de Computação em Nuvem

Infraestrutura como um serviço (IaaS)

A Infraestrutura como um serviço, às vezes abreviada como IaaS, contém os componentes básicos da TI em nuvem e, geralmente, dá acesso (virtual ou no hardware dedicado) a recursos de rede e computadores, como também espaço para o armazenamento de dados. A Infraestrutura como um serviço oferece a você o mais alto nível de flexibilidade e controle de gerenciamento sobre os seus recursos de TI e se assemelha bastante aos recursos de TI atuais com os quais muitos departamentos de TI e desenvolvedores estão familiarizados hoje em dia.

Plataforma como um serviço (PaaS):

Com a Plataforma como um serviço, as empresas não precisam mais gerenciar a infraestrutura subjacente (geralmente, hardware e sistemas operacionais), permitindo que você se concentre na implantação e no gerenciamento das suas aplicações. Isso o ajuda a tornar-se mais eficiente, pois elimina as suas preocupações com aquisição de recursos, planejamento de capacidade, manutenção de software, correção ou qualquer outro tipo de trabalho pesado semelhante envolvido na execução da sua aplicação.

Software como um serviço (SaaS)

O Software como um serviço oferece um produto completo, executado e gerenciado pelo provedor de serviços. Na maioria dos casos, as pessoas que se referem ao Software como um serviço estão se referindo às aplicações de usuário final. Com uma oferta de SaaS, não é necessário em como o serviço é mantido ou como a infraestrutura subjacente é gerenciada, você só precisa pensar em como usará este tipo específico de software. Um exemplo comum de aplicação do SaaS é o webmail, no qual você pode enviar e receber e-mails sem precisar gerenciar recursos adicionais para o produto de e-mail ou manter os servidores e sistemas operacionais no qual o programa de e-mail está sendo executado.

Metodologia ágil

- Com a unificação dos times, metódos como Kanban e Scrum são utilizados para garantir a qualidade nas entregas pensando na experiência do usuário.
- Utilizando o AWS Quick Start e o Atlassian Standard Infrastructure (ASI) implantamos o Jira Data Center na nuvem seguindo as práticas recomendadas da AWS e Atlassian.
- O Jira Software Data Center é utilizado por times ágeis como as equipes de desenvolvimento, e o Jira Service Management Data Center entrega gerenciamento de serviços de TI para equipes operacionais.

Infraestrutura como Código

- Processos repetitivos que demandam tempo dos times podem ser mapeados e automatizados.
- Com os serviços da AWS CloudFormation e AWS OpsWorks podemos modelar e automatizar utilizando a ferramenta chef. Além, é claro, de provisionar e gerenciar recursos.
- Com o AWS Systems Manager, podemos centralizar dados operacionais com apoio de serviços como Amazon CloudWatch, AWS CloudTrail e AWS Config ou ferramentas de terceiros, resolver problemas de aplicações automaticamente, implementar práticas recomendadas e corrigir eventos de segurança.

Entrega Contínua

O AWS CodePipeline, AWS CodeDeploy, AWS CodeStar e AWS CodeBuild ajudam os times nas esteiras de criação, teste e implementação automática, além de armazenar e controlar versões do código-fonte das aplicações de segurança.

AWS CodePipeline

- O AWS CodePipeline é um serviço de entrega contínua totalmente gerenciado que ajuda a automatizar pipelines de lançamento para atualizações rápidas e confiáveis de aplicativos e infraestrutura.
- O CodePipeline se integra a outros serviços da AWS, como CodeCommit, CodeBuild e CodeDeploy, para fornecer um fluxo de trabalho contínuo, desde alterações de código até implantação de produção.
- Com o CodePipeline, você pode automatizar os processos de criação, teste e implantação de seus aplicativos e implementar novos recursos e atualizações com segurança.

AWS CodePipeline

Casos de uso:

- Integração e entrega contínuas: o CodePipeline pode automatizar todo o processo de lançamento de software, desde a criação e teste de alterações de código até sua implantação na produção. Isso ajuda as equipes a fornecer novos recursos e atualizações aos clientes com mais rapidez e qualidade.
- Vários ambientes: o CodePipeline pode gerenciar vários ambientes, como desenvolvimento, preparação e produção. Cada ambiente pode ter seu pipeline, que pode ser configurado para implantar alterações de código automaticamente ou exigir aprovação manual antes da implantação.
- Implantações multirregionais: o CodePipeline também pode gerenciar implantações em diversas regiões, o que é especialmente útil para organizações ou aplicativos globais que exigem alta disponibilidade e baixa latência.
- Vários repositórios de origem: o CodePipeline pode ser integrado a vários repositórios de código-fonte, como GitHub, Bitbucket ou AWS CodeCommit. Isso permite que as equipes usem suas ferramentas preferidas e colaborem em alterações de código de diferentes fontes.
- Ações personalizadas: o CodePipeline oferece suporte a ações personalizadas, permitindo que as equipes ampliem o pipeline com suas ferramentas e processos. Por exemplo, as equipes podem adicionar uma ação personalizada para executar verificações de segurança nas alterações de código antes de implantá-las na produção.
- Monitoramento e notificações: o CodePipeline fornece monitoramento e notificações para execuções de pipeline, incluindo taxas de sucesso e falha, tempo de execução e logs. As equipes podem usar essas informações para identificar gargalos no pipeline e melhorar o desempenho.

AWS CodeBuild

- O AWS CodeBuild é um serviço de compilação totalmente gerenciado que compila seu código-fonte, executa testes de unidade e produz artefatos prontos para implantação.
- O CodeBuild se integra ao CodePipeline para fornecer uma solução completa de integração e entrega contínua (CI/CD).
- Com o CodeBuild, você pode dimensionar sua infraestrutura de construção sob demanda e pagar apenas pelos recursos de computação usados.
- O CodeBuild oferece suporte a uma ampla variedade de linguagens de programação, ferramentas de compilação e sistemas operacionais, para que você possa usar as ferramentas e estruturas que funcionam melhor para sua equipe.

AWS CodeBuild

Casos de uso:

- Integração contínua (CI): o AWS CodeBuild pode ser usado para fluxos de trabalho de integração contínua para criar e testar automaticamente alterações de código sempre que o código for enviado para um repositório.
- Entrega contínua (CD): o AWS CodeBuild também pode ser usado para fluxos de trabalho de entrega contínua para criar e empacotar aplicativos automaticamente, deixando-os prontos para implantação em ambientes de produção.
- Testes automatizados: o AWS CodeBuild pode executar testes automatizados como parte do processo de construção, fornecendo uma maneira consistente e confiável de testar alterações de código e detectar problemas antecipadamente.
- Criação de imagens Docker: o AWS CodeBuild pode criar imagens Docker automaticamente como parte do processo de compilação, facilitando o empacotamento e a implantação de aplicativos em contêineres.
- Construções multiplataforma: o AWS CodeBuild pode criar aplicativos para diversas plataformas, incluindo Linux, Windows e macOS, tornando-o uma ferramenta versátil para desenvolvedores e equipes que trabalham em diversos projetos.
- Ambientes de construção personalizados: o AWS CodeBuild permite que os usuários criem ambientes personalizados com configurações, dependências e ferramentas específicas, possibilitando construir e testar aplicativos com requisitos específicos.
- Otimização de custos: o AWS CodeBuild pode ajudar as equipes a otimizar seus custos de construção cobrando apenas pelos recursos usados durante o processo de construção, eliminando a necessidade de servidores de construção dedicados e reduzindo custos indiretos.

AWS CodeDeploy

- AWS CodeDeploy é um serviço de implantação totalmente gerenciado que automatiza implantações de software em vários serviços de computação, como Amazon EC2, AWS Fargate, AWS Lambda e servidores locais.
- O AWS CodeDeploy integra-se ao CodePipeline e ao CodeBuild para fornecer uma solução completa de CI/CD.
- Com o CodeDeploy, você pode implantar facilmente seus aplicativos, reverter implantações rapidamente em caso de erros e garantir que seus aplicativos estejam funcionando de maneira uniforme e confiável.

AWS CodeDeploy

Casos de uso:

- Implantações azuis/verdes: o AWS CodeDeploy é comumente usado para implantações azuis/verdes, o que permite liberações com tempo de inatividade zero, roteando o tráfego entre dois ambientes idênticos, um ativo e outro inativo, e invertendo o tráfego assim que a nova implantação for bem-sucedida.
- Implantações automatizadas: o AWS CodeDeploy pode automatizar o processo de implantação de aplicativos, permitindo que as equipes implantem de forma rápida e confiável novas versões de seus aplicativos em ambientes de produção.
- Reversão: o AWS CodeDeploy permite que as equipes revertam para uma versão anterior do aplicativo em caso de problemas ou erros na implantação mais recente.
- Implantação em várias instâncias: o AWS CodeDeploy pode implantar aplicativos simultaneamente em várias instâncias, possibilitando dimensionar implantações para oferecer suporte a aplicativos de alto tráfego.
- Implantação em diversas regiões: o AWS CodeDeploy pode implantar aplicações em diversas regiões, tornando a implantação de aplicações globalmente acessível e garantindo alta disponibilidade e tolerância a falhas.
- Configurações de implantação personalizadas: o AWS CodeDeploy permite que os usuários definam configurações de implantação personalizadas, incluindo ganchos pré e pós-implantação, possibilitando personalizar o processo de implantação para atender a requisitos específicos.
- Integração com outros serviços da AWS: o AWS CodeDeploy integra-se a outros serviços da AWS, como AWS CodePipeline, AWS CloudFormation e AWS Elastic Beanstalk, possibilitando automatizar e personalizar todo o pipeline de implantação de aplicativos.

AWS CloudFormation

- O AWS CloudFormation é um serviço que ajuda a modelar e provisionar recursos da AWS, automatizar a implantação de infraestrutura e gerenciar a infraestrutura como código. Com o CloudFormation, você pode definir sua infraestrutura como um modelo e implantá-la com um único comando.
- O AWS CloudFormation oferece suporte a uma ampla variedade de recursos da AWS e você pode estendê-lo com recursos personalizados usando o AWS Lambda.
- CloudFormation se integra a outros serviços AWS, como CodePipeline e CodeBuild, para fornecer uma solução DevOps completa.

AWS CloudFormation

Casos de uso:

- Infraestrutura como código: o AWS CloudFormation permite que desenvolvedores e equipes definam sua infraestrutura como código, facilitando o gerenciamento, o controle de versão e a replicação da infraestrutura em ambientes e regiões.
- Provisionamento automatizado: o AWS CloudFormation pode automatizar o provisionamento de recursos e serviços da AWS, permitindo que as equipes configurem e gerenciem infraestruturas complexas rapidamente.
- Fácil gerenciamento de alterações: o AWS CloudFormation facilita o gerenciamento de alterações de infraestrutura, permitindo que as equipes definam alterações de código, revisem-nas no controle de versão e apliquem-nas por meio de automação.
- Consistência em todos os ambientes: o AWS CloudFormation permite que as equipes garantam consistência em ambientes de desenvolvimento, preparação e produção, definindo a infraestrutura como código e aplicando o mesmo código a diferentes ambientes.
- Integração com ferramentas DevOps: O AWS CloudFormation integra-se com ferramentas DevOps como AWS CodePipeline e AWS CodeCommit, possibilitando automatizar todo o pipeline de entrega de software.
- Criação de recursos personalizados: o AWS CloudFormation permite que as equipes criem recursos personalizados indisponíveis nativamente na AWS, possibilitando estender a plataforma e integrá-la a outros serviços.
- Implantações multirregionais: o AWS CloudFormation pode implantar infraestrutura em diversas regiões, possibilitando a criação de arquiteturas altamente disponíveis e tolerantes a falhas.

AWS Elastic Beanstalk

- O AWS Elastic Beanstalk é um serviço totalmente gerenciado que facilita a implantação e o gerenciamento de aplicações na Nuvem AWS.
- O AWS Elastic Beanstalk gerencia automaticamente a implantação, a escalabilidade e o monitoramento de suas aplicações para que você possa se concentrar na escrita do código.
- O Elastic Beanstalk oferece suporte a uma ampla variedade de linguagens de programação, plataformas e estruturas e se integra a outros serviços da AWS, como CodePipeline e CodeBuild, para fornecer uma solução completa de CI/CD.

AWS Elastic Beanstalk

Casos de uso:

- Implantação rápida de aplicativos: o AWS Elastic Beanstalk simplifica o processo de implantação de aplicativos, fornecendo uma plataforma totalmente gerenciada que gerencia automaticamente o provisionamento, a implantação e o dimensionamento de aplicativos.
- Suporte multiplataforma: o AWS Elastic Beanstalk oferece suporte a uma ampla variedade de plataformas, incluindo Java, Python, .NET, Ruby, Node.js, Go e PHP, tornando-o uma plataforma versátil para desenvolvedores que trabalham em diversos projetos.
- Ambientes personalizáveis: o AWS Elastic Beanstalk permite que os desenvolvedores personalizem o ambiente em que seus aplicativos são executados, definindo variáveis de ambiente, configurando grupos de segurança e especificando limites de recursos.
- Escalonamento automático: o AWS Elastic Beanstalk aumenta ou diminui automaticamente os recursos com base na demanda da aplicação, possibilitando o suporte a aplicações de alto tráfego sem intervenção manual.
- Integração com ferramentas DevOps: o AWS Elastic Beanstalk integra-se com ferramentas DevOps como AWS CodePipeline e AWS CodeCommit, possibilitando automatizar todo o pipeline de entrega de software.
- Monitoramento e registro em log: o AWS Elastic Beanstalk fornece recursos integrados de monitoramento e registro em log, facilitando o monitoramento do desempenho de aplicativos, a solução de problemas e a análise de logs.
- Otimização de custos: o AWS Elastic Beanstalk pode ajudar as equipes a otimizar seus custos cobrando apenas pelos recursos usados pela aplicação, eliminando a necessidade de infraestrutura dedicada e reduzindo custos indiretos.

AWS OpsWorks

- AWS OpsWorks é um serviço de gerenciamento de configuração que ajuda a automatizar a implantação e o gerenciamento de aplicativos e infraestrutura.
- O AWS OpsWorks fornece modelos para pilhas de aplicativos populares e oferece suporte a configurações personalizadas usando o Chef, uma ferramenta popular de gerenciamento de configuração.
- O OpsWorks se integra a outros serviços da AWS, como CodePipeline e CodeDeploy, para fornecer uma solução DevOps completa.

AWS OpsWorks

Casos de uso:

- Gerenciamento automatizado de aplicativos: o AWS OpsWorks automatiza o gerenciamento de aplicativos, fornecendo uma plataforma totalmente gerenciada que lida com implantação, escalabilidade, monitoramento e manutenção de aplicativos.
- Gerenciamento de configuração baseado em Chef: o AWS OpsWorks é baseado na plataforma de automação Chef, possibilitando definir infraestrutura como código e aplicar alterações por meio de automação.
- Suporte multiplataforma: o AWS OpsWorks oferece suporte a muitas plataformas, incluindo Linux, Windows e macOS, tornando-o uma plataforma versátil para desenvolvedores e equipes que trabalham em diversos projetos.
- Ambientes personalizáveis: o AWS OpsWorks permite que os desenvolvedores personalizem o ambiente no qual seus aplicativos são executados, definindo receitas personalizadas, configurando grupos de segurança e especificando limites de recursos.
- Escalonamento automático: o AWS OpsWorks aumenta ou diminui automaticamente os recursos com base na demanda da aplicação, possibilitando o suporte a aplicações de alto tráfego sem intervenção manual.
- Integração com ferramentas DevOps: o AWS OpsWorks se integra a ferramentas DevOps como AWS CodePipeline e AWS CodeCommit, possibilitando automatizar todo o pipeline de entrega de software.
- Monitoramento e registro em log: o AWS OpsWorks fornece recursos integrados de monitoramento e registro em log, facilitando o monitoramento do desempenho de aplicativos, a solução de problemas e a análise de logs.
- Otimização de custos: o AWS OpsWorks pode ajudar as equipes a otimizar seus custos cobrando apenas pelos recursos usados pela aplicação, eliminando a necessidade de infraestrutura dedicada e reduzindo custos indiretos.

AWS Lambda

- AWS Lambda é um serviço de computação sem servidor que permite executar código sem provisionar ou gerenciar servidores. As funções Lambda podem ser acionadas por eventos de outros serviços da AWS, como S3, DynamoDB ou API Gateway, e podem ser escritas em várias linguagens de programação, como Python, Node.js ou Java.
- O Lambda se integra a outros serviços da AWS, como CodePipeline e CodeDeploy, para fornecer uma solução DevOps completa sem servidor.

AWS Lambda

Casos de uso:

- Infraestrutura sem servidor: o AWS Lambda permite que as equipes criem aplicativos sem servidor, eliminando a necessidade de infraestrutura dedicada e reduzindo custos indiretos.
- Arquitetura orientada a eventos: o AWS Lambda é ideal para arquiteturas orientadas a eventos, permitindo que as equipes acionem a execução de código em resposta a eventos de outros serviços da AWS, como Amazon S3, Amazon DynamoDB e Amazon Kinesis.
- Testes automatizados: o AWS Lambda pode automatizar testes acionando testes em resposta a confirmações de código ou alterações em outros serviços da AWS, possibilitando testar e implantar alterações de código de forma rápida e confiável.
- Integração e implantação contínuas: o AWS Lambda se integra a ferramentas DevOps como AWS CodePipeline e AWS CodeCommit, possibilitando automatizar todo o pipeline de entrega de software e implantar alterações de código continuamente.
- Integrações personalizadas: o AWS Lambda permite que as equipes criem integrações personalizadas com outros serviços da AWS ou APIs de terceiros, possibilitando estender a funcionalidade de aplicativos existentes ou criar novos aplicativos.
- Otimização de custos: o AWS Lambda pode ajudar as equipes a otimizar seus custos cobrando apenas pelo tempo de computação usado pelo código, eliminando a necessidade de infraestrutura dedicada e reduzindo custos indiretos.

AWS X-Ray

- AWS X-Ray é uma ferramenta de depuração e análise de desempenho que ajuda a depurar e analisar aplicações e microsserviços.
- O AWS X-Ray oferece visibilidade completa das solicitações à medida que elas passam pela sua aplicação.

AWS X-Ray

Casos de uso:

- Rastreamento distribuído: o AWS X-Ray permite que as equipes rastreiem solicitações à medida que elas fluem por arquiteturas complexas de microsserviços, facilitando a identificação de gargalos de desempenho, a solução de problemas e a otimização do desempenho de aplicativos.
- Visualização do mapa de serviços: o AWS X-Ray fornece uma representação visual dos serviços e dependências de uma aplicação, possibilitando entender as interações entre componentes e identificar áreas para otimização.
- Integração com ferramentas DevOps: o AWS X-Ray se integra a ferramentas DevOps como AWS CodePipeline e AWS CodeCommit, possibilitando automatizar todo o pipeline de entrega de software e monitorar o desempenho do aplicativo em tempo real.
- Anotações e metadados personalizados: o AWS X-Ray permite que as equipes adicionem anotações e metadados personalizados aos dados de rastreamento, possibilitando adicionar contexto aos rastreamentos e enriquecer os dados com informações específicas do negócio.
- Detecção de anomalias: o AWS X-Ray inclui recursos integrados de detecção de anomalias, possibilitando identificar picos incomuns no desempenho do aplicativo ou erros que afetam a experiência do usuário.
- Otimização de custos: o AWS X-Ray pode ajudar as equipes a otimizar seus custos cobrando apenas pelos rastreamentos coletados e pelos dados analisados, eliminando a necessidade de infraestrutura dedicada e reduzindo custos indiretos.
- Conformidade e segurança: o AWS X-Ray integra-se ao AWS CloudTrail para fornecer uma trilha de auditoria completa de atividades em toda a pilha de aplicativos, possibilitando o atendimento aos requisitos de conformidade e segurança.

Monitoramento e Registro de Log

Com o Amazon CloudWatch, AWS CloudTrail e o AWS X-Ray podemos monitorar recursos na nuvem, registrar chamadas de API e analisar e depurar aplicações, entendendo aplicativos e solucionando a causa raiz dos problemas.

Microsserviços:

Utilizando Amazon Elastic Container Service e o AWS Lambda, podemos gerenciar containers e execução de códigos sem provisionamento ou gerenciamento de servidores.

Controle de Versões

Com o serviço da AWS CodeCommit, hospedamos repositórios Gits seguros e escaláveis.

PaaS

- Faça upload do seu código sem se preocupar com a implementação através do AWS Elastic Beanstalk.

- Por fim, caso ainda não utilize a infraestrutura como código, a dica é migrar suas cargas de trabalho para a nuvem. Se você busca segurança, agilidade, escalabilidade, rápida entrega e implementação do método CI/CD de forma fácil e rápida, conte com estes e demais serviços da AWS e inicie sua jornada DevOps acelerando e otimizando entregas.

DevOps na AWS

- Ferramentas do desenvolvedor da AWS, ajudam a armazenar e a fazer o controle de versão do código-fonte de aplicações com segurança, além de criar, testar e implantar automaticamente aplicações no seu ambiente, local ou da AWS.

Fluxos de trabalho de lançamento de software
AWS CodePipeline

O AWS CodePipeline é um serviço de integração contínua e entrega contínua para atualizações rápidas e confiáveis de aplicações e infraestruturas. O CodePipeline cria, testa e implanta código sempre que ocorrer uma alteração de código, de acordo com modelos de processo de lançamento definidos. Isso permite disponibilizar recursos e atualizações de forma rápida e confiável.

Crie e teste código
AWS CodeBuild

O AWS CodeBuild é um serviço totalmente gerenciado que compila o código-fonte, roda testes e produz pacotes de software prontos para implantação. Com o CodeBuild, você não precisa provisionar, gerenciar e escalar seus próprios servidores do build. O CodeBuild escala continuamente e processa múltiplas compilações ao mesmo tempo, o que evita que elas fiquem esperando em uma fila.

Automação da implantação
AWS CodeDeploy

O AWS CodeDeploy automatiza as implantações de código para qualquer instância, inclusive instâncias do Amazon EC2 e servidores locais. O AWS CodeDeploy facilita o lançamento rápido de novos recursos, ajuda a evitar tempo de inatividade durante a implantação de aplicativos e lida com a complexidade de atualizá-los.

Projetos unificados de CI/CD
AWS CodeStar

O AWS CodeStar permite que você desenvolva, crie e implante rapidamente aplicações na AWS. O AWS CodeStar disponibiliza uma interface de usuário unificada, permitindo que você gerencie facilmente atividades de desenvolvimento de software em um só lugar. Com o AWS CodeStar, é possível configurar toda a sua cadeia de ferramentas de entrega contínua em alguns minutos, possibilitando que você comece a agilizar o lançamento de código.

https://aws.amazon.com/pt/devops/

DevOps na AWS

Microsserviços

Crie e implante uma arquitetura de microsserviços usando contêineres ou computação sem servidor.

Plataforma de produção do Docker

Amazon Elastic Container Service

O Amazon Elastic Container Service (ECS) é um serviço de gerenciamento de contêineres altamente escalável e de alto desempenho com suporte a contêineres do Docker, o que permite executar facilmente aplicativos em um cluster gerenciado de instâncias do Amazon EC2.

Computação sem servidor

AWS Lambda

O AWS Lambda permite que você execute códigos sem provisionar ou gerenciar servidores. Com o Lambda, você pode executar o código para praticamente qualquer tipo de aplicativo ou serviço de back-end, tudo sem precisar de administração. Basta carregar o código e o Lambda toma conta de tudo o que for necessário para executar e alterar a escala do seu código com alta disponibilidade.

https://aws.amazon.com/pt/devops/

DevOps na AWS

Infraestrutura como código

Provisione, configure e gerencie os recursos da sua infraestrutura da AWS usando código e modelos. Monitore e aplique a conformidade da infraestrutura.

Provisionamento da infraestrutura de modelos

AWS CloudFormation

O AWS CloudFormation oferece aos desenvolvedores e administradores de sistemas uma maneira fácil de criar e gerenciar um grupo de recursos relacionados à AWS, provisionando e atualizando-os de forma organizada e previsível. Você pode usar os exemplos de modelos do AWS CloudFormation ou criar seus próprios modelos.

Gerenciamento de configuração do Chef

AWS OpsWorks

O AWS OpsWorks é um serviço de gerenciamento de configurações que usa o Chef, uma plataforma de automação que trata configurações de servidor como código. O OpsWorks usa o Chef para automatizar a forma como os servidores são configurados, implantados e gerenciados em instâncias do Amazon Elastic Compute Cloud (Amazon EC2) ou em ambientes de computação locais. O OpsWorks oferece dois serviços, o AWS OpsWorks for Chef Automate e o AWS OpsWorks Stacks.

https://aws.amazon.com/pt/devops/

DevOps na AWS

Gerenciamento de configuração

AWS Systems Manager

O AWS Systems Manager é um serviço de gerenciamento que ajuda a coletar inventário de software, aplicar patches em sistemas operacionais, criar imagens de sistemas e configurar os sistemas operacionais Windows e Linux, tudo isso de forma automática. Esses recursos ajudam a definir e rastrear configurações de sistema, evitar desvios e manter a conformidade do software em configurações do EC2 e locais.

Política como código

AWS Config

O AWS Config é um serviço totalmente gerenciado que oferece inventário de recursos, histórico de configuração e notificações de alteração de configuração da AWS para proporcionar segurança e governança. O Config Rules permite que você crie regras que verificam automaticamente a configuração de recursos da AWS gravada pelo AWS Config.

https://aws.amazon.com/pt/devops/

Práticas de DevOps na AWS

Treinamento e certificação

Introdução à AWS
Veja nosso conjunto de vídeos e laboratórios gratuitos para ajudá-lo a começar a usar o serviços de DevOps.

Caminho de aprendizagem do desenvolvedor
Veja todos os nossos cursos de capacitação e laboratórios autoguiados que ensinam você a desenvolver aplicações de nuvem na AWS, além de orientá-lo durante a obtenção do certificado AWS Certified DevOps Engineer – Professional.

Caminho de aprendizagem do Operations
Veja todos os nossos cursos de capacitação e laboratórios autoguiados que ensinam você a criar implantações automáticas e repetidas de aplicações, redes e sistemas na AWS. Siga essa jornada em direção à obtenção do certificado AWS Certified DevOps Engineer – Professional.

Guias, tutoriais e whitepapers

Guias e tutoriais

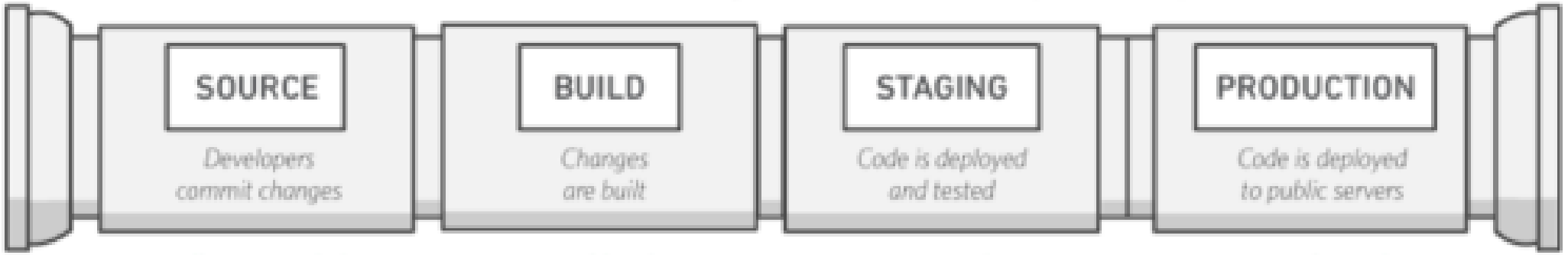
Guia de conceitos básicos sobre a AWS para DevOps
Este guia contêm informações sobre os serviços da AWS para DevOps e uma demonstração que você pode usar para testar esses serviços.

Tutorial de 10 minutos: implante código em uma máquina virtual

Tutorial de 10 minutos: configure um pipeline de implantação contínua

https://aws.amazon.com/pt/devops/resources/

Implementando a Integração Contínua e Entrega Contínua CI/CD na AWS



CI/CD pipeline

Caminho para a integração contínua e entrega contínua

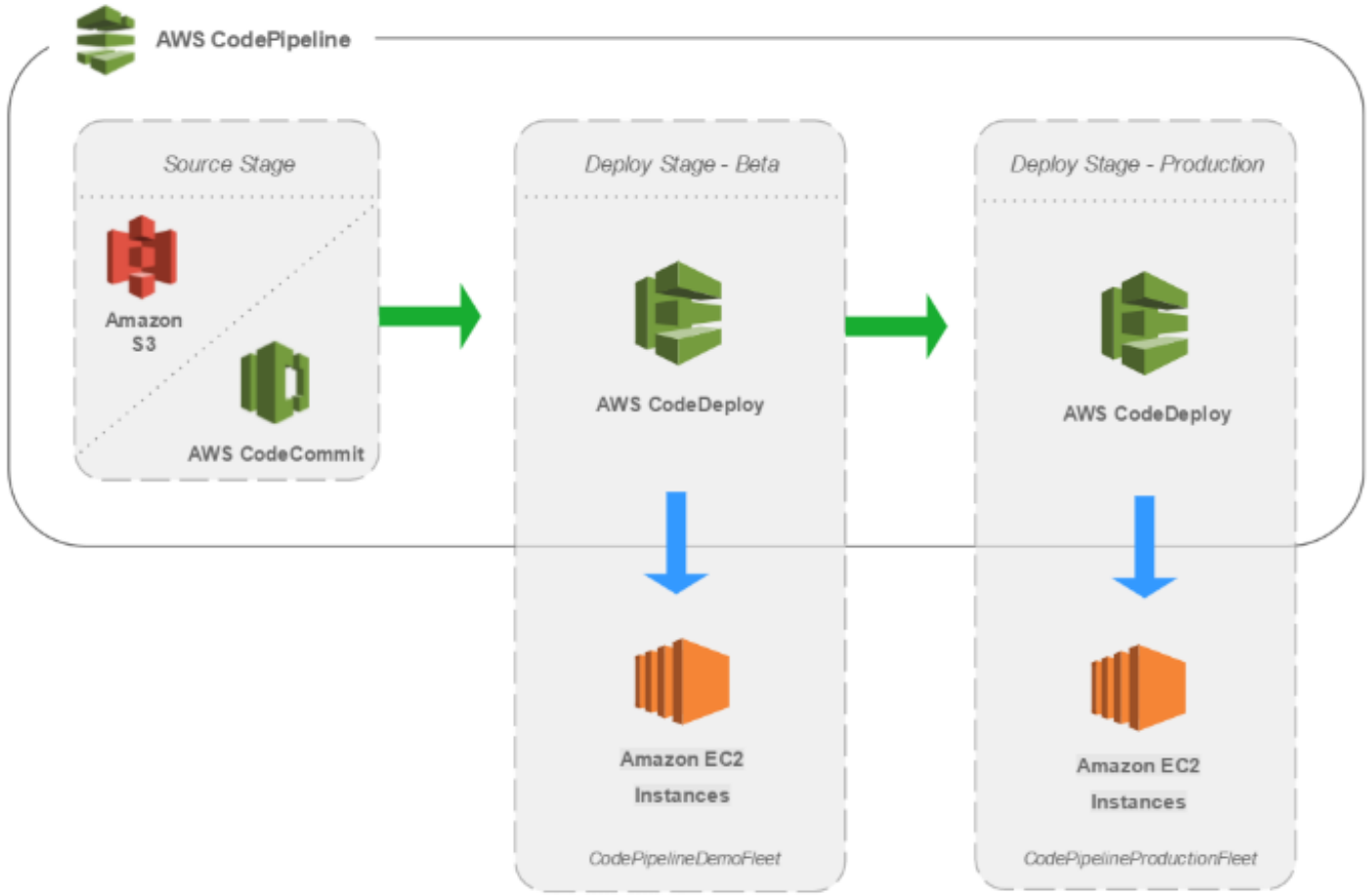
- Novo código é enviado em uma extremidade, testado em uma série de estágios (fonte, compilação, preparo e produção) e, em seguida, publicado como código pronto para produção.
- Se a organização é nova em CI/CD, ela pode abordar esse pipeline de forma iterativa.
- Isso significa que você deve começar pequeno e interagir em cada estágio, para poder desenvolver o código de uma forma que acompanhe o crescimento da organização.

Integração contínua na AWS

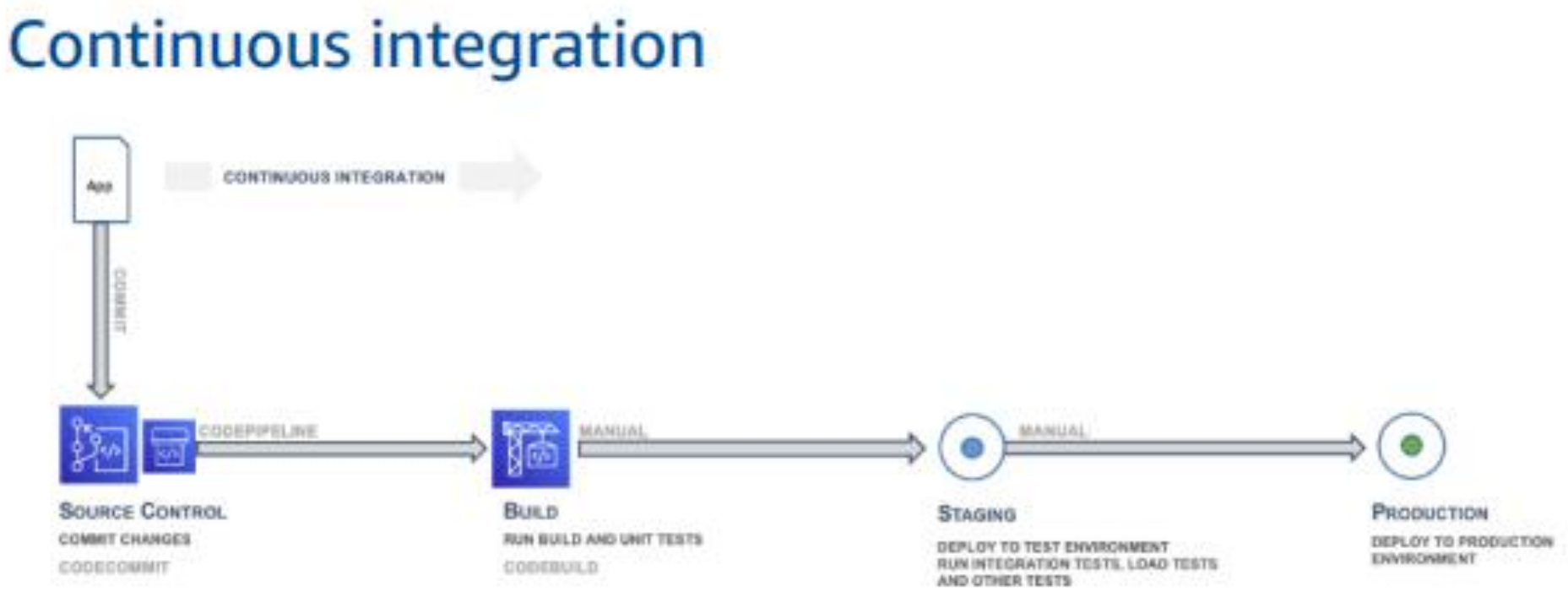
Integração contínua — Código fonte e build:

- A primeira fase da jornada CI/CD é desenvolver a maturidade na integração contínua.
- Certificar que todos os desenvolvedores regularmente enviem seu código para um repositório central (como um hospedado no CodeCommit ou GitHub) e mesclar todas as alterações em um pipeline de lançamento para o aplicativo.
- Nenhum desenvolvedor deve manter o código isoladamente. Se uma ramificação de recurso for necessária por um determinado período de tempo, deve manter-se atualizado, mesclando-se a partir do upstream, sempre que possível.
- Frequentes commits e merges completas de trabalho são recomendadas para que a equipe desenvolva disciplina e são incentivadas pelo processo.
- Um desenvolvedor que mescla o código cedo e com frequência provavelmente terá menos problemas de integração no futuro.

CI/CD Pipeline na AWS



Continuous Integration na AWS

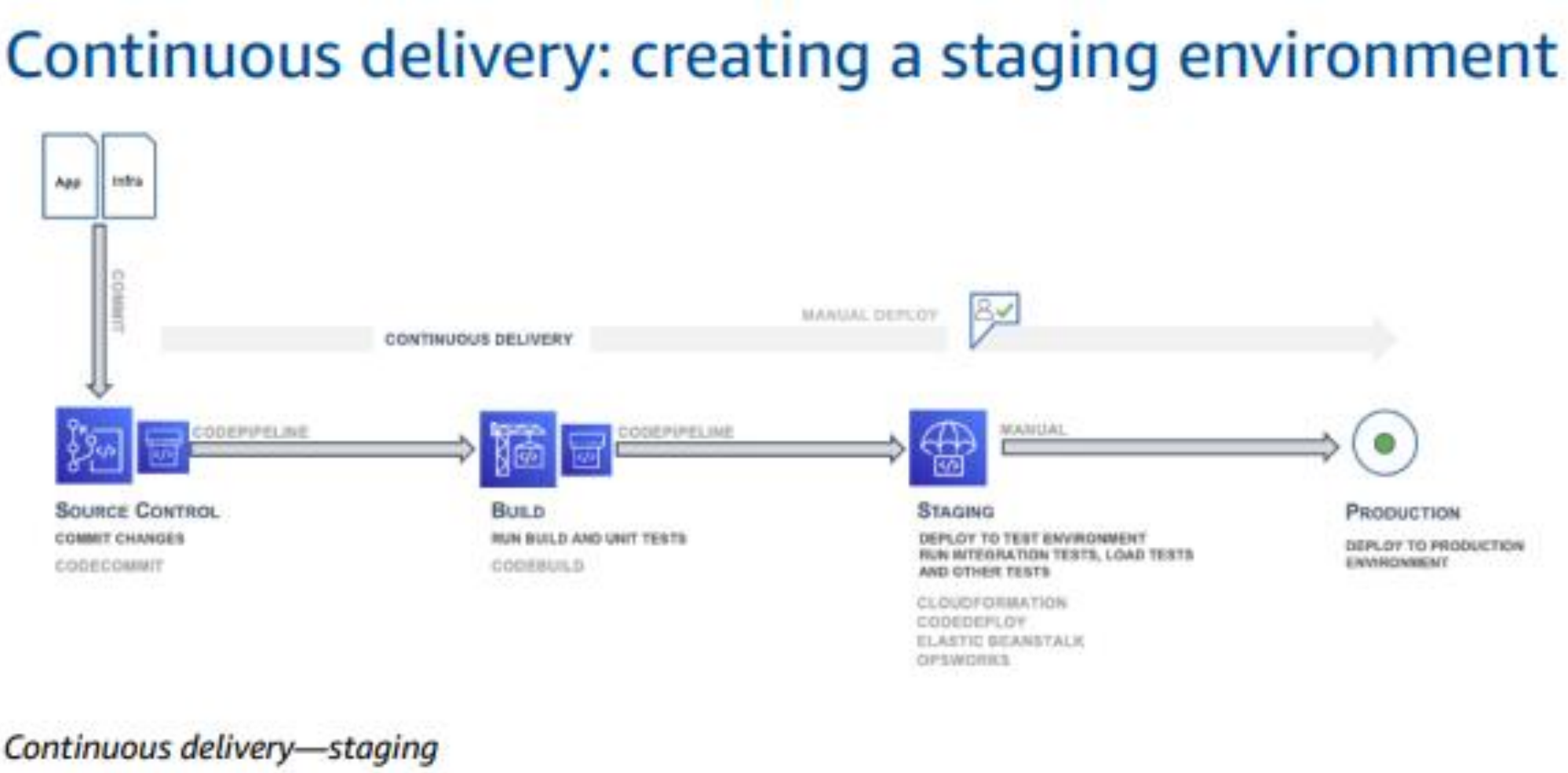


Continuous integration—source and build

Entrega Contínua - Ambiente de Preparação - AWS

- A entrega contínua (CD) é a próxima fase e envolve a implantação do código do aplicativo em um ambiente de preparação, que é uma réplica das camadas de produção, e executando testes mais funcionais.
- O ambiente de preparação pode ser um ambiente estático pré-fabricado para teste, ou você pode provisionar e configurar um ambiente dinâmico com infraestrutura comprometida e código de configuração para teste e implantação do código do aplicativo.

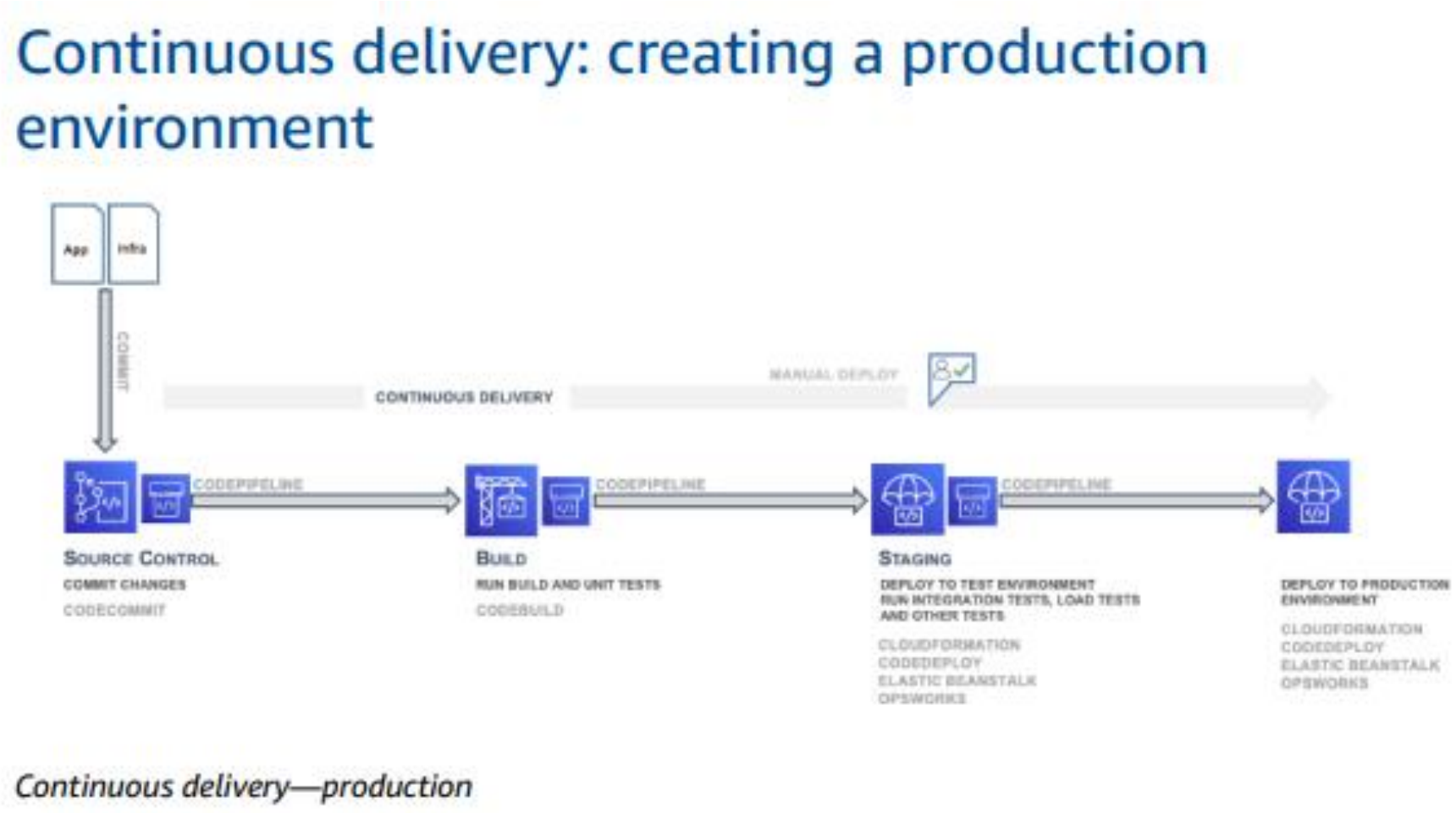
Entrega Contínua - Ambiente de Preparação - AWS



Continuous delivery—staging

Entrega Contínua - Ambiente de Preparação - AWS

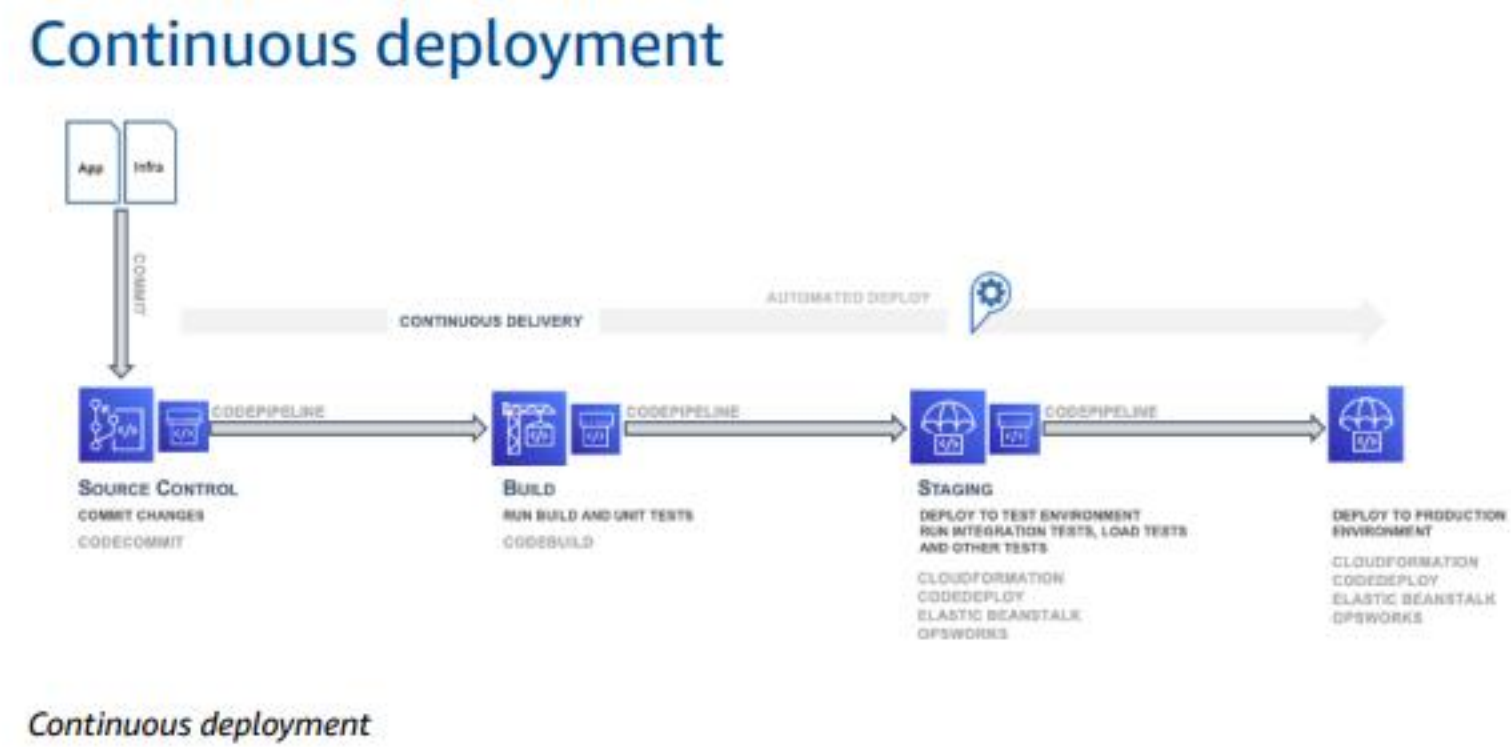
Na sequência do pipeline de implantação após o ambiente de preparação, está a o ambiente de produção, que também é construído usando infraestrutura como código (IaC).



Continuous delivery—production

Entrega Contínua na AWS

- A fase final no pipeline de implantação de CI/CD é a implantação contínua, que pode incluir automação de todo o processo de lançamento de software, incluindo a implantação no ambiente de produção.
- Em um ambiente de CI/CD totalmente maduro, o caminho para o ambiente de produção é totalmente automatizado, o que permite que o código seja implantado com alta confiança.

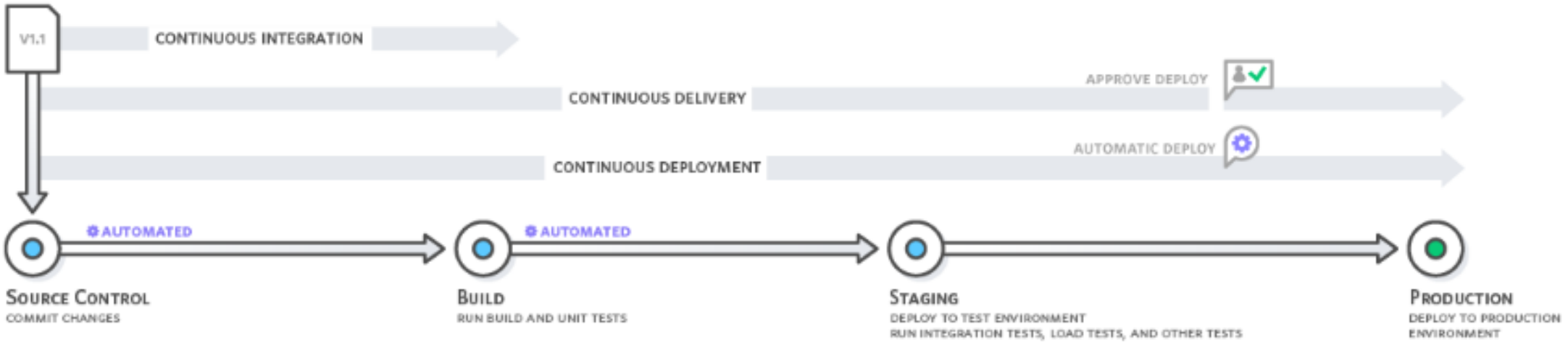


Continuous deployment

Opções de Provisionamento e Implantação na AWS

- Amazon Web Services (AWS) oferece várias opções para provisionamento de infraestrutura e implantação dos aplicativos.
- Quer a arquitetura do seu aplicativo seja um aplicativo web simples de três camadas ou um conjunto complexo de cargas de trabalho, a AWS oferece serviços de implantação para atender aos requisitos do aplicativo e da organização.

Distribuição Contínua x Implantação Contínua



- Com a distribuição contínua, cada alteração de código é criada, testada e enviada para um ambiente de teste ou preparação, que não pertence à produção. É possível que existam vários estágios de teste paralelos antes de uma ordem de produção ser implantada. A diferença entre entrega contínua e implantação contínua é a presença de uma aprovação manual para atualizar o ambiente de produção. Com a implantação contínua, a atualização da produção ocorre automaticamente, sem aprovação explícita.
- A distribuição contínua automatiza o processo de lançamento de software completo. Cada revisão confirmada aciona um fluxo automático que cria, testa e prepara a atualização. A decisão final de implantar em um ambiente de produção ativo é acionada pelo desenvolvedor.

AWS CloudFormation

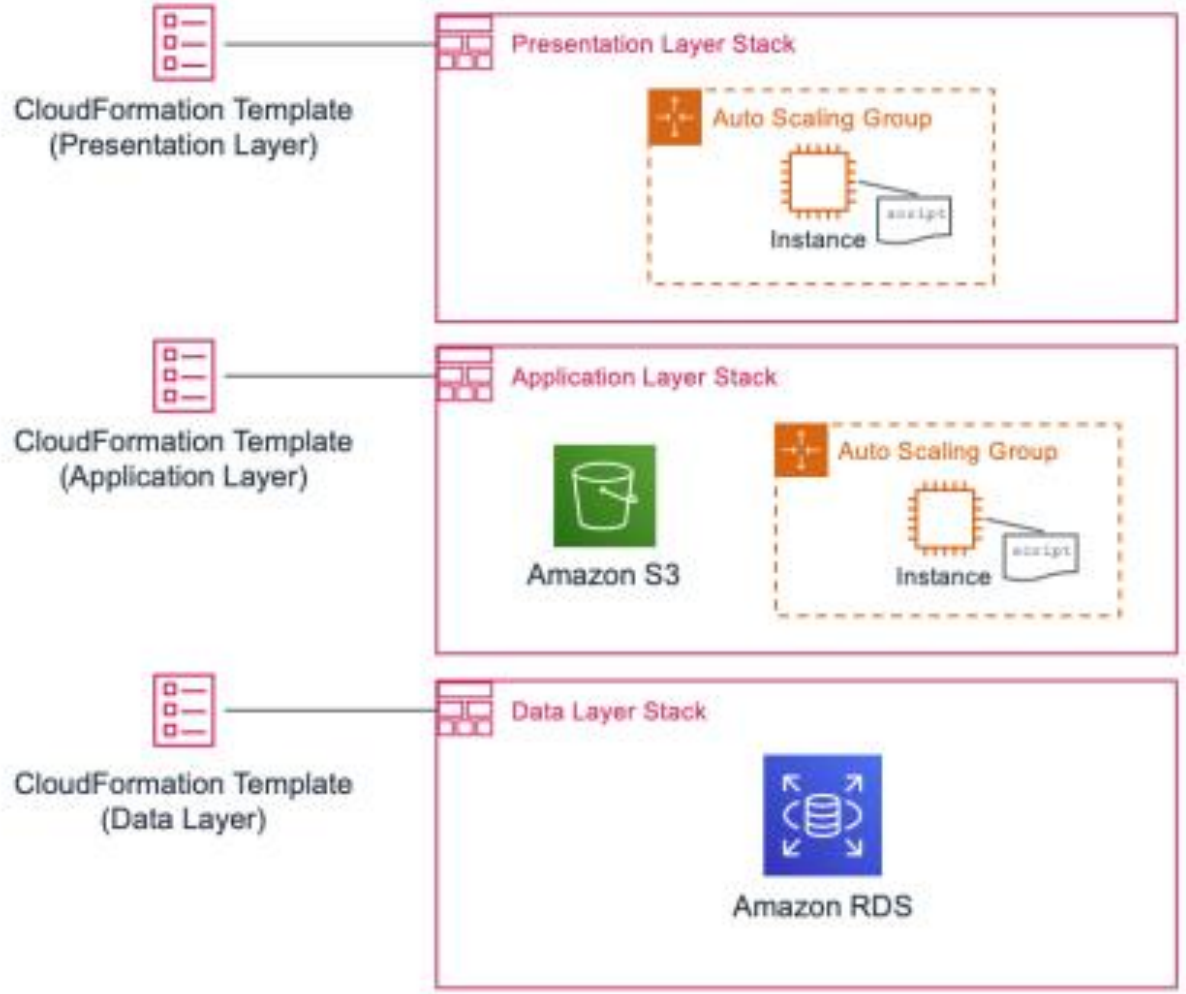
- Serviço que permite aos clientes provisionar e gerenciar praticamente qualquer recurso AWS usando uma linguagem de modelo personalizada expressa em YAML ou JSON.
- Um modelo do CloudFormation cria recursos de infraestrutura em um grupo chamado “stack” e permite que você defina e personalize todos os componentes necessários para operar seu aplicativo, mantendo o controle total desses recursos.
- O uso de templates possibilita a implementação do controle de versão na infraestrutura e na capacidade de replicar a infraestrutura de forma rápida e confiável.

AWS CloudFormation

A AWS oferece extensões para o serviço CloudFormation, além de seus recursos básicos:

- AWS Cloud Development Kit (AWS CDK) (AWS CDK) é um kit de desenvolvimento de software (SDK) de código aberto para modelar a infraestrutura da AWS, com TypeScript, Python, Java ou .NET.
- AWS Serverless Application Model (SAM) é uma estrutura de código aberto para simplificar a criação de aplicativos serveless na AWS.

Utilização do AWS CloudFormation

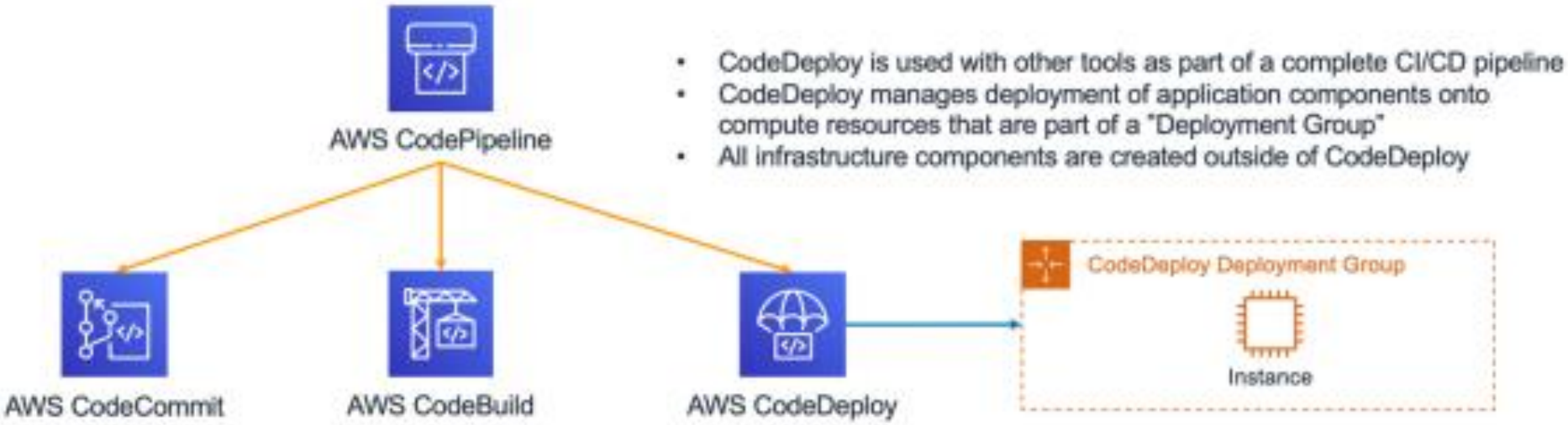


- CloudFormation used to create all infrastructure components including IAM Roles, VPCs, Subnets, Route Tables, Security Groups, S3 bucket policies, etc.
- Separate CloudFormation templates used to build each "domain" of the application architecture

AWS CodeDeploy

- O AWS CodeDeploy é um serviço de implantação totalmente gerenciado que automatiza as implantações de aplicativos para serviços de computação como Amazon EC2, Amazon Elastic Container Service (Amazon ECS), AWS Lambda, ou servidores locais.
- As empresas podem usar o CodeDeploy para automatizar implantações de um aplicativo e remover as operações manuais propensas a erros do processo de implantação.
- O CodeDeploy pode ser usado com uma ampla variedade de conteúdo de aplicativos, incluindo código, funções servelss, arquivos de configuração e outros.
- O CodeDeploy deve ser usado como um serviço de “bloco de construção” focado em ajudar os desenvolvedores de aplicativos para implantarem e atualizarem o software que está sendo executado na infraestrutura existente.
- Destina-se a ser usado em conjunto com outras implantações de serviços AWS como AWS CodeStar, AWS CodePipeline, outras ferramentas de desenvolvedor da AWS e serviços de terceiros.

CodeDeploy como parte de um completo CI/CD

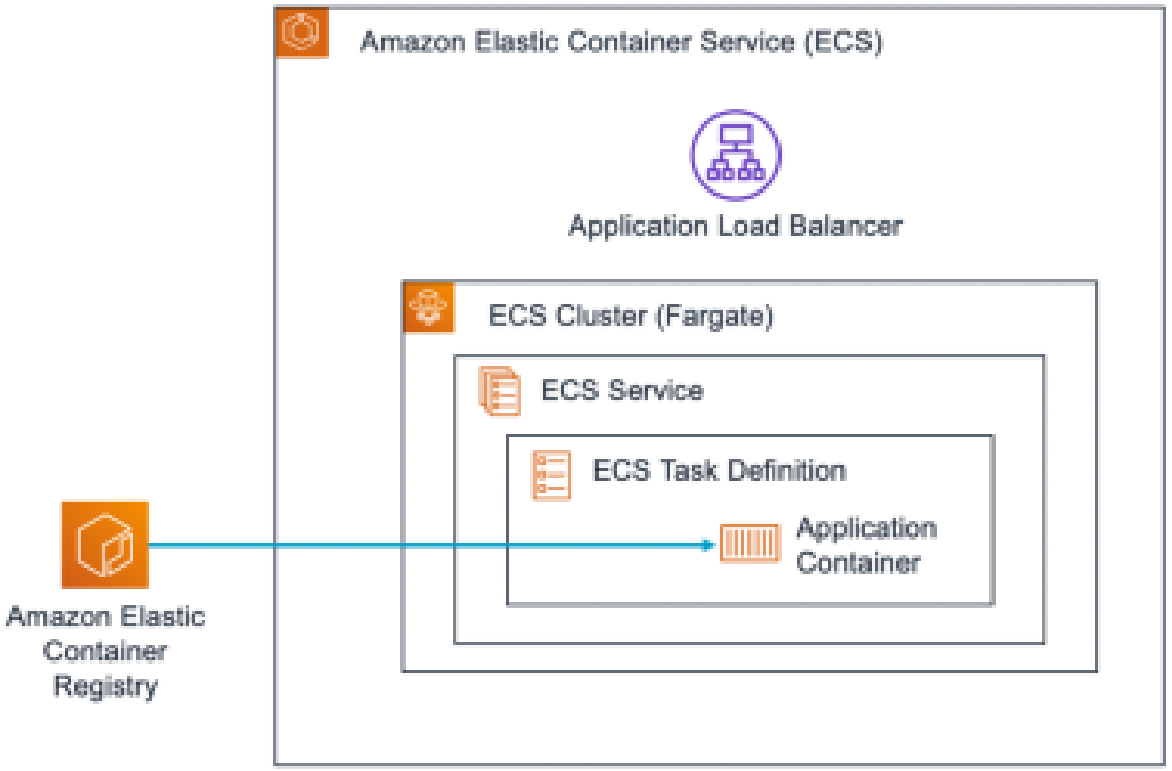


- CodeDeploy is used with other tools as part of a complete CI/CD pipeline
- CodeDeploy manages deployment of application components onto compute resources that are part of a "Deployment Group"
- All infrastructure components are created outside of CodeDeploy

Elastic Container Service (Amazon ECS)

- Serviço de orquestração de containers totalmente gerenciado que oferece suporte ao Docker e permite que execute facilmente aplicativos em um cluster gerenciado.
- O AWS ECS elimina a necessidade de instalar, operar e dimensionar a infraestrutura de gerenciamento de containers e simplifica a criação de ambientes com recursos básicos familiares da AWS, como grupos de segurança, Balanceamento e AWS Identity and Access Management (IAM).

Amazon ECS – Aplicativo em Container

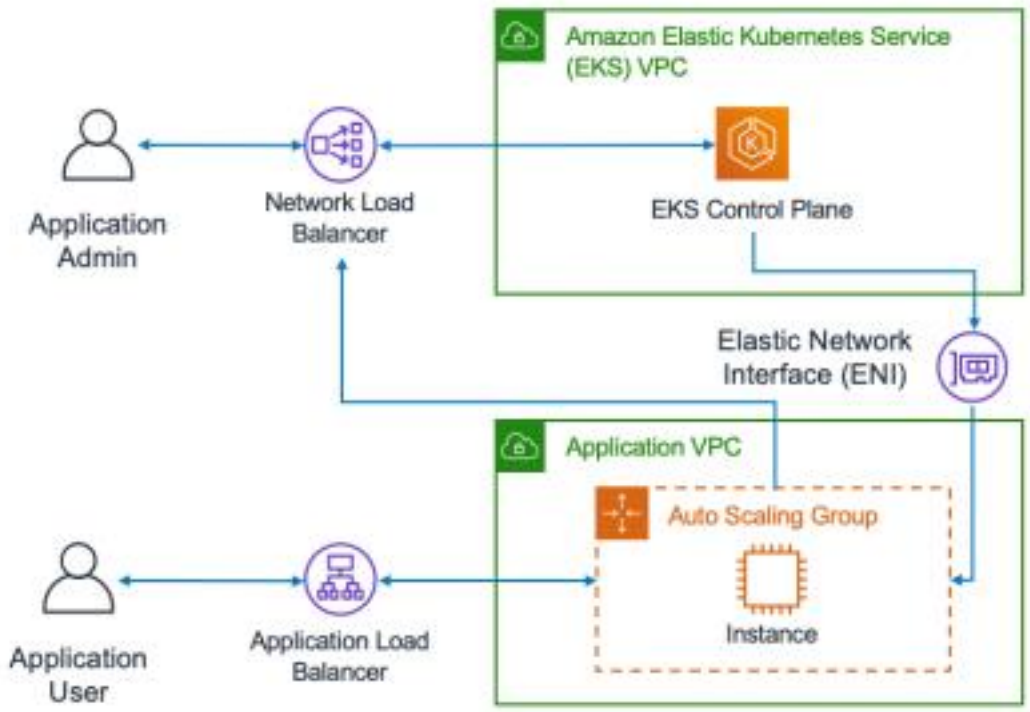


- Application infrastructure (including ECR Repositories, ECS Configurations, Load Balancers, etc.) is provisioned and managed outside ECS
- ECS manages the deployment and scale of application container instances that are sourced from a container registry

Amazon Elastic Kubernetes Service (Amazon EKS)

- O Amazon Elastic Kubernetes Service (Amazon EKS) é umserviço de Kubernetes totalmente gerenciado e certificado, que simplifica o processo de criação, proteção, operação e manutenção de clusters Kubernetes na AWS.
- O Amazon EKS integra-se aos principais serviços da AWS, como CloudWatch, Auto Scaling Groups e IAM para fornecer uma experiência perfeita para monitoramento, dimensionamento e balanceamento de carga em containers.

Amazon EKS – Aplicação com Container



- Fully-managed Kubernetes control plane is accessed by application admins through an NLB with a static IP address
- Control plane monitors and manages application infrastructure
- Users access application from load balancer

AWS OpsWorks

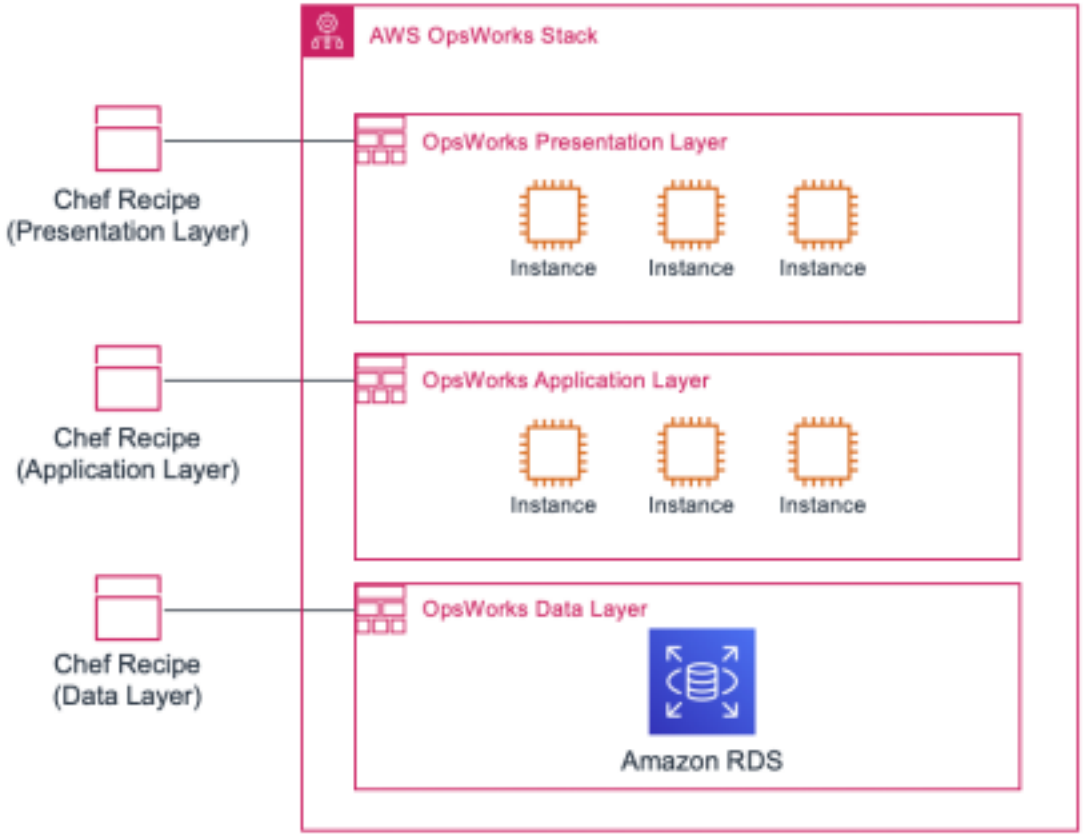
- Serviço de gerenciamento de configuração que permite construir, gerenciar, operar uma ampla variedade de arquiteturas de aplicativos, desde aplicativos Web simples até aplicativos altamente complexos e personalizados.
- As organizações que implantam aplicativos com OpsWorks usam as plataformas de automação Chef ou Puppet para gerenciar as principais atividades operacionais, como provisionamento de servidor, configurações de software, instalações de pacotes, configurações de banco de dados, dimensionamento e implantações de código.

AWS OpsWorks

Existem três maneiras de usar o OpsWorks:

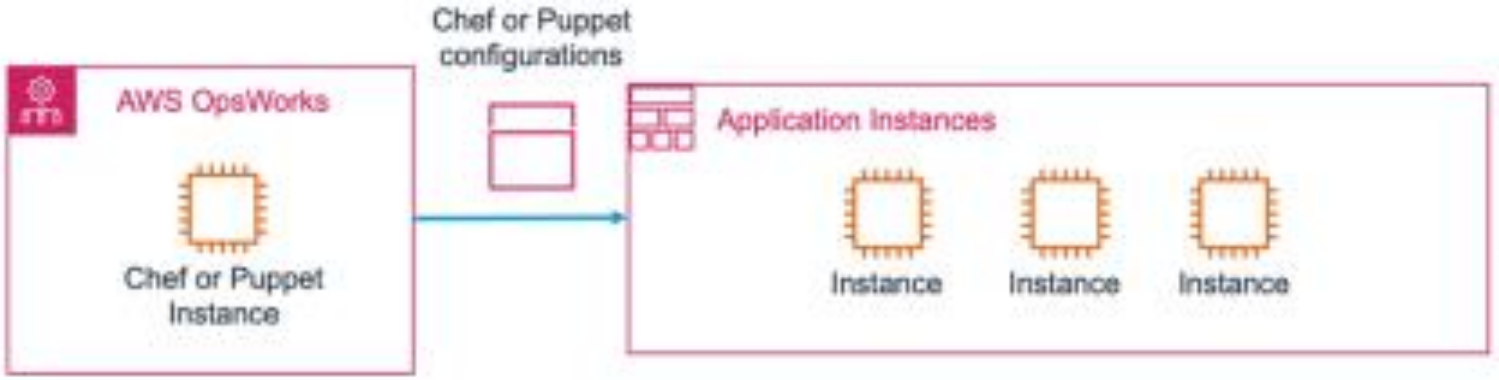
- AWS OpsWorks for Chef Automate: serviço de gerenciamento de configuração totalmente gerenciado que hospeda o Chef Automate.
- AWS OpsWorks for Puppet Enterprise: serviço de gerenciamento de configuração totalmente gerenciado que hospeda Puppet Enterprise
- AWS OpsWorks Stacks: serviço de gerenciamento de aplicativos e servidores que oferece suporte à modelagem de aplicações usando as abstrações de “stacks” e “layers” que dependem do Chef para gerenciamento de configurações.

AWS OpsWorks para Gerenciamento de aplicações Web de Três Camadas.



- A stack is divided into layers representing different parts of the application
- Chef recipes are used to define layer configurations
- Some AWS resources (e.g., Amazon RDS) need to be created outside OpsWorks and added to the layer

AWS OpsWorks com Chef Automate ou Puppet Enterprise



- Application instances are registered to a Chef or Puppet AWS OpsWorks instance
- Chef or Puppet configurations used to manage application instances

Serviços de Segurança da AWS para DevOps

1. AWS Identity and Access Management (IAM): O IAM permite que você controle o acesso aos serviços e recursos da AWS, criando e gerenciando identidades de acesso, como usuários, grupos e funções, e definindo permissões granulares.

2. AWS Key Management Service (KMS): O KMS permite a criação e gerenciamento de chaves de criptografia para proteger dados sensíveis em armazenamento e em trânsito.

3. AWS Web Application Firewall (WAF): O WAF protege suas aplicações web contra ataques comuns, como injeção de SQL e cross-site scripting (XSS), por meio de regras personalizadas.

4. AWS Shield: O AWS Shield oferece proteção contra ataques DDoS (Distributed Denial of Service) para manter suas aplicações disponíveis e seguras.

Serviços de Segurança da AWS para DevOps

5. AWS Inspector: O AWS Inspector automatiza a avaliação da segurança das suas aplicações e instâncias EC2, identificando possíveis vulnerabilidades e violações de segurança.

6. Amazon GuardDuty: O GuardDuty usa aprendizado de máquina para detectar atividades maliciosas e ameaças de segurança na sua conta da AWS.

7. AWS CloudTrail: O CloudTrail registra todas as atividades da API na sua conta, permitindo auditorias de segurança e a criação de trilhas de auditoria.

8. Amazon Macie: O Macie é uma ferramenta de segurança que utiliza machine learning para identificar dados sensíveis e monitorar a atividade de acesso a esses dados.

Serviços de Segurança da AWS para DevOps

9. AWS Organizations: Organizações permite a criação de políticas de controle de acesso centralizado e a consolidação de contas AWS para gerenciar eficientemente várias contas.

10. AWS Security Hub: O Security Hub fornece uma visão centralizada da segurança da sua infraestrutura AWS, agregando informações de vários serviços de segurança.

11. AWS Certificate Manager (ACM): O ACM facilita a implantação de certificados SSL/TLS para proteger a comunicação segura entre seus serviços.

12. AWS Secrets Manager: Secrets Manager ajuda a proteger e gerenciar senhas, tokens e outros segredos de forma segura.

Serviços de Segurança da AWS para DevOps

13. Amazon VPC (Virtual Private Cloud): A VPC permite a criação de redes privadas isoladas para suas instâncias EC2 e outros recursos, permitindo um maior controle de segurança.

14. Amazon Inspector: O Inspector ajuda a identificar vulnerabilidades de segurança em suas instâncias EC2 e fornece recomendações para mitigação.

15. AWS Firewall Manager: O Firewall Manager permite gerenciar regras de firewall em escala para aplicações em várias contas e VPCs.

- Estes são apenas alguns dos serviços de segurança da AWS que podem ser integrados à sua esteira de DevOps para melhorar a segurança da informação. A escolha dos serviços dependerá das necessidades específicas do seu projeto e do nível de segurança desejado.
- Importante é configurar e usar esses serviços de acordo com as melhores práticas de segurança da AWS para garantir a proteção adequada dos seus recursos e dados.

Parametrização da segurança em sua esteira de DevOps - AWS

Passo 1: Avaliação de Riscos e Requisitos de Segurança

- Antes de começar, é importante entender os requisitos de segurança específicos do seu aplicativo e avaliar os riscos associados. Isso ajudará a determinar quais medidas de segurança são necessárias em sua esteira de DevOps.

Passo 2: Identificação de Recursos AWS

- Identifique os recursos da AWS que serão usados na esteira de DevOps, como instâncias EC2, buckets S3, grupos de segurança, etc. Liste esses recursos para acompanhar e configurar suas políticas de segurança.

Passo 3: Gerenciamento de Acessos

- 3.1. Crie e gerencie identidades de acesso para equipes e indivíduos. Use o AWS Identity and Access Management (IAM) para conceder permissões apenas aos recursos necessários.
- 3.2. Implemente autenticação de dois fatores (2FA) para contas de administrador e outros usuários com privilégios elevados.

Passo 4: Políticas de Segurança

- 4.1. Defina políticas de segurança que restrinjam o acesso a recursos sensíveis. Use políticas do IAM para controlar permissões de recursos.
- 4.2. Implemente uma política de retenção de chaves de acesso e roteamento para garantir a revogação de chaves comprometidas.

Parametrização da segurança em sua esteira de DevOps - AWS

Passo 5: Proteção de Dados e Comunicações

- 5.1. Criptografe dados sensíveis em repouso usando o AWS Key Management Service (KMS).
- 5.2. Use HTTPS para comunicações seguras entre os componentes da esteira de DevOps.

Passo 6: Monitoramento e Auditoria

- 6.1. Configure registros detalhados de auditoria e atividade usando o AWS CloudTrail e o Amazon CloudWatch.
- 6.2. Implemente alertas para atividades suspeitas ou não autorizadas.

Passo 7: Segurança na Infraestrutura

- 7.1. Mantenha sistemas operacionais, aplicativos e bibliotecas atualizados com patches de segurança.
- 7.2. Utilize grupos de segurança para controlar o tráfego de entrada e saída para instâncias EC2.

Passo 8: Integração Contínua/Entrega Contínua (CI/CD)

- 8.1. Garanta que os pipelines de CI/CD estejam seguros, verificando automaticamente as vulnerabilidades de segurança no código-fonte e nas dependências.
- 8.2. Implemente controles de aprovação antes da implantação em ambientes de produção.

Parametrização da segurança em sua esteira de DevOps - AWS

Passo 9: Testes de Segurança Automatizados

- 9.1. Integre testes de segurança automatizados, como varreduras de segurança e testes de penetração, em seu pipeline de CI/CD.
- 9.2. Configure uma política de "negação padrão" para recursos, permitindo apenas o tráfego necessário.

Passo 10: Documentação e Treinamento

- 10.1. Documente todas as políticas de segurança, procedimentos e configurações.
- 10.2. Forneça treinamento de segurança para a equipe de DevOps e outros envolvidos na esteira.

Passo 11: Monitoramento Contínuo e Melhorias

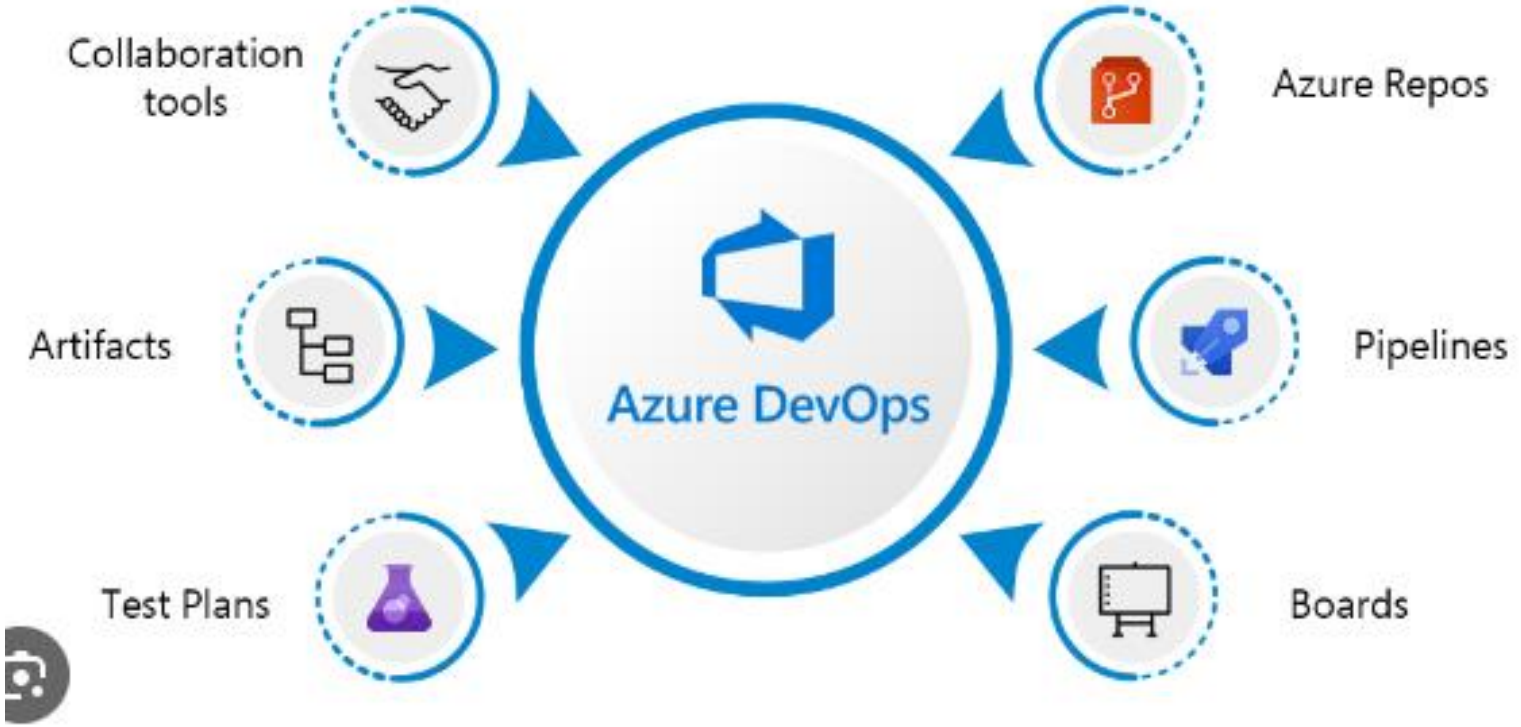
- 11.1. Estabeleça monitoramento contínuo e reveja periodicamente as políticas e controles de segurança para fazer melhorias.
- 11.2. Esteja preparado para responder a incidentes de segurança com um plano de resposta a incidentes.

Lembrando que a segurança na esteira de DevOps é uma tarefa contínua. Ameaças e requisitos podem evoluir ao longo do tempo, então é essencial manter-se atualizado e ajustar suas medidas de segurança conforme necessário.

AWS – Tutoriais de DevOps Opcionais

- CodeDeploy tutorials: https://docs.aws.amazon.com/codedeploy/latest/userguide/tutorials.html
- Configure um pipeline de implantação contínua: https://aws.amazon.com/pt/getting-started/hands-on/continuous-deployment-pipeline/
- Migrate to AWS CodeCommit: https://docs.aws.amazon.com/codecommit/latest/userguide/how-to-migrate-repository.html
- Migrate a Git repository to AWS CodeCommit: https://docs.aws.amazon.com/codecommit/latest/userguide/how-to-migrate-repository-existing.html
- Migrate a repository incrementally: https://docs.aws.amazon.com/codecommit/latest/userguide/how-to-push-large-repositories.html
- CodePipeline tutorials: https://docs.aws.amazon.com/codepipeline/latest/userguide/tutorials.html
- Using AWS CodePipeline with AWS OpsWorks Stacks: https://docs.aws.amazon.com/opsworks/latest/userguide/other-services-cp.html
- AWS OpsWorks for Puppet Enterprise: https://docs.aws.amazon.com/opsworks/latest/userguide/welcome_opspup.html
- AWS CodePipeline with AWS OpsWorks Stacks - Chef 12 Stacks: https://docs.aws.amazon.com/opsworks/latest/userguide/other-services-cp-chef12.html
- Create a pipeline with AWS CloudFormation: https://docs.aws.amazon.com/codepipeline/latest/userguide/tutorials-cloudformation.html
- Tutorial: Amazon ECS Standard Deployment with CodePipeline: https://docs.aws.amazon.com/codepipeline/latest/userguide/ecs-cd-pipeline.html
- Tutorial: Create a pipeline that uses Amazon S3 as a deployment provider: https://docs.aws.amazon.com/codepipeline/latest/userguide/tutorials-s3deploy.html

Microsoft Azure DevOps



DevOps na Microsoft Azure

- O Azure oferece o Azure DevOps, um serviço empacotado com um conjunto de ferramentas para gerenciar projetos de desenvolvimento de software de ponta a ponta. Este serviço inclui o Azure DevOps Server e o serviço de nuvem Azure DevOps.
- O servidor fornece um ambiente de tempo de execução com ferramentas baseadas em nuvem para facilitar as implantações no local.
- O serviço de nuvem, por outro lado, apresenta ferramentas como os Boards do Azure, Azure Pipeline, relatórios do Azure, planos de teste do Azure e artefatos do Azure para desenvolvimento, teste e implantação de software na nuvem.

https://azure.microsoft.com/pt-br/products/devops/#overview

DevOps na Microsoft Azure

GitHub

Aumente a colaboração, automatize seus fluxos de trabalho de código para nuvem e ajude a proteger seu código com recursos avançados.

Azure Pipelines

Implemente CI/CD para realizar a criação, o teste e a implantação de maneira contínua em praticamente qualquer plataforma ou nuvem.

Azure Boards

Planeje, acompanhe e discuta o trabalho em suas equipes usando quadros Kanban, listas de pendências, painéis de equipe e relatórios personalizados.

Azure Monitor

Tenha uma observabilidade completa de aplicativos, da infraestrutura e da rede.

Visual Studio

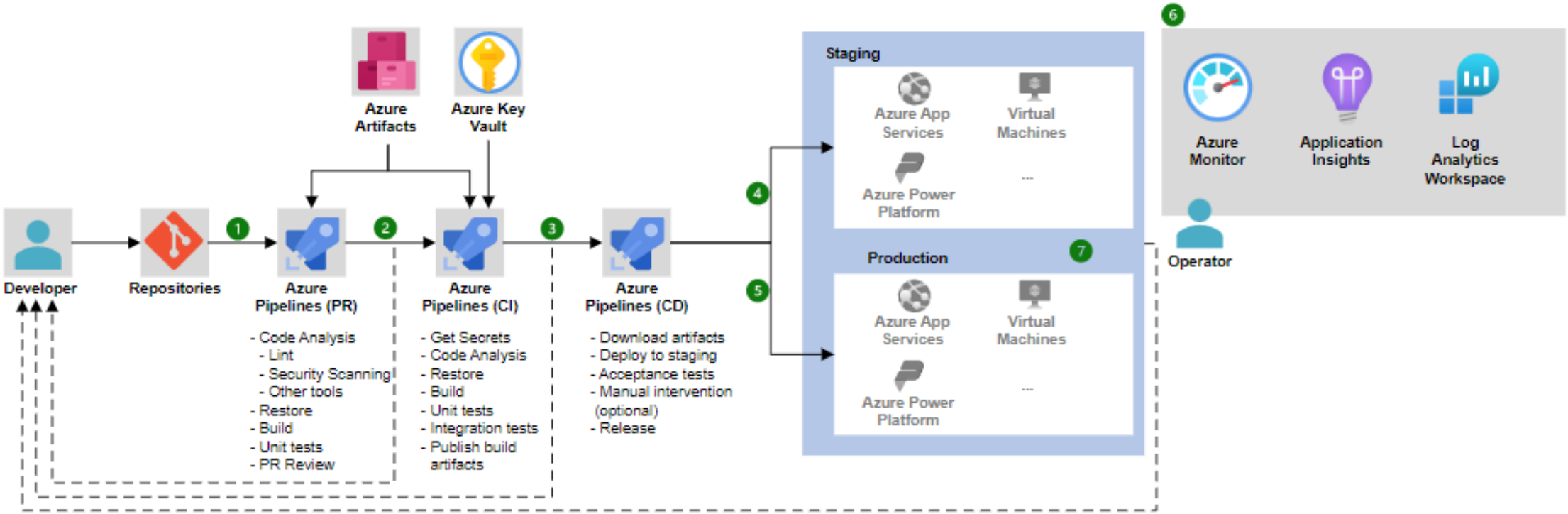
Use o IDE (ambiente de desenvolvimento integrado) projetado para criar aplicativos avançados e escalonáveis para o Azure.

AKS (Serviço de Kubernetes do Azure)

Envie aplicativos em contêineres mais rapidamente e opere-os com mais facilidade usando um serviço de Kubernetes totalmente gerenciado.

https://azure.microsoft.com/pt-br/solutions/devops

DevOps no Azure



DevOps de alto nível para implantar alterações de aplicativo em ambientes de preparo e produção no Azure. com práticas de CI/CD (integração contínua/implantação contínua) e o Azure Pipelines.

Fluxo de Dados – Azure Pipelines

1. **Pipeline de PR:** uma PR (solicitação de pull) para Azure Repos Git dispara um pipeline de PR. Esse pipeline executa verificações de qualidade rápidas. Essas verificações devem incluir:
   - Compilar o código, que requer a extração de dependências de um sistema de gerenciamento de dependências.
   - O uso de ferramentas para analisar o código, como análise de código estático, lint e verificação de segurança
   - Testes de unidade

Fluxo de Dados – Azure Pipelines

2. Pipeline de CI – uma mesclagem para Azure Repos Git dispara um pipeline de CI.

- Esse pipeline executa as mesmas verificações que o pipeline de PR com algumas adições importantes.
- O pipeline de CI executa testes de integração. Esses testes de integração não devem exigir a implantação da solução, pois os artefatos de build ainda não foram criados.
- Se os testes de integração exigirem segredos, o pipeline obterá esses segredos do Azure Key Vault.
- Se qualquer uma das verificações falhar, o pipeline terminará e o desenvolvedor terá que fazer as alterações necessárias.
- O resultado de uma execução bem-sucedida desse pipeline é a criação e publicação de artefatos de build

Fluxo de Dados – Azure Pipelines

3. Gatilho de pipeline de CD – a publicação de artefatos dispara o pipeline de CD.

4. Versão de CD para preparo – o pipeline de CD baixa os artefatos de build criados no pipeline de CI e implanta a solução em um ambiente de preparo.

- Em seguida, o pipeline executa testes de aceitação no ambiente de preparo para validar a implantação.
- Se algum teste de aceitação falhar, o pipeline será encerrado e o desenvolvedor precisará fazer as alterações necessárias.
- Se os testes forem bem-sucedidos, uma tarefa de validação manual poderá ser implementada para exigir que uma pessoa ou grupo valide a implantação e retome o pipeline.

Fluxo de Dados – Azure Pipelines

4. Lançamento de CD para produção – se a intervenção manual for retomada ou não houver intervenção manual implementada, o pipeline liberará a solução para produção.

- O pipeline deve executar smoke tests em produção para garantir que a versão esteja funcionando conforme o esperado.
- Se uma etapa de intervenção manual resultar em um cancelamento, a versão falhar ou os smoke tests falharem, a versão for revertida, o pipeline terminar e o desenvolvedor terá que fazer as alterações necessárias.

5. Monitoramento – o Azure Monitor coleta dados de observabilidade, como logs e métricas, para que um operador possa analisar dados de integridade, desempenho e uso.

- O Application Insights coleta todos os dados de monitoramento específicos do aplicativo, como rastreamentos.
- O Azure Log Analytics é usado para armazenar todos esses dados.

Componentes da Arquitetura

- Um repositório git **Azure Repos** serve como um repositório de código que fornece controle de versão e uma plataforma para projetos colaborativos.
- O **Azure Pipelines** fornece uma maneira de criar, testar, empacotar e liberar código de aplicativo e infraestrutura. Este exemplo tem três pipelines distintos com as seguintes responsabilidades:
    - Os pipelines de PR validam o código antes de permitir que uma PR mescle por meio de lint, compilação e teste de unidade.
    - Os pipelines de CI são executados depois que o código é mesclado. Eles executam a mesma validação que os pipelines de PR, mas adicionam testes de integração e publicam artefatos de build se tudo for bem-sucedido.
    - Os pipelines de CD implantam artefatos de build, executam testes de aceitação e liberam para produção.

Componentes da Arquitetura

- Os **Feeds de Artefatos do Azure** permitem que você gerencie e compartilhe pacotes de software, como Maven, npm e NuGet. Os feeds de artefato permitem que você gerencie o ciclo de vida de seus pacotes, incluindo controle de versão, promoção e desativação de pacotes. Isso ajuda você a garantir que sua equipe esteja usando as versões mais recentes e seguras de seus pacotes.
- **Key Vault** fornece uma maneira de gerenciar dados seguros para sua solução, incluindo segredos, chaves de criptografia e certificados. Nessa arquitetura, ela é usada para armazenar segredos do aplicativo. Esses segredos são acessados por meio do pipeline. Os segredos podem ser acessados pelo Azure Pipelines com uma tarefa Key Vault ou vinculando segredos de Key Vault.

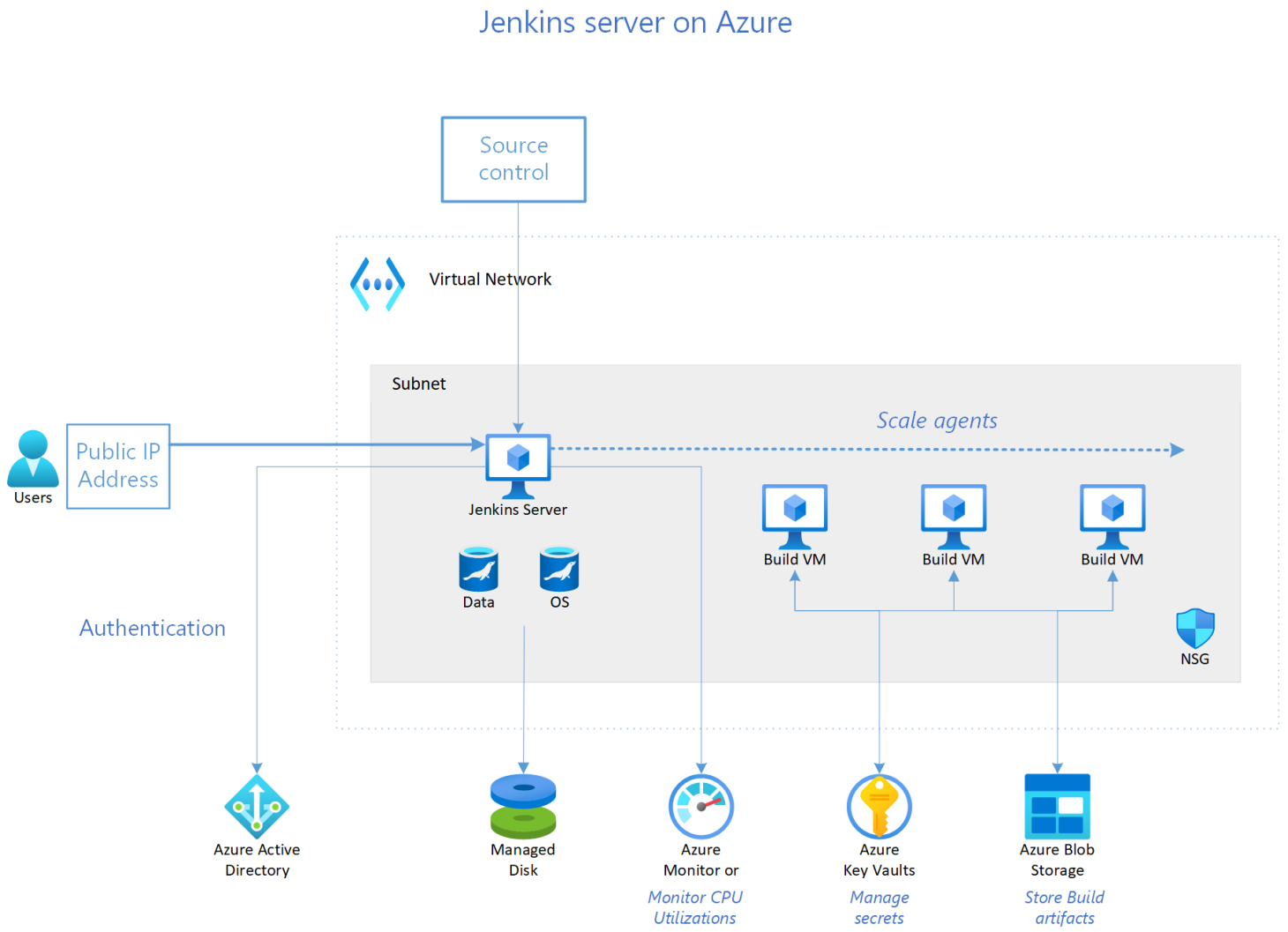
Componentes da Arquitetura

- **Monitor** é um recurso de observabilidade que coleta e armazena métricas e logs, telemetria de aplicativos e métricas de plataforma para os serviços do Azure. Use esses dados para monitorar o aplicativo, configurar alertas e painéis e executar a análise da causa raiz de falhas.
- O **Application Insights** é um serviço de monitoramento que fornece insights em tempo real sobre o desempenho e o uso de seus aplicativos Web.
- O **Workspace do Log Analytics** fornece um local central onde você pode armazenar, consultar e analisar dados de várias fontes, incluindo recursos, aplicativos e serviços do Azure.

Alternativas para o Azure Pipeline

- **Azure DevOps Server** (anteriormente conhecido como Team Foundation Server) poderia ser usado como um substituto local.
- O **Jenkins** é uma ferramenta de código aberto usada para automatizar builds e implantações.
- **GitHub Actions** permitir que você automatize seus fluxos de trabalho de CI/CD diretamente do GitHub.
- Os **repositórios do GitHub** podem ser substituídos como o repositório de código. O Azure Pipelines integra-se perfeitamente aos repositórios do GitHub.

Executar um servidor Jenkins no Azure



Práticas de CI/CD

O uso de práticas comprovadas de CI e CD para implantar alterações de aplicativo ou infraestrutura oferece vários benefícios, incluindo:

- **Ciclos de versão mais curtos** – os processos automatizados de CI/CD permitem que você implante mais rapidamente do que as práticas manuais. Muitas organizações implantam várias vezes por dia.
- **Melhor qualidade de código** – os portões de qualidade em pipelines de CI, como lint e teste de unidade, resultam em código de maior qualidade.
- **Diminuição do risco de liberação** – as práticas adequadas de CI/CD reduzem drasticamente o risco de liberar novos recursos. A implantação pode ser testada antes do lançamento.
- **Aumento da produtividade** – a CI/CD automatizada libera os desenvolvedores de trabalhar em integrações e implantações manuais para que eles possam se concentrar em novos recursos.
- **Habilitar reversões – embora as práticas adequadas** de CI/CD reduzam o número de bugs ou regressões liberados, elas ainda ocorrem. A CI/CD pode habilitar reversões automatizadas para versões anteriores.

Possíveis Casos de Uso

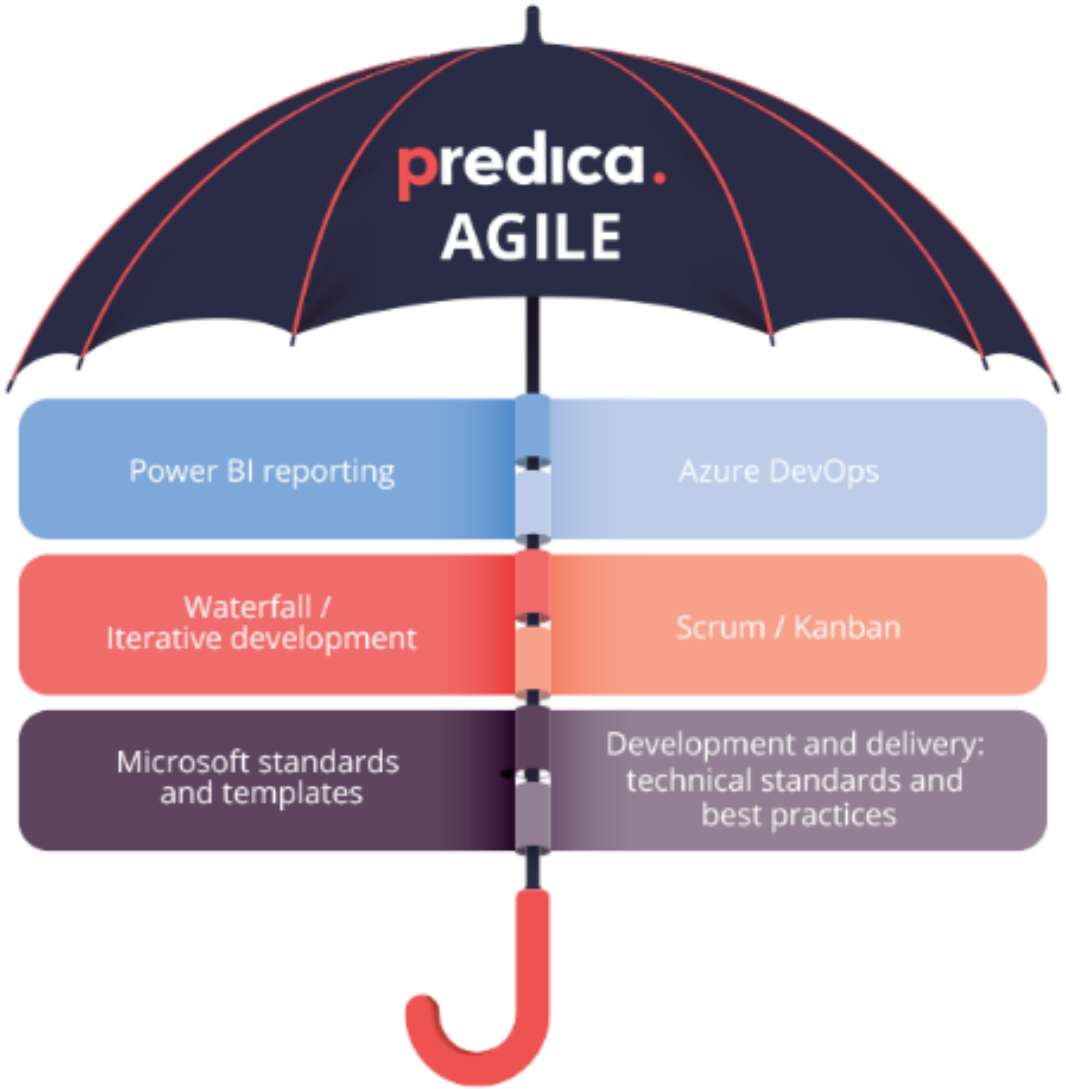
Considere os processos de CI/CD e do Azure Pipelines para:

- Acelerar o desenvolvimento de aplicativos e ciclos de vida de desenvolvimento.
- Criar qualidade e consistência em um processo automatizado de build e lançamento.
- Aumentar a estabilidade e o tempo de atividade do aplicativo.

O Framework “Predica Agile”

“Predica Agile”

- Uma abordagem padronizada para a entrega de serviços, na forma de melhores práticas, que estão sendo continuamente compartilhadas e desenvolvidas relatórios e KPIs de projetos alinhados com a abordagem. Eles, por sua vez, nos ajudam a comparar,
- Avaliar e comunicar rapidamente o status do projeto (para isso, usamos o Power BI junto com o backlog do Azure DevOps)os funcionários podem fluir entre projetos e engenheiros/equipes dedicados com pouco aumento e interrupção.



Serviços de Segurança da Azure para sua Esteira de DevOps

A Microsoft Azure oferece uma ampla gama de serviços de segurança da informação que podem ser usados para fortalecer a segurança em sua esteira de DevOps.

1. Azure Active Directory (Azure AD): O Azure AD permite gerenciar identidades e autenticação, fornecendo autenticação de dois fatores (2FA), controle de acesso baseado em funções (RBAC) e integração com serviços de terceiros.

2. Azure Key Vault: O Azure Key Vault permite gerenciar chaves de criptografia e segredos de forma segura, garantindo o armazenamento e o acesso seguros a informações sensíveis.

3. Azure Security Center: O Security Center oferece visibilidade de segurança e recomendações para melhorar a postura de segurança de suas cargas de trabalho na Azure.

4. Azure Policy: O Azure Policy permite a implementação de políticas de conformidade para controlar e auditar recursos na sua assinatura Azure.

Serviços de Segurança da Azure para sua Esteira de DevOps

5. Azure Sentinel: O Azure Sentinel é uma solução SIEM (Security Information and Event Management) baseada na nuvem que ajuda na detecção e resposta a ameaças.

6. Azure Firewall: O Azure Firewall fornece proteção de firewall de rede para aplicativos e recursos na nuvem.

7. Azure DDoS Protection: A DDoS Protection oferece proteção contra ataques distribuídos de negação de serviço (DDoS).

8. Azure Virtual Network (VNet): O VNet permite criar redes virtuais isoladas para proteger suas instâncias e recursos.

Serviços de segurança da Azure para sua Esteira de DevOps

9. Azure DevOps Security Features: O Azure DevOps oferece recursos de segurança integrados, como controle de acesso baseado em funções (RBAC) e integração com Azure AD para autenticação segura.

10. Azure Information Protection: Este serviço permite classificar e proteger documentos e emails confidenciais.

11. Azure Advanced Threat Protection: O Azure ATP monitora e detecta atividades suspeitas em ambientes do Active Directory.

12. Azure Bastion: O Bastion fornece um gateway seguro para acesso RDP e SSH a máquinas virtuais em redes virtuais.

Serviços de Segurança da Azure para sua Esteira de DevOps

13. Azure Monitor: O Monitor ajuda a coletar, analisar e agir sobre dados de telemetria de recursos na Azure.

14. Azure Application Gateway: Este serviço fornece balanceamento de carga de aplicativos com recursos de segurança, incluindo proteção contra ataques de aplicativos da web (WAF).

15. Azure Security Information and Event Management (SIEM): Além do Azure Sentinel, você pode integrar serviços de SIEM de terceiros à Azure para maior visibilidade e detecção de ameaças.

- Esses são apenas alguns dos serviços de segurança da informação fornecidos pela Azure que podem ser integrados à sua esteira de DevOps.
- A escolha dos serviços dependerá das necessidades específicas do seu projeto e do nível de segurança desejado.
- Lembre-se de configurar e usar esses serviços de acordo com as melhores práticas de segurança da Azure para garantir a proteção adequada dos seus recursos e dados.

Configuração da Segurança em sua Esteira de DevOps na Microsoft Azure

Passo 1: Avaliação de Riscos e Requisitos de Segurança

Antes de começar, é importante entender os requisitos de segurança específicos do seu aplicativo e avaliar os riscos associados. Isso ajudará a determinar quais medidas de segurança são necessárias em sua esteira de DevOps.

Passo 2: Identificação de Recursos Azure

Identifique os recursos da Azure que serão usados em sua esteira de DevOps, como máquinas virtuais, contêineres, bancos de dados, etc. Liste esses recursos para acompanhar e configurar políticas de segurança.

Passo 3: Gerenciamento de Acessos

3.1. Configure o Azure Active Directory (Azure AD) para gerenciar identidades de acesso. Use o Azure AD para conceder permissões apenas aos recursos necessários.

3.2. Implemente autenticação de dois fatores (2FA) para contas de administrador e outros usuários com privilégios elevados.

Configuração da Segurança em sua Esteira de DevOps na Microsoft Azure

Passo 4: Políticas de Segurança

4.1. Defina políticas de segurança que restrinjam o acesso a recursos sensíveis. Use o Azure Policy para controlar permissões de recursos.

4.2. Implemente uma política de retenção de chaves de acesso e roteamento para garantir a revogação de chaves comprometidas.

Passo 5: Proteção de Dados e Comunicações

5.1. Criptografe dados sensíveis em repouso usando o Azure Disk Encryption e o Azure SQL Database Transparent Data Encryption (TDE).

5.2. Use HTTPS para comunicações seguras entre os componentes da esteira de DevOps.

Passo 6: Monitoramento e Auditoria

6.1. Configure registros detalhados de auditoria usando o Azure Monitor e o Azure Security Center.

6.2. Implemente alertas para atividades suspeitas ou não autorizadas usando o Azure Monitor.

Configuração da Segurança em sua Esteira de DevOps na Microsoft Azure

Passo 7: Segurança na Infraestrutura

7.1. Mantenha sistemas operacionais, aplicativos e bibliotecas atualizados com patches de segurança.

7.2. Utilize grupos de segurança de rede (Network Security Groups) para controlar o tráfego de entrada e saída para máquinas virtuais.

Passo 8: Integração Contínua/Entrega Contínua (CI/CD)

8.1. Configure um serviço de CI/CD, como Azure DevOps, para automatizar compilações e implantações seguras.

8.2. Implemente práticas de segurança no pipeline de CI/CD, como verificação de vulnerabilidades no código-fonte e nos contêineres.

Passo 9: Testes de Segurança Automatizados

9.1. Integre ferramentas de teste de segurança automatizado, como Azure Security Center e Azure Defender, em seu pipeline de CI/CD.

9.2. Configure uma política de "negação padrão" para recursos, permitindo apenas o tráfego necessário.

Configuração da Segurança em sua Esteira de DevOps na Microsoft Azure

Passo 10: Documentação e Treinamento

10.1. Documente todas as políticas de segurança, procedimentos e configurações.

10.2. Forneça treinamento de segurança para a equipe de DevOps e outros envolvidos na esteira.

Passo 11: Monitoramento Contínuo e Melhorias

11.1. Estabeleça monitoramento contínuo e reveja periodicamente as políticas e controles de segurança para fazer melhorias.

11.2. Esteja preparado para responder a incidentes de segurança com um plano de resposta a incidentes.

A segurança em uma esteira de DevOps na Azure é uma tarefa contínua.

Ameaças e requisitos podem evoluir ao longo do tempo, então é essencial manter-se atualizado e ajustar suas medidas de segurança conforme necessário, seguindo as melhores práticas de segurança da Azure.

Tutoriais Azure DevOps Opcionais

1. Criar um pipeline de CI/CD para .NET com o Azure DevOps Starter: https://learn.microsoft.com/pt-br/previous-versions/azure/devops-project/azure-devops-project-aspnet-core
2. Create a CI/CD pipeline for .NET with the DevOps Starter Project: https://www.azuredevopslabs.com/labs/vstsextend/azuredevopsprojectdotnet/
3. Examine the CI/CD pipelines configured by Azure DevOps Project: https://www.azuredevopslabs.com/labs/vstsextend/azuredevopsprojectdotnet/#exercise-2-examine-the--cicd-pipelines-configured-by-azure-devops-project
4. Commit the code changes and execute CI/CD: https://www.azuredevopslabs.com/labs/vstsextend/azuredevopsprojectdotnet/#exercise-3-commit-the-code-changes-and-execute-cicd

Google Cloud Plataform (GCP) DevOps

DevOps na Google GCP

- GCP oferece suporte ao DevOps fornecendo os serviços necessários para desenvolver, armazenar e implantar software de alta qualidade, em ciclos mais curtos.
- A plataforma do Google Cloud apresenta instâncias de até 96 vCPUs e 624 GB de RAM, juntamente com serviços como o console de nuvem, o Google Compute Engine e o gerenciador de implantação do GCP, que oferece suporte à implementação de DevOps no Google Cloud Platform.
- O Google Cloud apresenta como recursos de inteligência artificial, aprendizado de máquina e análise de dados, como diferenciais de DevOps.

DevOps na Google GCP

Cloud Build

Personalize fluxos de trabalho para criar, testar e implantar em vários ambientes.

Artifact Registry

Armazene, gerencie e proteja imagens de contêineres e pacotes de linguagens.

Autorização binária

Garanta que apenas imagens de contêiner confiáveis sejam implantadas no Google Kubernetes Engine.

Tekton

Um framework de código aberto para criar sistemas de integração e entrega contínuas (CI/CD).

Google Cloud Deploy

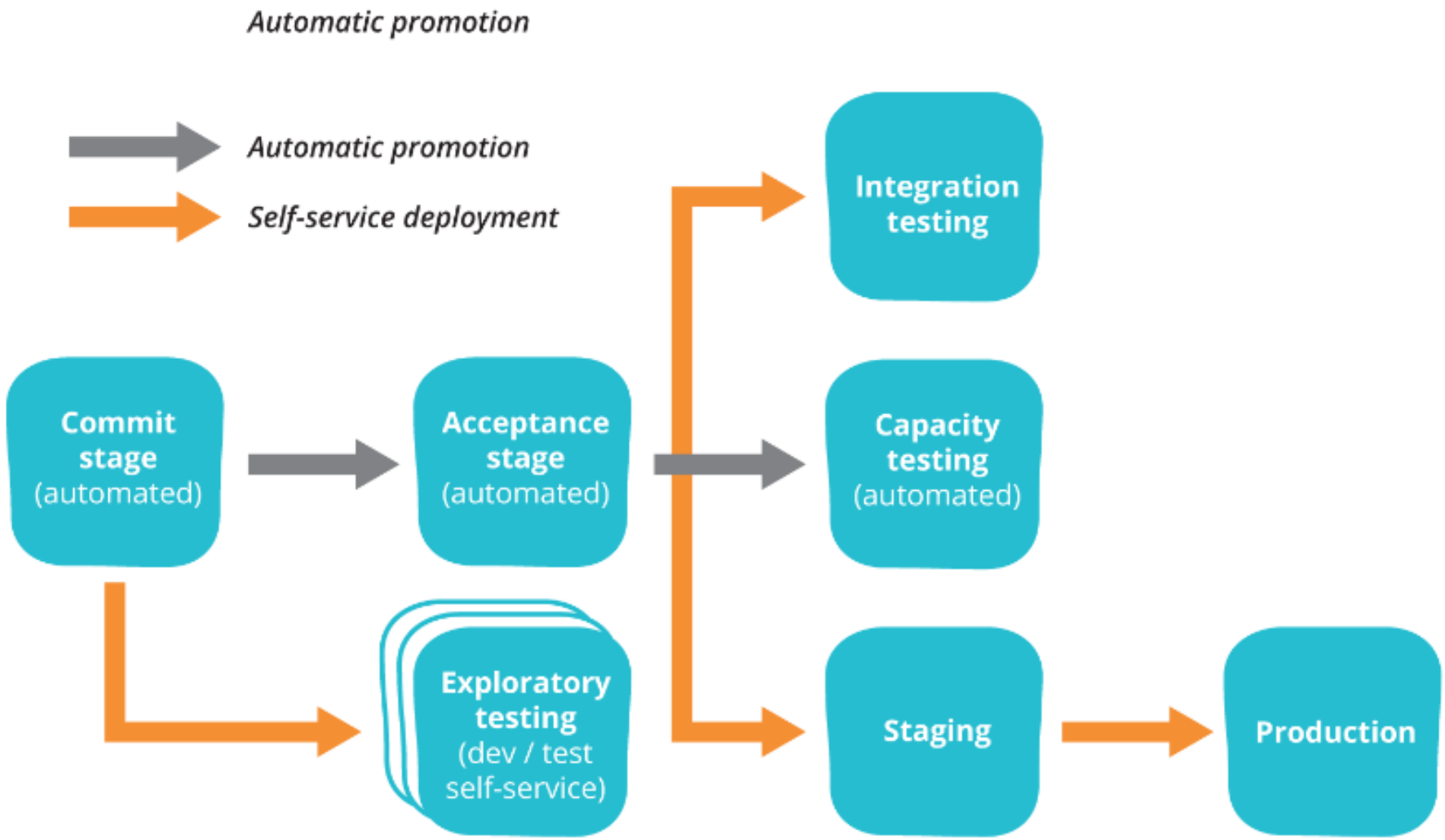
Entrega contínua totalmente gerenciada para o Google Kubernetes Engine com métricas, aprovações e segurança integradas.

Pacote de operações

Monitore, solucione problemas e melhore a infraestrutura e o desempenho dos apps.

https://cloud.google.com/devops?hl=pt-br#section-1

Deployment Pipeline



Formas de Medir a CI

Fator a testar	O que medir
Confirmações de código acionam uma versão do software	A porcentagem de confirmações de código que gera uma versão de software sem intervenção manual.
Confirmações de código acionam uma série de testes automatizados	A porcentagem de confirmações de código que gera um conjunto de testes automatizados executados sem intervenção manual.
Versões e testes automatizados executados com êxito todos os dias	A porcentagem de versões automatizadas e a porcentagem de testes automatizados que são executados com êxito todos os dias.
Versões atuais estão disponíveis para testers para testagem exploratória	A disponibilidade das versões para os testers ou o inverso, isto é, a indisponibilidade das versões para os testers.
Os desenvolvedores recebem feedback dos testes de aceitação e desempenho todos os dias	A disponibilidade de feedback proveniente de testes de aceitação e desempenho aos desenvolvedores, ou seja, a porcentagem de testes que fornecem feedback disponível aos desenvolvedores no período de um dia.
Versões com falha são corrigidas imediatamente	Tempo decorrido entre a falha da versão e a respectiva correção, seja com uma entrada que corrige o problema ou com a reversão da alteração que causa falha.

Dificuldades na adoção da CI

- Não incluir tudo no repositório de código. Tudo o que é preciso para criar e configurar o aplicativo e o sistema deverá estar no seu repositório. Embora possa parecer fora do escopo da CI, isso é uma base importante.
- Não automatizar o processo de criação. Etapas manuais criam oportunidades de erros e não são documentadas.
- Não acionar testes rápidos em cada alteração. Testes completos são necessários, mas os testes rápidos (geralmente de unidade) também são importantes para permitir feedback rápido.
- Não corrigir versões com falha imediatamente. Uma meta essencial da CI é ter uma versão estável que sirva como base de desenvolvimento para todos. Se não for possível corrigir a versão em poucos minutos, a alteração que gerou a falha da versão deverá ser identificada e revertida.
- Adotar testes de execução muito demorada. A execução dos testes não deve demorar mais que alguns minutos, com um limite superior de aproximadamente 10 minutos, de acordo com a pesquisa do DORA (PDF, em inglês). Se sua versão demora mais que isso, aprimore a eficácia dos testes, adicione mais recursos de computação para executá-los em paralelo ou divida os testes de longa duração em um build separado usando o padrão de pipeline de implantação (em inglês).
- Não mesclar no tronco com a devida frequência. Muitas organizações têm testes e versões automatizados, mas não aplicam uma mesclagem diária no tronco. Isso acarreta branches de longa duração, que são muito mais difíceis de integrar, e longos ciclos de feedback para os desenvolvedores.

GCP – Core Build

- Crie, teste e implante na nossa plataforma de CI/CD sem servidor
- Crie software rapidamente em todas as linguagens de programação, incluindo Java, Go, Node.js e mais.
- Escolha entre 15 tipos de máquinas e execute centenas de builds simultâneas por pool
- Implante em vários ambientes, como VMs, sem servidor, Kubernetes ou Firebase.
- Acesse fluxos de trabalho CI/CD totalmente gerenciados, hospedados na nuvem, na sua rede particular
- Mantenha seus dados em repouso dentro de uma região geográfica ou um local específico com a residência de dados

GCP – Core Build

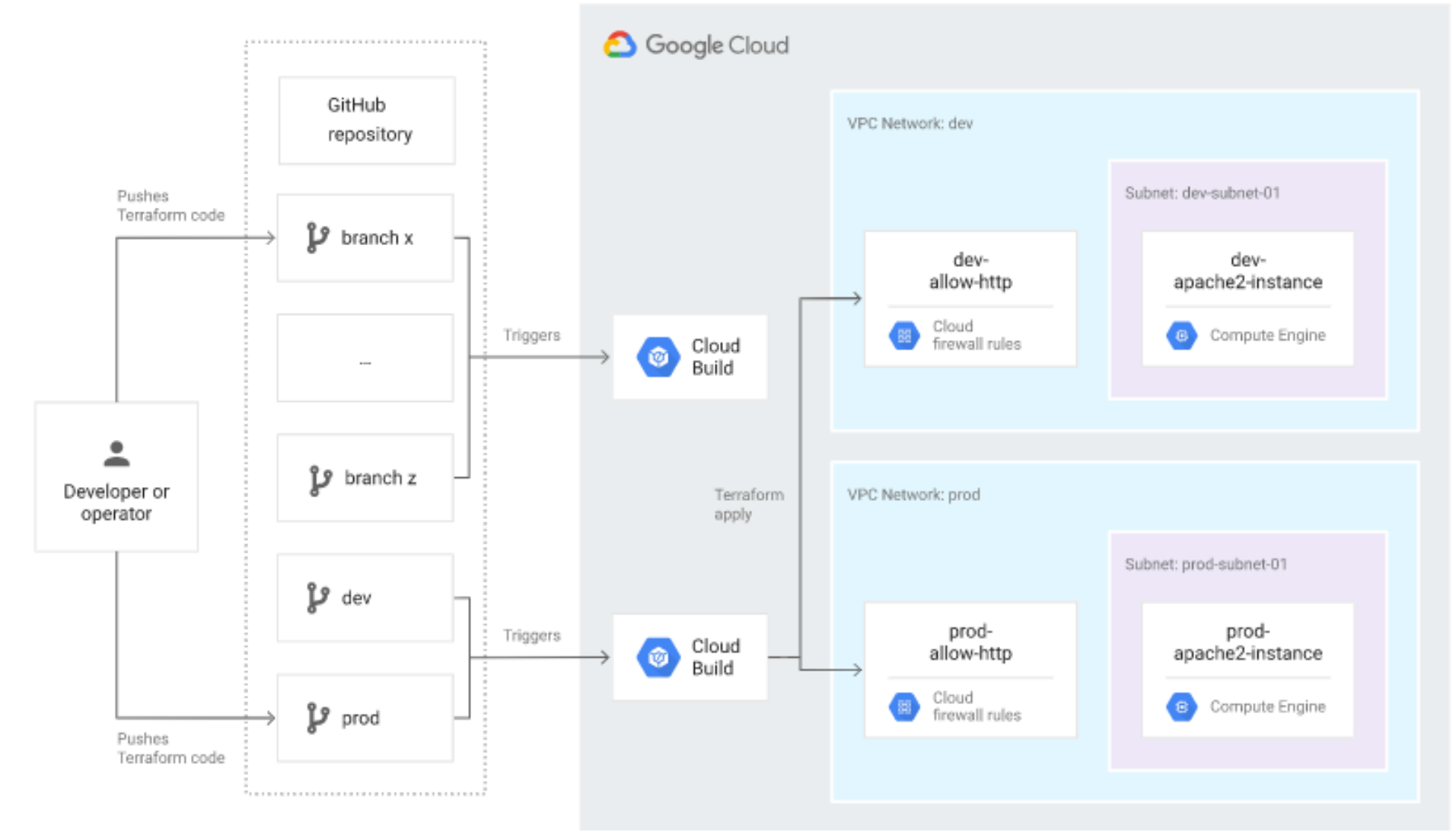
- Plataforma totalmente sem servidor para escalonamento
  - O Cloud Build oferece escalonamento vertical e horizontal sem infraestrutura para configurar, atualizar ou escalonar. Execute suas versões em um ambiente totalmente gerenciado no Google Cloud com conectividade à sua própria rede privada.
- Integrações de origens nativas da empresa
  - Integre com alguns dos sistemas de controle de origem mais conhecidos com o suporte predefinido do Cloud Build para GitHub Enterprise, GitLab Enterprise e Data center do Bitbucket.

GCP – Core Build

- Segurança e conformidade da cadeia de suprimentos de software

  - Verifique vulnerabilidades nas imagens localmente ou no seu registro.
  - Use a procedência para auditar e controlar as implantações na produção.
  - Proteja-se contra ataques da cadeia de suprimentos de software com suporte à compilação no SLSA de nível 3.

IaaC - Terraform, o Cloud Build e o GitOps



CI/CD no Google Cloud

https://cloud.google.com/docs/ci-cd?hl=pt-br

Serviços de Segurança da GCP para DevOps

1. Identity and Access Management (IAM): O IAM permite gerenciar o acesso a recursos da GCP, permitindo a criação de políticas de acesso granulares para usuários, grupos e funções.

2. Cloud Identity Platform: Esse serviço oferece autenticação de usuário e gerenciamento de identidades, incluindo autenticação multifator (MFA) e SSO (Single Sign-On).

3. Cloud Security Command Center: O Security Command Center oferece visibilidade de segurança centralizada e insights sobre a postura de segurança de sua infraestrutura na GCP.

4. Cloud Identity-Aware Proxy (IAP): O IAP permite criar pontos de extremidade seguros para suas aplicações, restringindo o acesso com base nas políticas definidas.

Serviços de Segurança da GCP para DevOps

5. VPC Service Controls: Esse serviço fornece restrições de segurança avançadas para recursos dentro de suas redes virtuais, ajudando a proteger contra vazamentos de dados.

6. Cloud Key Management Service (KMS): O KMS permite criar e gerenciar chaves de criptografia para proteger dados sensíveis armazenados na GCP.

7. Cloud Security Scanner: Esse serviço automatiza a detecção de vulnerabilidades em suas aplicações da web, ajudando a protegê-las contra ameaças.

8. Cloud Armor: O Cloud Armor oferece proteção DDoS (Distributed Denial of Service) e segurança de aplicativos da web para suas aplicações na GCP.

Serviços de Segurança da GCP para DevOps

9. Cloud Audit Logs: Os registros de auditoria da GCP fornecem informações detalhadas sobre todas as atividades realizadas em sua conta.

10. Cloud Data Loss Prevention (DLP): O DLP ajuda a identificar e proteger informações confidenciais, como números de cartão de crédito e informações pessoais.

11. Cloud Monitoring (anteriormente Stackdriver): O Cloud Monitoring permite a coleta e análise de telemetria para identificar anomalias e ameaças à segurança.

12. Cloud Threat Detection: O Threat Detection utiliza machine learning para detectar ameaças e atividades maliciosas em suas cargas de trabalho na GCP.

Serviços de Segurança da GCP para DevOps

13. Cloud HSM: O Hardware Security Module (HSM) oferece proteção de chaves de criptografia em hardware dedicado.

14. Cloud Security Scanner: Essa ferramenta de análise de segurança automatizada ajuda a identificar vulnerabilidades em suas aplicações da web.

15. Google Cloud Armor: Oferece proteção de aplicativos web (WAF) e prevenção de ataques DDoS para suas aplicações.

Esses são apenas alguns dos serviços de segurança da informação fornecidos pela GCP que podem ser integrados à sua esteira de DevOps.

A escolha dos serviços dependerá das necessidades específicas do seu projeto e do nível de segurança desejado.

Importante é configurar e usar esses serviços de acordo com as melhores práticas de segurança da GCP para garantir a proteção adequada de seus recursos e dados.

Configuração da Segurança de DevOps na GCP

Passo 1: Avaliação de Riscos e Requisitos de Segurança

Antes de começar, é importante entender os requisitos de segurança específicos do seu aplicativo e avaliar os riscos associados. Isso ajudará a determinar quais medidas de segurança são necessárias em sua esteira de DevOps.

Passo 2: Identificação de Recursos na GCP

Identifique os recursos da GCP que serão usados em sua esteira de DevOps, como máquinas virtuais, contêineres, bancos de dados, etc. Liste esses recursos para acompanhar e configurar políticas de segurança.

Passo 3: Gerenciamento de Acessos

3.1. Configure o Identity and Access Management (IAM) da GCP para gerenciar identidades de acesso. Use o IAM para conceder permissões somente aos recursos necessários.

3.2. Implemente a autenticação de dois fatores (2FA) para contas de administrador e outros usuários com privilégios elevados.

Configuração da Segurança de DevOps na GCP

Passo 4: Políticas de Segurança

4.1. Defina políticas de segurança que restrinjam o acesso a recursos sensíveis. Use o Cloud Identity-Aware Proxy (IAP) e o VPC Service Controls para controlar permissões.

4.2. Implemente políticas de retenção e criptografia de dados sensíveis usando o Cloud Data Loss Prevention (DLP).

Passo 5: Proteção de Dados e Comunicações

5.1. Criptografe dados sensíveis em repouso usando o Cloud Key Management Service (KMS) e o Google Cloud Storage Encryption.

5.2. Use HTTPS para comunicações seguras entre os componentes da esteira de DevOps.

Passo 6: Monitoramento e Auditoria

6.1. Configure logs de auditoria usando o Cloud Audit Logs e o Stackdriver Logging para registrar todas as atividades em sua conta GCP.

6.2. Implemente alertas usando o Stackdriver Monitoring para monitorar eventos suspeitos e atividades não autorizadas.

Configuração da Segurança de DevOps na GCP

Passo 7: Segurança na Infraestrutura

7.1. Mantenha sistemas operacionais, aplicativos e bibliotecas atualizados com patches de segurança.

7.2. Utilize o Google Cloud Armor para proteção contra ataques DDoS e o Cloud Security Scanner para varreduras de segurança automatizadas.

Passo 8: Integração Contínua/Entrega Contínua (CI/CD)

8.1. Configure um serviço de CI/CD, como o Google Cloud Build, para automatizar compilações e implantações seguras.

8.2. Implemente práticas de segurança no pipeline de CI/CD, como verificação de vulnerabilidades no código-fonte e nos contêineres.

Passo 9: Testes de Segurança Automatizados

9.1. Integre ferramentas de teste de segurança automatizado, como o Google Cloud Security Scanner e o Google Cloud Armor, em seu pipeline de CI/CD.

9.2. Configure uma política de "negação padrão" para recursos, permitindo apenas o tráfego necessário.

Configuração da Segurança de DevOps na GCP

Passo 9: Testes de Segurança Automatizados

9.1. Integre ferramentas de teste de segurança automatizado, como o Google Cloud Security Scanner e o Google Cloud Armor, em seu pipeline de CI/CD.

9.2. Configure uma política de "negação padrão" para recursos, permitindo apenas o tráfego necessário.

Passo 10: Documentação e Treinamento

10.1. Documente todas as políticas de segurança, procedimentos e configurações.

10.2. Forneça treinamento de segurança para a equipe de DevOps e outros envolvidos na esteira.

Configuração da Segurança de DevOps na GCP

Passo 11: Monitoramento Contínuo e Melhorias

11.1. Estabeleça monitoramento contínuo e revise periodicamente as políticas e controles de segurança para fazer melhorias.

11.2. Esteja preparado para responder a incidentes de segurança com um plano de resposta a incidentes.

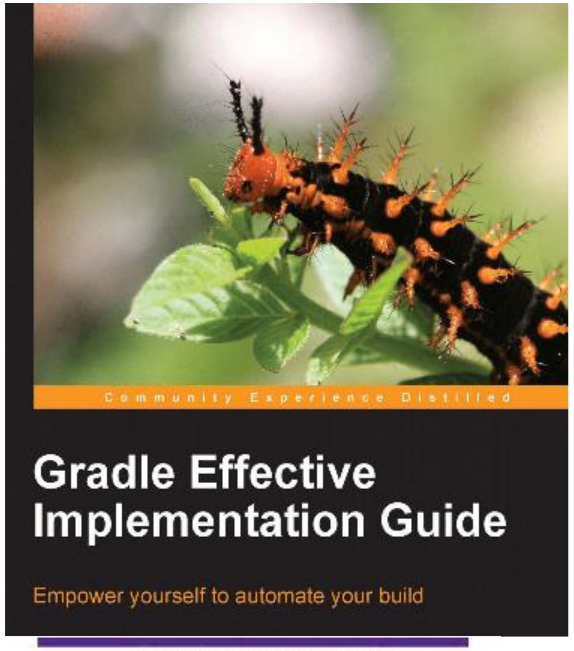
A segurança em uma esteira de DevOps na GCP é um processo contínuo.

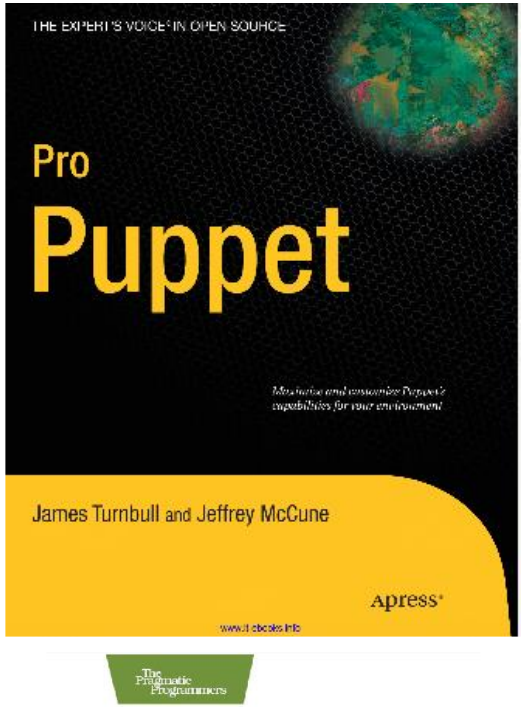
Ameaças e requisitos podem evoluir ao longo do tempo, portanto, é essencial manter-se atualizado e ajustar suas medidas de segurança conforme necessário, seguindo as melhores práticas de segurança da GCP.

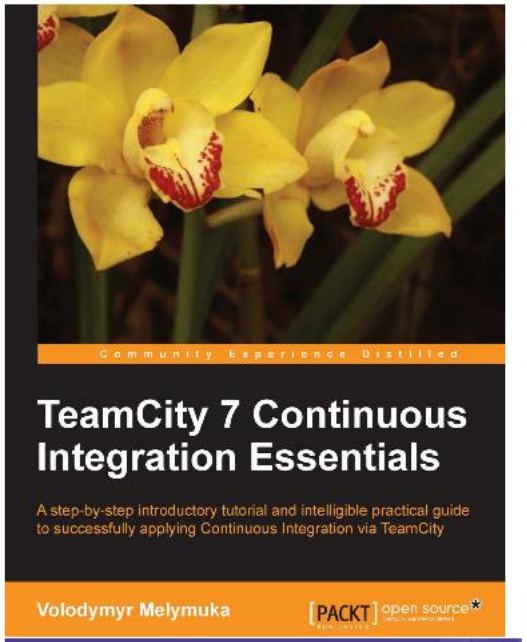
Tutoriais GCP DevOPs Opcionais

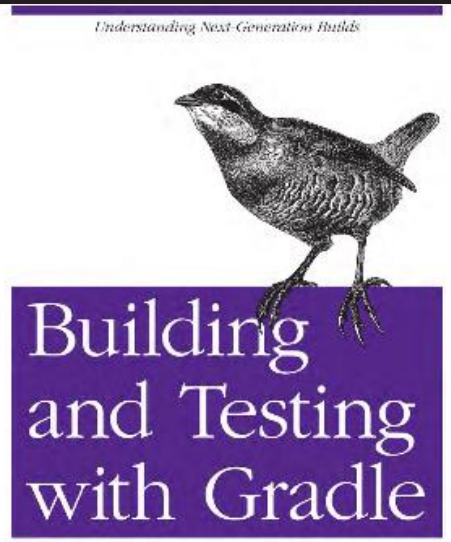
1. Continuous deployment to Google Kubernetes Engine (GKE) with Cloud Build: https://codelabs.developers.google.com/codelabs/cloud-builder-gke-continuous-deploy#0
2. Armazenar pacotes Java no Artifact Registry: https://cloud.google.com/artifact-registry/docs/java/store-java?hl=pt-br
3. Armazenar imagens de contêiner do Docker no Artifact Registry: https://cloud.google.com/artifact-registry/docs/docker/store-docker-container-images?hl=pt-br
4. Como usar builders de contribuições da comunidade e builders personalizados: https://cloud.google.com/build/docs/configuring-builds/use-community-and-custom-builders?hl=pt-br
5. App Dev: Setting up a Development Environment – Python: https://www.cloudskillsboost.google/focuses/1074?parent=catalog
6. App Dev: Deploying the Application into Kubernetes Engine – Python: https://www.cloudskillsboost.google/focuses/1073?parent=catalog
7. Como gerenciar a infraestrutura como código com o Terraform, o Cloud Build e o GitOps : https://cloud.google.com/docs/terraform/resource-management/managing-infrastructure-as-code?hl=pt-br
8. Como criar uma configuração do Terraform: https://cloud.google.com/service-catalog/docs/terraform-configuration?hl=pt-br
9. Como criar uma solução baseada no Deployment Manager: https://cloud.google.com/service-catalog/docs/dm-based-solution?hl=pt-br
10.

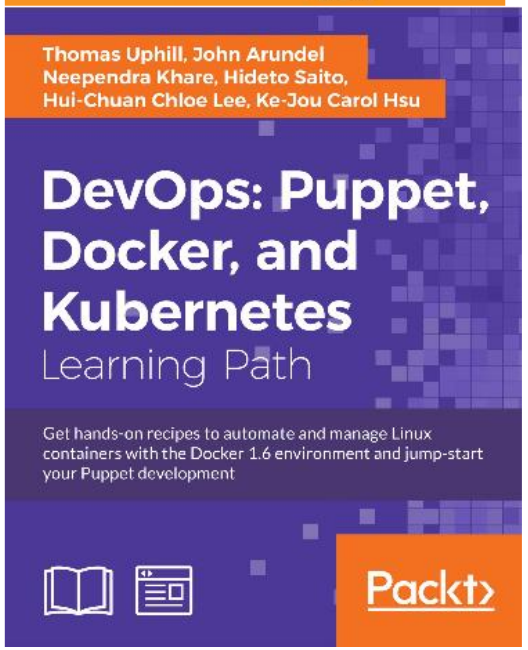
Livros de Apoio

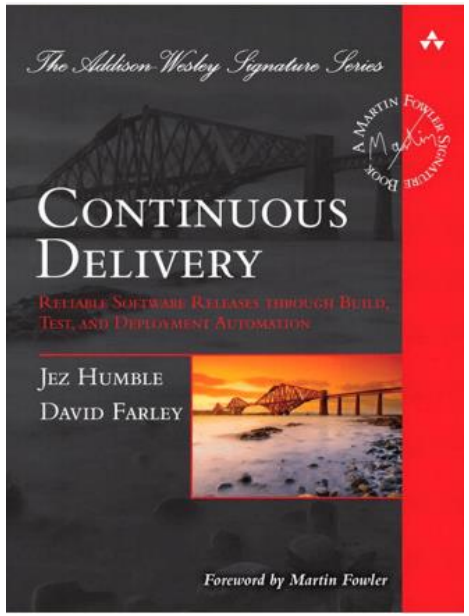












Livros: https://www.dropbox.com/scl/fo/jwsjcjw1ne0a6t5q7n4z0/h?dl=0&rlkey=qz3uufrts4mb599wmbp34bazm

OBRIGADO!


FIAP