Whitepaper da AWS

Introdução ao DevOps na AWS

Introdução ao DevOps na AWS: Whitepaper da AWS

Table of Contents

Introdução ao DevOps na AWS

Data de publicação: 16 de outubro de 2020 (Revisões do documento)

Resumo

Hoje, mais do que nunca, as empresas estão embarcando em sua jornada de transformação digital para construir conexões mais profundas com seus clientes e alcançar valor empresarial sustentável e duradouro. Organizações de todos os formatos e portes estão deixando os concorrentes para trás e entrando em novos mercados inovando mais rapidamente do que nunca. Para essas organizações, é importante se concentrar na inovação e na ruptura de software, sendo a simplificação um item essencial para a entrega do software. As organizações que reduzem o tempo da ideia à produção, tornando a velocidade e a agilidade uma prioridade, podem ser as revolucionárias do futuro.

Embora haja vários fatores a serem considerados para se tornar o próximo revolucionário digital, este whitepaper se concentra em DevOps e nos serviços e recursos da plataforma da AWS que ajudarão a aumentar a capacidade de uma organização de entregar aplicações e serviços em alta velocidade.

Introdução

DevOps é a combinação de práticas e padrões culturais e de engenharia e ferramentas que aumentam a capacidade de uma organização de fornecer aplicações e serviços em alta velocidade e melhor qualidade. Com o tempo, várias práticas essenciais surgiram ao adotar o DevOps: integração contínua, entrega contínua, infraestrutura como código e monitoramento e registro.

Este documento destaca os recursos da AWS que ajudam a acelerar sua jornada de DevOps e como os serviços da AWS podem ajudar a remover o trabalho pesado indiferenciado associado à adaptação do DevOps. Também destacamos como criar uma capacidade de integração e entrega contínuas sem gerenciar servidores ou nós de compilação, e como aproveitar a Infraestrutura como Código para provisionar e gerenciar seus recursos de nuvem de maneira consistente e repetível.

- Integração contínua: é uma prática de desenvolvimento de software em que os desenvolvedores, com frequência, juntam suas alterações de código em um repositório central. Depois disso, criações e testes são executados.
- Entrega contínua: é uma prática de desenvolvimento de software em que alterações de código são criadas, testadas e preparadas automaticamente para liberação para produção.
- Infraestrutura como código: é uma prática em que a infraestrutura é provisionada e gerenciada usando técnicas de desenvolvimento de código e software, como controle de versão e integração contínua.
- Monitoramento e registro: permite que as empresas vejam como a performance da aplicação e da infraestrutura afeta a experiência do usuário final do seu produto.
- Comunicação e colaboração: práticas são estabelecidas para aproximar as equipes e para criar fluxos de trabalho e distribuir as responsabilidades para DevOps.
- Segurança: deve ser uma preocupação transversal. Seus pipelines de integração e entrega contínua (CI/CD) e serviços relacionados devem ser protegidos e as permissões de controle de acesso adequadas devem ser configuradas.

Um exame de cada um desses princípios revela uma conexão estreita com as ofertas disponíveis da Amazon Web Services (AWS).

Integração contínua

Integração contínua (CI) é uma prática de desenvolvimento de software em que os desenvolvedores mesclam regularmente alterações de código em um repositório central para permitir a execução de compilações e testes automáticos. A CI ajuda a encontrar e resolver bugs mais rapidamente, melhorar a qualidade do software e reduzir o tempo necessário para validar e lançar novas atualizações de software.

A AWS oferece os seguintes serviços para integração contínua:

Tópicos

- AWS CodeCommit
- AWS CodeBuild
- AWS CodeArtifact

AWS CodeCommit

O AWS CodeCommit é um serviço de controle de código-fonte seguro, altamente escalável e gerenciado que hospeda repositórios git privados. O CodeCommit elimina a necessidade de operar seu próprio sistema de controle de origem, sem provisionamento e escalonamento de hardware ou instalação, configuração e operação de software. Você pode usar o CodeCommit para armazenar qualquer coisa, de código a binários, e ele oferece suporte à funcionalidade padrão do Git, permitindo que ele funcione perfeitamente com suas ferramentas baseadas em Git existentes. Sua equipe pode ainda usar as ferramentas online do CodeCommit para procurar, editar e colaborar em projetos. O AWS CodeCommit tem vários benefícios:

Colaboração: o AWS CodeCommit foi projetado para o desenvolvimento colaborativo de software. Você pode facilmente confirmar, detectar diferenças e mesclar o código para facilitar o controle dos projetos da equipe. O CodeCommit também oferece suporte a solicitações de pull, que proporciona um mecanismo para solicitar revisões de código e discutir código com colaboradores.

Criptografia: você pode transferir seus arquivos do AWS CodeCommit usando HTTPS e SSH, como preferir. Seus repositórios também são criptografados automaticamente em repouso por meio de AWS Key Management Service (AWS KMS) usando chaves específicas do cliente.

Controle de acesso: o AWS CodeCommit usa o AWS Identity and Access Management (IAM) para controlar e monitorar quem pode acessar seus dados, além de como, quando e onde eles podem ser

acessados. O CodeCommit também ajuda a monitorar seus repositórios por meio do AWS CloudTrail e do Amazon CloudWatch.

Alta disponibilidade e durabilidade: O AWS CodeCommit armazena seus repositórios no Amazon Simple Storage Service (Amazon S3) e no Amazon DynamoDB. Seus dados criptografados são armazenados de modo redundante em várias instalações. Essa arquitetura aumenta a disponibilidade e a durabilidade dos dados do repositório.

Notificações e scripts personalizados: agora você pode receber notificações de eventos que afetam seus repositórios. As notificações virão como notificações do Amazon Simple Notification Service (Amazon SNS). Cada notificação incluirá uma mensagem de status e um link para os recursos do evento que gerou essa notificação. Além disso, ao usar acionadores de repositório do AWS CodeCommit, você poderá enviar notificações e criar webhooks HTTP com Amazon SNS ou invocar funções do AWS Lambda em resposta aos eventos de repositório selecionados.

AWS CodeBuild

AWS CodeBuild é um serviço de integração contínua e totalmente gerenciado que compila o código-fonte, executa testes e produz pacotes de software prontos para implantação. Você não precisa provisionar, gerenciar e dimensionar seus próprios servidores de compilação. O CodeBuild pode usar GitHub, GitHub Enterprise, BitBucket, AWS CodeCommit ou Amazon S3 como provedor de origem.

O CodeBuild escala continuamente e pode processar várias compilações simultaneamente. O CodeBuild oferece vários ambientes pré-configurados para várias versões do Microsoft Windows e Linux. Os clientes também podem trazer seus ambientes de compilação personalizados como contêineres do Docker. O CodeBuild também se integra a ferramentas de código aberto, como Jenkins e Spinnaker.

O CodeBuild também pode criar relatórios para testes unitários, funcionais ou de integração. Esses relatórios fornecem uma visão de quantos casos de teste foram executados e quantos foram aprovados ou reprovados. O processo de compilação também pode ser executado dentro de uma Amazon Virtual Private Cloud (Amazon VPC), o que pode ser útil se seus serviços de integração ou bancos de dados forem implantados dentro de uma VPC.

AWS CodeArtifact

O AWS CodeArtifact é um serviço de repositório de artefatos totalmente gerenciado que pode ser usado por organizações para armazenar, publicar e compartilhar com segurança pacotes de

software usados no processo de desenvolvimento de software. O CodeArtifact pode ser configurado para obter automaticamente pacotes de software e dependências de repositórios públicos de artefatos para que os desenvolvedores tenham acesso às versões mais recentes.

As equipes de desenvolvimento de software estão cada vez mais confiando em pacotes de código aberto para realizar tarefas comuns em seus pacotes de aplicações. Agora, tornou-se fundamental para as equipes de desenvolvimento de software manter o controle sobre uma versão específica do software de código aberto livre de vulnerabilidades. Com o CodeArtifact, você pode configurar controles para que isso aconteça.

O CodeArtifact funciona com gerenciadores de pacotes comumente usados e ferramentas de compilação como Maven, Gradle, npm, yarn, twine e pip, o que facilita a integração em fluxos de trabalho de desenvolvimento existentes.

Entrega contínua

A entrega contínua é uma prática de desenvolvimento de software na qual as alterações de código são automaticamente preparadas para uma liberação para produção. Um pilar do desenvolvimento de aplicações modernas, a entrega contínua expande com base na integração contínua implantando todas as alterações de código em um ambiente de teste e/ou ambiente de produção, após o estágio de criação. Quando implementada adequadamente, os desenvolvedores sempre terão um artefato de compilação pronto para implantação que já passou por um processo de teste padronizado.

A entrega contínua permite que os desenvolvedores automatizem testes que vão além dos testes de unidade, de forma que seja possível verificar atualizações de aplicações em várias dimensões antes de implantá-las para os clientes. Esses testes podem incluir testes de IU, carga, integração, confiabilidade de API, etc. Isso ajuda os desenvolvedores a validar atualizações com maior precisão e a descobrir problemas de modo preventivo. Com a nuvem, é fácil e econômico automatizar a criação e a replicação de vários ambientes de teste, o que, no passado, era difícil de fazer on-premises.

A AWS oferece os seguintes serviços para entrega contínua:

- AWS CodeBuild
- AWS CodeDeploy
- AWS CodePipeline

Tópicos

AWS CodeDeploy

AWS CodeDeploy é um serviço de implantação totalmente gerenciado que automatiza implantações de software para uma variedade de serviços computacionais como Amazon Elastic Compute Cloud (Amazon EC2), AWS Fargate, AWS Lambda, e seus servidores on-premises. O AWS CodeDeploy facilita o rápido lançamento de novos recursos, ajuda a evitar inatividade durante a implantação da aplicação, além de lidar com a complexidade das atualizações de suas aplicações. Use o CodeDeploy para automatizar implantações de software, dispensando operações manuais

propensas a erros. O serviço é escalado para estar de acordo com as suas necessidades de implantação.

O CodeDeploy tem vários benefícios que se alinham ao princípio DevOps de implantação contínua:

Implantações automatizadas: O CodeDeploy automatiza completamente as implantações de software, o que permite implantar de forma confiável e rápida.

Controle centralizado: O CodeDeploy permite que você inicie e monitore facilmente o status das implantações de aplicações usando o Console de Gerenciamento da AWS ou a AWS CLI. O CodeDeploy oferece um relatório detalhado que permite ver quando e onde cada revisão de aplicação foi implantada. Além disso, você pode criar notificações por push para receber atualizações em tempo real sobre as implantações.

Inatividade minimizada: O CodeDeploy ajuda a maximizar a disponibilidade de aplicações durante o processo de implantação do software. O serviço apresenta alterações de forma incremental e rastreia a integridade das aplicações de acordo com regras configuráveis. As implantações de software poderão ser facilmente interrompidas e revertidas se houver erros.

Fácil de adotar: o CodeDeploy funciona com qualquer aplicação e fornece a mesma experiência em diferentes plataformas e linguagens. Você pode reutilizar facilmente seu código de configuração atual. O CodeDeploy também pode ser integrado ao seu processo de lançamento de software ou cadeia de ferramentas de entrega contínua atual (por exemplo, AWS CodePipeline, GitHub e Jenkins).

O AWS CodeDeploy oferece suporte a várias opções de implantação. Para obter mais informações, consulte Estratégias de implantação.

AWS CodePipeline

OAWS CodePipeline é um serviço de entrega contínua que permite que você modele, visualize e automatize as etapas necessárias para lançar seu software. Com o AWS CodePipeline, você modela o processo de lançamento completo para criar seu código, implantar em ambientes de pré-produção, testar sua aplicação e liberá-la para produção. O AWS CodePipeline então cria, testa e implanta sua aplicação de acordo com o fluxo de trabalho definido sempre que houver uma alteração de código. Você pode integrar ferramentas de parceiros APN, além das suas próprias ferramentas personalizadas, em qualquer etapa do processo de liberação para compor uma solução de entrega contínua completa.

O AWS CodePipeline tem vários benefícios que se alinham com o princípio DevOps de implantação contínua:

Entrega rápida: O AWS CodePipeline automatiza o processo de liberação de software, o que agiliza a liberação de novos recursos para os usuários. O CodePipeline permite que você reaja rapidamente ao feedback e acelere a disponibilização de novos recursos aos usuários.

Qualidade aprimorada: Ao automatizar seus processos de compilação, teste e liberação, o AWS CodePipeline permite o aumento da velocidade e a qualidade das atualizações de software, submetendo todas as novas alterações a um conjunto consistente de verificações de qualidade.

Fácil integração: O AWS CodePipeline pode ser facilmente ampliado para se adaptar a necessidades específicas. Você pode usar nossos plugins predefinidos ou seus próprios plugins personalizados em qualquer etapa do processo de liberação. Por exemplo, é possível extrair o código-fonte do GitHub, usar o seu servidor de compilação Jenkins on-premises, executar testes de carga usando um serviço de terceiros ou passar informações de implantação ao seu painel de operações personalizado.

Fluxo de trabalho configurável: o AWS CodePipeline permite modelar os diferentes estágios do processo de liberação de software usando a interface do console, a AWS CLI, o AWS CloudFormation ou os SDKs da AWS. Você pode facilmente especificar os testes a serem executados e personalizar as etapas para implantar uma aplicação e suas dependências.

Estratégias de implantação

As estratégias de implantação definem como você deseja entregar seu software. As organizações seguem diferentes estratégias de implantação com base em seu modelo de negócios. Alguns podem escolher fornecer software totalmente testado, enquanto outros podem querer feedback dos usuários e permitir que eles avaliem os recursos em desenvolvimento (por exemplo, versões beta). Na seção a seguir, falaremos sobre várias estratégias de implantação.

Tópicos

Implantações no local

Nesta estratégia, a implantação é feita com a aplicação em cada instância no grupo de implantação interrompida, a última revisão da aplicação é instalada, e a nova versão da aplicação é iniciada e validada. Você pode usar um balanceador de carga de forma que cada instância é cancelada durante sua implantação e restaurada para o serviço após a conclusão da implantação. As implantações no local podem ser feitas de uma só vez, supondo uma interrupção do serviço ou feitas como uma atualização contínua. O AWS CodeDeploy e o AWS Elastic Beanstalk oferecem configurações de implantação para uma de cada vez, metade de cada vez e todas de uma vez. Essas mesmas estratégias de implantação para implantações no local estão disponíveis em implantações azul/verde.

Implantações azul/verde

A implantação azul/verde, às vezes chamada de vermelho-preto, é uma técnica para liberar aplicações pelo tráfego de deslocamento entre dois ambientes idênticos executando versões diferentes da aplicação. As implantações azul/verde ajudam a minimizar o tempo de inatividade durante atualizações de aplicações, reduzindo os riscos relacionados ao tempo de inatividade e à funcionalidade de reversão. As implantações azul/verde permitem que você inicie uma nova versão

(verde) de sua aplicação junto com a versão antiga (azul) e monitore e teste a nova versão antes de redirecionar o tráfego para ela, revertendo na detecção de problemas.

Implantações Canário

O tráfego é deslocado em dois incrementos. Uma implantação canary é uma estratégia azul/verde mais avessa ao risco, na qual uma abordagem em fases é usada. Isso pode acontecer em duas etapas ou de forma linear, em que o novo código de aplicação é implantado e exposto para avaliação e, após a aceitação, implantado no restante do ambiente ou de forma linear.

Implantações lineares

Implantações lineares significam que o tráfego é deslocado em incrementos iguais com um número igual de minutos entre cada incremento. Você pode escolher entre opções lineares predefinidas que especificam a porcentagem de tráfego deslocado em cada incremento e o número de minutos entre cada incremento.

Implantações de uma só vez

As implantações de uma só vez significam que todo o tráfego é transferido do ambiente original para o ambiente substituto ao mesmo tempo.

Matriz de estratégias de implantação

A matriz a seguir lista as estratégias de implantação com suporte para o Amazon Elastic Container Service (Amazon ECS), AWS Lambda e Amazon EC2/On-Premise.

- O Amazon ECS é um serviço de orquestração totalmente gerenciado.
- O AWS Lambda permite que você execute código sem provisionar ou gerenciar servidores.
- O Amazon EC2 permite que você execute capacidade computacional segura e redimensionável na nuvem.

	A	B	C	D
1	Matriz de estratégias de implantação	Amazon ECS	AWS Lambda	Amazon EC2/ On-premise
2	No local	✓	✓	✓
3	Azul/Verde	✓	✓	✓*
4	Canary	✓	✓	X
5	Linear	✓	✓	X
6	Tudo de uma vez	✓	✓	X

> **Note**
>
> A implantação azul/verde com o EC2/On-premise só funciona com instâncias do EC2.

Estratégias de implantação do AWS Elastic Beanstalk

O AWS Elastic Beanstalk oferece suporte aos seguintes tipos de estratégias de implantação:

- All-at-once (Tudo de uma vez): executa a implantação no local em todas as instâncias.
- Rolling (Rolagem): divide as instâncias em lotes e implanta em um lote de cada vez.
- Rolling with Additional Batch (Rolagem com lote adicional): divide as implantações em lotes, mas, para o primeiro lote, cria novas instâncias do EC2 em vez de implantar nas instâncias EC2 existentes.
- Immutable (Imutável): se você precisar implantar com uma nova instância em vez de usar uma instância existente.
- Traffic Splitting (Divisão de tráfego): executa uma implantação imutável e encaminha a porcentagem de tráfego para as novas instâncias por um período predeterminado. Se as instâncias permanecerem íntegras, encaminhe todo o tráfego para novas instâncias e encerre as instâncias antigas.

Infraestrutura como código

Um princípio fundamental de DevOps é tratar a infraestrutura da mesma maneira que os desenvolvedores tratam o código. O código da aplicação tem um formato e uma sintaxe definidos. Se o código não for escrito de acordo com as regras da linguagem de programação, as aplicações não poderão ser criadas. O código é armazenado em um sistema de gerenciamento de versão ou controle de origem que registra um histórico de desenvolvimento, alterações e correções de bugs de código. Quando o código é compilado ou incorporado em aplicações, esperamos que uma aplicação consistente seja criada e a compilação seja repetível e confiável.

Praticar a infraestrutura como código significa aplicar o mesmo rigor do desenvolvimento de código de aplicação ao provisionamento de infraestrutura. Todas as configurações devem ser definidas de forma declarativa e armazenadas em um sistema de controle de origem, como AWS CodeCommit, o mesmo que o código da aplicação. O provisionamento, a orquestração e a implantação da infraestrutura também devem oferecer suporte ao uso da infraestrutura como código.

Tradicionalmente, a infraestrutura era provisionada usando uma combinação de scripts e processos manuais. Às vezes, esses scripts eram armazenados em sistemas de controle de versão ou documentados detalhadamente em arquivos de texto ou runbooks. Muitas vezes, a pessoa que escreve os runbooks não é a mesma pessoa que executa esses scripts ou acompanha os runbooks. Se esses scripts ou runbooks não forem atualizados com frequência, eles têm o potencial de se tornar um obstáculo nas implantações. Isso faz com que a criação de novos ambientes nem sempre seja repetível, confiável ou consistente.

Em contraste com o anterior, a AWS fornece uma maneira focada em DevOps de criar e manter a infraestrutura. Semelhante à maneira como os desenvolvedores de software escrevem o código da aplicação, a AWS fornece serviços que permitem a criação, a implantação e a manutenção da infraestrutura de maneira programática, descritiva e declarativa. Esses serviços fornecem rigor, clareza e confiabilidade. Os serviços da AWS discutidos neste documento são fundamentais para uma metodologia de DevOps e formam a base de vários princípios e práticas de DevOps da AWS de nível superior.

A AWS oferece os seguintes serviços para definir a infraestrutura como um código:

- AWS CloudFormation
- AWS Cloud Development Kit (AWS CDK)
- AWS Cloud Development Kit for Kubernetes

AWS CloudFormation

AWS CloudFormation é um serviço que permite que os desenvolvedores criem recursos da AWS de forma ordenada e previsível. Os recursos são escritos em arquivos de texto usando o formato JavaScript Object Notation (JSON) ou Yet Another Markup Language (YAML). Os modelos exigem uma sintaxe e uma estrutura específicas que dependem dos tipos de recursos que estão sendo criados e gerenciados. Você cria seus recursos em JSON ou YAML com qualquer editor de código, como o AWS Cloud9, coloca-os em um sistema de controle de versão e o CloudFormation cria os serviços especificados de maneira segura e repetível.

Um modelo do CloudFormation é implantado no ambiente da AWS como uma pilha. Você pode gerenciar pilhas por meio do Console de Gerenciamento da AWS, da Interface da Linha de Comando da AWS ou das APIs do AWS CloudFormation. Caso precise fazer alterações nos recursos em execução em uma pilha, atualize a pilha. Antes de fazer alterações nos recursos, você pode gerar um conjunto de alterações, que é o resumo das alterações propostas. Os conjuntos de alterações permitem ver como as alterações podem afetar os recursos em execução, especialmente para recursos críticos, antes de implementá-las.

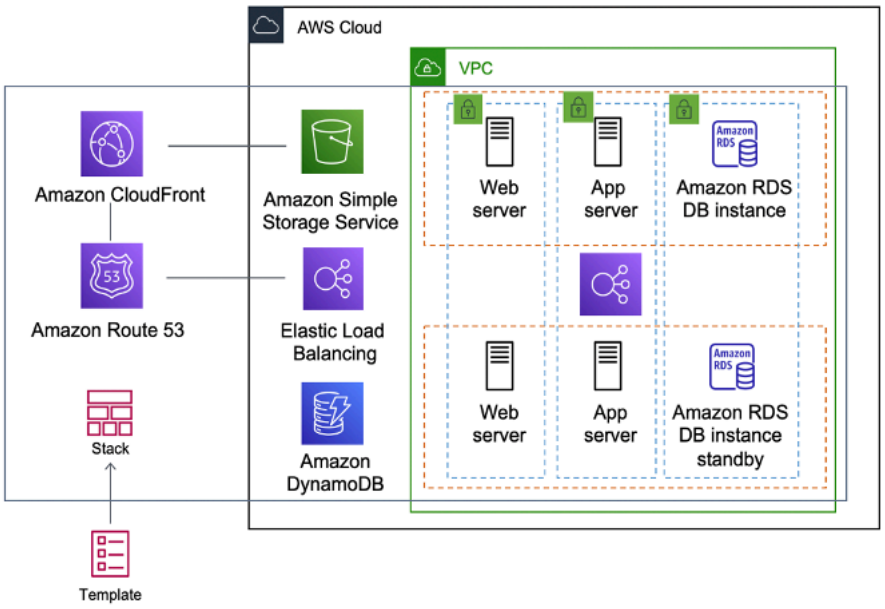

Figura 1: AWS CloudFormation criando um ambiente inteiro (pilha) a partir de um fluxo de trabalho de modelo

Você pode usar um único modelo para criar e atualizar um ambiente inteiro ou modelos separados para gerenciar várias camadas dentro de um ambiente. Isso permite que os modelos sejam modularizados e fornece uma camada de governança, que é importante para muitas organizações.

Quando você cria ou atualiza uma pilha no console, os eventos são exibidos mostrando o status da configuração. Se ocorrer um erro, por padrão, a pilha é revertida para o estado anterior. O Amazon Simple Notification Service (Amazon SNS) fornece notificações sobre eventos. Por exemplo, você pode usar o Amazon SNS para rastrear o progresso da criação e exclusão de pilhas por e-mail e integrá-lo a outros processos de forma programática.

O AWS CloudFormation facilita a organização e implantação de uma coleção de recursos da AWS e permite descrever qualquer dependência ou transmitir parâmetros especiais quando a pilha está configurada.

Com os modelos do CloudFormation, você pode trabalhar com um amplo conjunto de serviços da AWS, como Amazon S3, Auto Scaling, Amazon CloudFront, Amazon DynamoDB, Amazon EC2, Amazon ElastiCache, AWS Elastic Beanstalk, Elastic Load Balancing, IAM, AWS OpsWorks e Amazon VPC. Para obter a lista mais recente de recursos compatíveis, consulte Referência de tipos de recursos e propriedades da AWS.

AWS Cloud Development Kit

O AWS Cloud Development Kit (AWS CDK) é uma framework de desenvolvimento de software de código aberto para modelar e provisionar recursos de aplicações em nuvem usando linguagens de programação conhecidas. O AWS CDK permite que você modele a infraestrutura de aplicações usando TypeScript, Python, Java e .NET. Os desenvolvedores podem aproveitar seu ambiente de desenvolvimento integrado (IDE) existente, aproveitando ferramentas como preenchimento automático e documentação em linha para acelerar o desenvolvimento da infraestrutura.

O AWS CDK utiliza o AWS CloudFormation em segundo plano para provisionar recursos de maneira segura e repetível. Construções são a base do código CDK. Uma construção representa um componente de nuvem e encapsula tudo de que o AWS CloudFormation precisa para criar o componente. O AWS CDK inclui a AWS Construct Library contendo construções que representam muitos serviços da AWS. Ao combinar construções, você pode criar arquiteturas complexas de forma rápida e fácil para implantação na AWS.

AWS Cloud Development Kit for Kubernetes

O AWS Cloud Development Kit for Kubernetes (cdk8s) é um framework de desenvolvimento de software de código aberto para definir aplicações Kubernetes usando linguagens de programação de uso geral.

Depois de definir sua aplicação em uma linguagem de programação (a partir da data de publicação, apenas Python e TypeScript são compatíveis), o cdk8s converterá a descrição de sua aplicação em YML pré-Kubernetes. Esse arquivo YML pode ser consumido por qualquer cluster do Kubernetes em execução em qualquer lugar. Como a estrutura é definida em uma linguagem de programação, você pode usar os recursos avançados fornecidos pela linguagem de programação. Você pode usar o recurso de abstração da linguagem de programação para criar seu próprio código boiler-plate e reutilizá-lo em todas as implantações.

Automação

Outra filosofia e prática fundamentais do DevOps é a automação. A automação se concentra na instalação, configuração, implantação e suporte da infraestrutura e das aplicações executadas nela. Ao usar a automação, você pode configurar ambientes mais rapidamente, de maneira padronizada e repetível. A remoção de processos manuais é a chave para uma estratégia de DevOps de sucesso. Historicamente, a configuração do servidor e a implantação de aplicações têm sido predominantemente um processo manual. Os ambientes se tornam fora do padrão e é difícil reproduzir um ambiente quando surgem problemas.

O uso da automação é fundamental para obter todos os benefícios da nuvem. Internamente, a AWS depende muito da automação para fornecer os principais recursos de elasticidade e escalabilidade. Os processos manuais são propensos a erros, não são confiáveis e são inadequados para oferecer suporte a um negócio ágil. Uma organização pode, com frequência, fazer com que recursos altamente qualificados forneçam configuração manual, quando o tempo poderia ser melhor gasto oferecendo suporte a atividades mais essenciais e de maior valor dentro da empresa.

Os ambientes operacionais modernos geralmente dependem de automação total para eliminar a intervenção manual ou para acessar ambientes de produção. Isso inclui lançamentos de software, configuração da máquina, aplicação de patches do sistema operacional, solução de problemas ou correção de bugs. Muitos níveis de práticas de automação podem ser usados juntos para fornecer um processo automatizado de ponta a ponta de alto nível.

A automação tem os seguintes benefícios principais:

- Mudanças rápidas
- Maior produtividade
- Configurações repetíveis
- Ambientes reproduzíveis
- Elasticidade potencializada
- Escalabilidade automática potencializada
- Teste automatizado

A automação é a base dos serviços da AWS e é compatível internamente com todos os serviços, recursos e ofertas.

Tópicos

AWS OpsWorks

AWS OpsWorks leva os princípios do DevOps ainda mais longe do que o AWS Elastic Beanstalk. Ele pode ser considerado um serviço de gerenciamento de aplicações, não simplesmente um contêiner de aplicações. O AWS OpsWorks oferece ainda mais níveis de automação com recursos adicionais, como integração com o software de gerenciamento de configuração (Chef) e gerenciamento do ciclo de vida da aplicação. Você pode usar o gerenciamento do ciclo de vida da aplicação para definir quando os recursos são instalados, configurados, implantados, desimplantados ou encerrados.

Para maior flexibilidade, o AWS OpsWorks faz com que você defina sua aplicação em pilhas configuráveis. Você também pode selecionar pilhas de aplicações predefinidas. As pilhas de aplicação contêm todo o provisionamento de recursos da AWS que seu aplicativo requer, incluindo servidores de aplicações, servidores web, bancos de dados e balanceadores de carga.

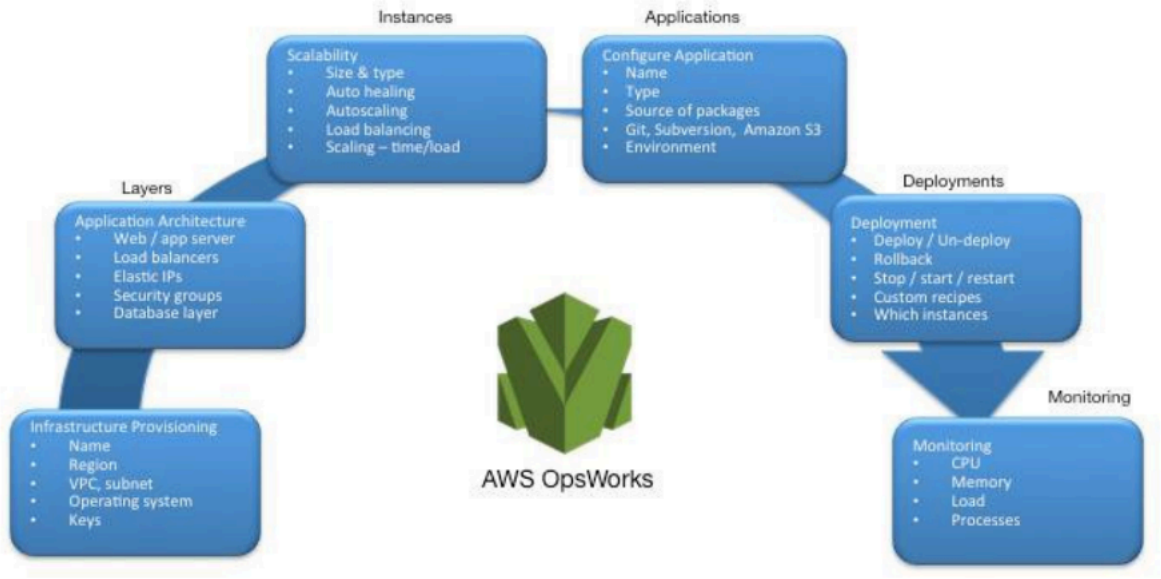

Figura 2 - AWS OpsWorks mostrando os recursos e a arquitetura do DevOps

As pilhas de aplicações são organizadas em camadas arquitetônicas para que as pilhas possam ser mantidas de forma independente. Camadas de exemplo podem incluir camada da web, camada de aplicação e camada de banco de dados. Fora da caixa, o AWS OpsWorks também

simplifica a configuração de grupos do Auto Scaling e balanceadores de carga do Elastic Load Balancing, ilustrando ainda mais o princípio de automação do DevOps. Assim como o AWS Elastic Beanstalk, o AWS OpsWorks oferece suporte ao versionamento da aplicação, implantação contínua e gerenciamento de configuração de infraestrutura.

O AWS OpsWorks também oferece suporte às práticas de DevOps de monitoramento e registro (abordadas na próxima seção). O suporte ao monitoramento é fornecido pelo Amazon CloudWatch. Todos os eventos do ciclo de vida são registrados, e um log separado do Chef documenta todas as receitas do Chef que são executadas, juntamente com quaisquer exceções.

AWS Elastic Beanstalk

O AWS Elastic Beanstalk é um serviço para rápida implantação e escalabilidade de aplicações web desenvolvidas com Java, .NET, PHP, Node.js, Python, Ruby, Go e Docker em servidores familiares, como Apache, Nginx, Passenger e IIS.

O Elastic Beanstalk é uma abstração sobre o Amazon EC2, Auto Scaling, e simplifica a implantação fornecendo recursos adicionais, como clonagem, implantações azul/verde, Elastic Beanstalk Command Line Interface (eb cli) e integração com o AWS Toolkit for Visual Studio, Visual Studio Code, Eclipse e IntelliJ para aumentar a produtividade do desenvolvedor.

Monitoramento e registro em log

Comunicação e colaboração são fundamentais em uma filosofia de DevOps. Para facilitar isso, o feedback é essencial. Na AWS, o feedback é fornecido por dois serviços principais: Amazon CloudWatch e AWS CloudTrail. Juntos, eles fornecem uma infraestrutura robusta de monitoramento, alertas e auditoria para que desenvolvedores e equipes de operações possam trabalhar juntos de forma próxima e transparente.

A AWS fornece os seguintes serviços para monitoramento e registro:

Tópicos

Amazon CloudWatch

As métricas do Amazon CloudWatch coletam automaticamente dados de serviços da AWS, como instâncias do Amazon EC2, volumes do Amazon EBS e instâncias de banco de dados do Amazon RDS. Essas métricas podem, então, ser organizadas como painéis e alarmes ou eventos podem ser criados para acionar eventos ou executar ações de Auto Scaling.

Alarmes do Amazon CloudWatch

Você pode configurar alarmes com base nas métricas coletadas pelo Amazon CloudWatch Metrics. O alarme pode, então, enviar uma notificação para o tópico do Amazon Simple Notification Service (Amazon SNS) ou iniciar ações de Auto Scaling. Um alarme requer período (duração da avaliação de uma métrica), período de avaliação (número dos pontos de dados mais recentes) e pontos de dados para alarme (número de pontos de dados dentro do período de avaliação).

Amazon CloudWatch Logs

Amazon CloudWatch Logs é um serviço de agregação e monitoramento de logs. AWS CodeBuild, CodeCommit, CodeDeploy e CodePipeline fornecem integrações com o CloudWatch Logs para que todos os logs possam ser monitorados centralmente. Além dos serviços mencionados anteriormente, vários outros serviços da AWS fornecem integração direta com o CloudWatch.

Com o CloudWatch Logs, você pode:

- Consultar seus dados de log
- Monitorar logs de instâncias do Amazon EC2
- Monitorar eventos registrados do AWS CloudTrail
- Definir a política de retenção de logs

Amazon CloudWatch Logs Insights

O Amazon CloudWatch Logs Insights verifica seus logs e permite que você realize consultas e visualizações interativas. Ele compreende vários formatos de log e descobre automaticamente campos de logs JSON.

Amazon CloudWatch Events

O Amazon CloudWatch Events oferece uma transmissão quase em tempo real de eventos do sistema que descrevem mudanças nos recursos da AWS. Com regras simples que você pode configurar rapidamente, é possível corresponder eventos e roteá-los para uma ou mais funções ou transmissões. O CloudWatch Events fica ciente das mudanças operacionais à medida que elas ocorrem. O CloudWatch Events responde a essas alterações operacionais e executa a ação corretiva conforme necessário, enviando mensagens para responder ao ambiente, ativando funções, fazendo alterações e capturando informações de estado.

Você pode configurar regras no CloudWatch Events para alertá-lo sobre alterações nos serviços da AWS e integrar esses eventos a outros sistemas de terceiros usando o Amazon EventBridge. Veja a seguir os serviços relacionados ao AWS DevOps que têm integração com o CloudWatch Events.

- Eventos do Application Auto Scaling
- Eventos do CodeBuild

- Eventos do CodeCommit
- Eventos do CodeDeploy
- Eventos do CodePipeline

Amazon EventBridge

O Amazon CloudWatch Events e o EventBridge são o mesmo serviço subjacente e API, mas o EventBridge oferece mais recursos.

O Amazon EventBridge é um barramento de eventos sem servidor que permite integrações entre serviços da AWS, software como serviço (SaaS) e suas aplicações. Além de criar aplicações orientadas a eventos, o EventBridge pode ser usado para notificar sobre os eventos de serviços como CodeBuild, CodeDeploy, CodePipeline e CodeCommit.

AWS CloudTrail

Para adotar os princípios de colaboração, comunicação e transparência do DevOps, é importante entender quem está fazendo modificações em sua infraestrutura. Na AWS, essa transparência é fornecida pelo serviço AWS CloudTrail. Todas as interações da AWS são tratadas por meio de chamadas de API da AWS que são monitoradas e registradas pelo AWS CloudTrail. Todos os arquivos de log gerados são armazenados em um bucket do Amazon S3 definido por você. Os arquivos de log são criptografados usando a criptografia no lado do servidor (SSE) do Amazon S3. Todas as chamadas de API são registradas, quer venham diretamente de um usuário ou em nome de um usuário por meio de um serviço da AWS. Vários grupos podem se beneficiar dos logs do CloudTrail, incluindo equipes de operações para suporte, equipes de segurança para governança e equipes financeiras para faturamento.

Comunicação e colaboração

Se você está adotando a cultura de DevOps em sua organização ou passando por uma comunicação de transformação da cultura de DevOps, a colaboração é uma parte importante de sua abordagem. Na Amazon, percebemos que é necessário mudar a mentalidade das equipes, por isso adotamos o conceito de Equipes de duas pizzas.

Tópicos

Equipes de duas pizzas

“Tentamos criar equipes que tenham a quantidade de pessoas que possam ser alimentadas por duas pizzas”, disse Bezos. “Chamamos isso de regra da equipe de duas pizzas.”

Quanto menor a equipe, melhor a colaboração. A colaboração também é muito importante, pois os lançamentos de software estão acontecendo mais rápido do que nunca. E a capacidade de uma equipe de entregar o software pode ser um diferencial para sua organização em relação à concorrência. Imagine uma situação em que um novo recurso de produto precisa ser lançado ou um bug precisa ser corrigido. Você deseja que isso aconteça o mais rápido possível para que você possa ter um tempo menor de entrada no mercado. Isso também é importante porque você não quer que a transformação seja um processo lento, mas sim uma abordagem ágil, onde ondas de mudanças começam a causar impacto.

A comunicação entre as equipes também é importante enquanto avançamos em direção ao modelo de responsabilidade compartilhada e começamos a sair da abordagem de desenvolvimento em silos. Isso traz o conceito de propriedade na equipe e muda a perspectiva dos membros, para que encararem isso como algo de ponto a ponta. Sua equipe não deve pensar nos ambientes de produção como caixas pretas das quais eles não tem visibilidade.

A transformação cultural também é importante, pois você pode estar construindo uma equipe comum de DevOps ou, em outra abordagem, um ou mais membros da equipe estão focados em DevOps. Ambas as abordagens apresentam responsabilidade compartilhada na equipe.

Segurança

Se você está passando por uma transformação de DevOps ou implementando princípios de DevOps pela primeira vez, você deve pensar na segurança como integrada em seus processos de DevOps. Isso deve ser uma preocupação em todos os estágios de construção, teste e implantação.

Antes de falarmos sobre segurança em DevOps na AWS, vamos dar uma olhada no Modelo de responsabilidade compartilhada da AWS.

Tópicos

Modelo de responsabilidade compartilhada da AWS

A segurança é uma responsabilidade compartilhada entre a AWS e o cliente. As diferentes partes do Modelo de responsabilidade compartilhada estão explicadas abaixo:

- AWS responsibility “Security of the Cloud” (Responsabilidade da AWS: “Segurança da nuvem”: a AWS é responsável por proteger a infraestrutura que executa todos os serviços oferecidos na Nuvem AWS. Essa infraestrutura abrange o hardware, o software, as redes e as instalações que executam os serviços da Nuvem AWS.
- Customer responsibility “Security in the Cloud” (Responsabilidade do cliente “Segurança na nuvem”): a responsabilidade do cliente será determinada pelos serviços da Nuvem AWS selecionados por ele. Isso define o volume do trabalho de configuração que o cliente deve executar como parte de suas responsabilidades de segurança.

Esse modelo compartilhado pode auxiliar a reduzir os encargos operacionais do cliente, pois a AWS opera, gerencia e controla os componentes do sistema operacional do host e a camada de virtualização, incluindo a segurança física das instalações em que o serviço opera. Isso é fundamental nos casos em que o cliente deseja entender a segurança de seus ambientes de construção.

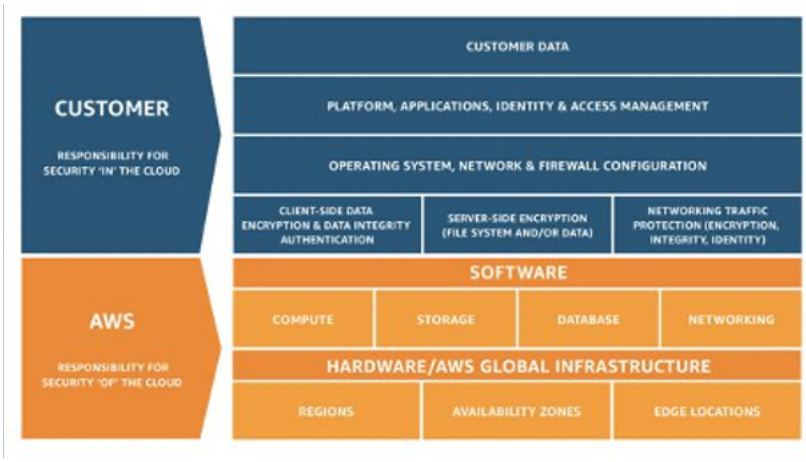

Figura 3: Modelo de responsabilidade compartilhada da AWS

Identity and Access Management

AWS Identity and Access Management (IAM) define os controles e as políticas usadas para gerenciar o acesso aos recursos da AWS. Com o IAM, você pode criar usuários e grupos e definir permissões para vários serviços de DevOps.

Além dos usuários, vários serviços também podem precisar de acesso aos recursos da AWS. Por exemplo, seu projeto do CodeBuild pode precisar de acesso para armazenar imagens do Docker no Amazon Elastic Container Registry (Amazon ECR) e precisar de permissões para gravar no Amazon ECR. Esses tipos de permissões são definidos por uma função de tipo especial conhecida como função de serviço.

O IAM é um componente da infraestrutura de segurança da AWS. Com o IAM, você pode gerenciar centralmente grupos, usuários, funções de serviço e credenciais de segurança, como senhas, chaves de acesso e políticas de permissões que controlam quais serviços e recursos da AWS os usuários podem acessar. A política do IAM permite definir o conjunto de permissões. Essa política pode ser anexada a uma função, a um usuário ou a um serviço para definir sua permissão. Você também pode usar o IAM para criar funções amplamente usadas na estratégia de DevOps desejada. Em alguns casos, pode fazer todo o sentido usar programaticamente AssumeRole em vez de obter as permissões diretamente. Quando um serviço ou usuário assume funções, ele recebe credenciais temporárias para acessar um serviço ao qual normalmente não tem acesso.

Conclusão

Para que a jornada para a nuvem seja tranquila, eficiente e eficaz, as empresas de tecnologia devem adotar os princípios e práticas de DevOps. Esses princípios estão incorporados na plataforma da AWS. Na verdade, eles formam a base de vários serviços da AWS, especialmente os inclusos nas ofertas de implantação e monitoramento.

Comece definindo sua infraestrutura como código usando o serviço AWS CloudFormation ou AWS Cloud Development Kit (AWS CDK). O próximo passo é definir a maneira pela qual suas aplicações usarão a implantação contínua com a ajuda de serviços como AWS CodeBuild, AWS CodeDeploy, AWS CodePipeline e AWS CodeCommit. No nível da aplicação, use contêineres como AWS Elastic Beanstalk, Amazon Elastic Container Service (Amazon ECS) ou o Amazon Elastic Kubernetes Service (Amazon EKS) e o AWS OpsWorks para simplificar a configuração de arquiteturas comuns. O uso desses serviços também facilita a inclusão de outros serviços importantes, como Auto Scaling e Elastic Load Balancing. Por fim, use a estratégia de monitoramento de DevOps, como o Amazon CloudWatch, e práticas de segurança sólidas, como o AWS IAM.

Com a AWS como parceira, seus princípios de DevOps trazem agilidade para sua empresa e organização de TI, além de acelerar sua jornada para a nuvem.

Revisões do documento

Para ser notificado sobre atualizações deste whitepaper, inscreva-se no RSS feed.

update-history-change	update-history-description	update-history-date
Restauração da seção Contribuidores que estava ausente	Restauração da seção Contribuidores que estava ausente e pequenas alteraçõe s de texto	21 de novembro de 2020
Atualização de seções para incluir novos serviços	Atualização de seções para incluir novos serviços	16 de outubro de 2020
Publicação inicial	Primeira publicação do whitepaper	1º de dezembro de 2014

Colaboradores

Dentre os colaboradores deste documento estão:

- Muhammad Mansoor, arquiteto de soluções
- Ajit Zadgaonkar, líder mundial em tecnologia, modernização
- Juan Lamadrid, arquiteto de soluções
- Darren Ball, arquiteto de soluções
- Rajeswari Malladi, arquiteto de soluções
- Pallavi Nargund, arquiteto de soluções
- Bert Zahniser, arquiteto de soluções
- Abdullahi Olaoye, arquiteto de soluções de nuvem
- Mohamed Kiswani, gerente de desenvolvimento de software
- Tara McCann, gerente de arquitetura de soluções

Avisos

Os clientes são responsáveis por fazer sua própria avaliação independente das informações neste documento. Este documento é: (a) fornecido apenas para fins informativos, (b) representa as ofertas e práticas de produtos atuais da AWS, que estão sujeitas a alterações sem aviso prévio e (c) não cria nenhum compromisso ou garantia da AWS e suas afiliadas, fornecedores ou licenciadores. Os produtos ou serviços da AWS são fornecidos no “estado em que se encontram”, sem garantias, declarações ou condições de qualquer tipo, explícitas ou implícitas. As responsabilidades e obrigações da AWS com seus clientes são regidas por contratos da AWS, e este documento não modifica nem faz parte de nenhum contrato entre a AWS e seus clientes.