WHITE PAPER

Como considerar a abordagem nativa de nuvem

Pequenos artigos sobre a definição e o desenvolvimento dos recursos nativos de nuvem

vmware®

Sumário

Definições 3

Na prática 3

DevOps 4

Contêineres e clusters 5

Microsserviços 6

Segurança 7

Segurança da imagem de contêiner 7

Microsserviço e rede de segurança..... 8

"A abordagem nativa de nuvem está estruturando as equipes, a cultura e a tecnologia para utilizar a automação e as arquiteturas a fim de gerenciar a complexidade e promover a agilidade."

JOE BEDA
ENGENHEIRO-CHEFE
VMWARE

Definições

Não existe uma definição absoluta para o termo "nativo da nuvem". Na verdade, existem outros termos e ideologias que se sobrepõem. Originalmente, a abordagem nativa de nuvem está estruturando as equipes, a cultura e a tecnologia para utilizar a automação e as arquiteturas a fim de gerenciar a complexidade e promover a agilidade. Dessa forma, tanto o lado da equação referente às pessoas quanto o lado da infraestrutura são dimensionados.

Um lembrete importante: você não precisa estar na nuvem para ser considerado nativo da nuvem. Essas técnicas podem ser aplicadas de forma incremental, conforme apropriado, e ajudarão a suavizar qualquer transição para a nuvem.

O valor real da abordagem nativa de nuvem ultrapassa o leque das tecnologias que estão estreitamente associadas a ela. Para compreender totalmente o rumo que nosso setor está tomando, precisamos analisar em que área e como podemos ajudar empresas, equipes e pessoas a ter mais sucesso.

Nesse ponto, essas técnicas foram comprovadas em empresas inovadoras e centradas em tecnologia que dedicaram muitos recursos a essa iniciativa. Pense no Google, na Netflix ou no Facebook. Empresas menores e mais flexíveis também estão percebendo valor aqui. No entanto, não há muitos casos da aplicação dessa filosofia fora dos pioneiros das tecnologias. Ainda estamos no início da jornada se considerarmos o amplo mundo da TI.

Agora que algumas das primeiras experiências já foram comprovadas e compartilhadas, os seguintes temas estão surgindo:

- Equipes mais eficientes e satisfeitas: as ferramentas nativas de nuvem permitem que grandes problemas sejam fragmentados em partes menores para que as equipes possam ser mais focadas e ágeis.
- Menos esforço: este resultado é obtido por meio da automatização de grande parte do trabalho manual que gera os problemas e o tempo de inatividade das operações. Isso assume a forma de uma infraestrutura de autocorreção e autogerenciamento. A expectativa é que os sistemas assumam mais tarefas.
- Infraestrutura e aplicativos mais confiáveis: o desenvolvimento da automação para lidar com a rotatividade esperada geralmente resulta em melhores modos de falha para eventos e falhas inesperados. Por exemplo, se um único comando ou clique de botão implantar um aplicativo para desenvolvimento, teste ou produção, poderá ser muito mais fácil automatizar a implantação em um cenário de recuperação de desastres (seja automática ou manualmente).
- Auditável, visível e depurável: aplicativos complexos podem ser obscuros. As ferramentas usadas para aplicativos nativos de nuvem geralmente fornecem, por necessidade, muito mais informações sobre o que está acontecendo em um aplicativo.
- Segurança profunda: atualmente, muitos sistemas de TI têm um shell externo rígido e um centro flexível. Por padrão, os sistemas modernos devem ser seguros e menos confiáveis. A abordagem nativa de nuvem permite que os desenvolvedores de aplicativos desempenhem um papel ativo na criação de aplicativos que possam ser protegidos.
- Uso mais eficiente de recursos: maneiras automatizadas de implantar e gerenciar aplicativos e serviços, semelhantes ao modo de funcionamento da nuvem, oferecem oportunidades para a aplicação da automação algorítmica. Por exemplo, um orquestrador de cluster pode automatizar o posicionamento das cargas de trabalho nas máquinas, em vez de ter uma equipe de operações gerenciando o posicionamento em uma planilha.

Na prática

Assim como qualquer área com inovação ativa, há bastante rotatividade no mundo nativo da nuvem. Nem sempre fica claro qual é a melhor forma de aplicar as ideias apresentadas na parte anterior. Além disso, qualquer projeto relevante será muito importante e muito grande para ser submetido a uma reescrita do zero. Em vez disso, recomendo que você experimente essas novas estruturas em projetos mais recentes ou em novas partes de um projeto existente. À medida que as partes mais antigas do sistema forem aprimoradas, reserve um tempo para aplicar novas técnicas e aprendizados, conforme apropriado. Procure maneiras de descobrir novos recursos ou sistemas como microsserviços.

"O que define um SRE é o que acontece às 10h da manhã seguinte."

JOE BEDA
ENGENHEIRO-CHEFE
VMWARE

Não existem regras definidas. Cada organização é diferente, e as práticas de desenvolvimento de software devem ser dimensionadas de acordo com a equipe e o projeto em questão. O mapa não é o território. Alguns projetos são passíveis de experimentação, enquanto outros são tão importantes que devem ser abordados com muito mais cuidado. Além disso, surgem situações no meio do processo em que as técnicas comprovadas precisam ser formalizadas e testadas em escala antes de serem aplicadas a sistemas críticos.

A abordagem nativa de nuvem é definida pela presença de melhores ferramentas e sistemas. Sem essas ferramentas, cada novo serviço em produção terá um alto custo operacional. É algo separado que precisa ser monitorado, rastreado, aprovisionado e assim por diante. Essa sobrecarga é uma das principais razões pelas quais o dimensionamento de microsserviços deve ser feito de maneira apropriada. Os benefícios da velocidade do desenvolvimento devem ser pesados com base nos custos da execução de mais serviços na produção. Da mesma forma, a introdução de novas tecnologias e linguagens, embora empolgante, acarreta custos e riscos que devem ser avaliados com cuidado.

A automação é a chave para reduzir as despesas operacionais associadas à criação e à execução de novos serviços. Sistemas como Kubernetes, contêineres, integração e fornecimento contínuos (CI/CD, pela sigla em inglês) e monitoramento têm um objetivo em comum: tornar as equipes de operações e desenvolvimento de aplicativos mais eficientes para que possam tomar medidas com mais rapidez e criar produtos mais confiáveis.

A última geração de ferramentas e sistemas estão mais preparadas para cumprir a promessa da abordagem nativa de nuvem do que as ferramentas tradicionais e mais antigas de gerenciamento de configurações, pois as novas ferramentas ajudam a fragmentar o problema, de modo que ele possa ser facilmente distribuído entre as equipes. As ferramentas mais recentes geralmente capacitam as equipes de desenvolvimento e operações a manterem a propriedade e serem mais produtivas por meio da TI de autoatendimento.

DevOps

Provavelmente, é mais útil considerar o DevOps como uma mudança cultural na qual os desenvolvedores devem se concentrar em como os aplicativos são executados em um ambiente de produção. Além disso, os membros da equipe de operações têm consciência e capacidade para entender como o aplicativo funciona e, assim, podem trabalhar ativamente para tornar o aplicativo mais confiável. Desenvolver uma compreensão e empatia entre essas equipes é fundamental.

Mas isso pode ir mais longe. Se reexaminarmos o modo como os aplicativos são criados e como a equipe de operações é estruturada, poderemos melhorar e aprofundar esse relacionamento. O Google não emprega equipes de operações tradicionais. Em vez disso, ele define um novo tipo de engenheiro chamado engenheiro de confiabilidade de site (SRE, pela sigla em inglês). São engenheiros altamente treinados (remunerados no mesmo nível que outros engenheiros) que não apenas carregam um pager, mas devem e estão capacitados a desempenhar um papel essencial para promover a maior confiabilidade dos aplicativos por meio da automação.

Quando o pager dispara às 2h, qualquer um que o atende faz exatamente a mesma coisa: tenta descobrir logo o que está acontecendo para que possa voltar para a cama. O que define um SRE é o que acontece às 10h da manhã seguinte. O pessoal da equipe de operações apenas reclama ou trabalha com a equipe de desenvolvimento para garantir que isso nunca mais aconteça novamente? O SRE e as equipes de desenvolvimento têm incentivos alinhados para tornar o produto o mais confiável possível. Isso, combinado com a emissão de laudos pós-incidentes isentos de culpados (blameless postmortems), pode levar a projetos íntegros que não cobram dívidas técnicas.

Os SREs estão entre as pessoas mais valorizadas pelo Google. Na verdade, os produtos geralmente são lançados sem a participação dos SREs, na expectativa de que a equipe de desenvolvimento executará os produtos na produção. Para incorporar os SREs, geralmente é necessário que a equipe de desenvolvimento comprove para a equipe de SREs que o produto está pronto. Espera-se que a equipe de desenvolvimento tenha feito todo o trabalho de campo, incluindo configuração de monitoramento e alertas, manual de alertas e processos de liberação. A equipe de desenvolvimento precisa mostrar que as páginas foram reduzidas à quantidade mínima e que a maioria dos problemas foi automatizada.

Como o papel da equipe de operações se torna muito mais participativo e específico de aplicativo, não faz muito sentido que uma única equipe possua toda a pilha de operações. Essa postura converge para a ideia de especialização em operações. De certa forma, essa é uma abordagem antiDevOps. Vamos analisar essa situação de baixo para cima:

- Operações de hardware: isso já pode ser claramente separado. Na verdade, é fácil enxergar a infraestrutura em nuvem como serviço (IaaS) como operações de hardware como serviço.
- Operações do SO: alguém precisa garantir que as máquinas sejam inicializadas e que haja um kernel satisfatório. A ruptura com o gerenciamento de dependências de aplicativos reflete a tendência das distribuições mínimas de SO focada na hospedagem de contêineres (como Project Photon OS™, CoreOS, Red Hat Project Atomic, Ubuntu Snappy, RancherOS e Container-Optimized OS do Google).
- Operações de cluster: no mundo em contêineres, um cluster de processamento se torna uma plataforma de infraestrutura lógica. O sistema de cluster (Kubernetes) fornece um conjunto de primitivas que permitem que várias tarefas de operações tradicionais sejam realizadas por autoatendimento.
- Operações de aplicativos: agora, cada aplicativo pode ter uma equipe de aplicativos dedicada, conforme apropriado. Como mencionado anteriormente, a equipe de desenvolvimento pode e deve desempenhar esse papel quando necessário. Espera-se que essa equipe de operações analise detalhadamente o aplicativo, já que eles não precisam ser especialistas nas outras camadas. Por exemplo, no Google, a equipe de SREs de front-end do AdWords conversará muito mais com a equipe de desenvolvimento de front-end do AdWords do que com a equipe de SREs do cluster (Borg SRE). Esse alinhamento de incentivos pode levar a melhores resultados.

Provavelmente, haverá espaço para outras equipes de SREs especializadas, dependendo das necessidades da organização. Por exemplo, os serviços de armazenamento podem ser divididos como um serviço separado com SREs dedicados. Ou pode haver uma equipe responsável por criar e validar a imagem de contêiner base que todas as equipes devem usar por uma questão de política.

Contêineres e clusters

A expectativa em torno dos contêineres é muito grande. É recomendável tentar descobrir por que os contêineres empolgam tantas pessoas. Na minha opinião, existem três motivos para isso:

1. Pacotes e portabilidade
2. Eficiência
3. Segurança

Vamos analisar cada um desses itens.

Primeiramente, os contêineres fornecem um mecanismo de empacotamento. Isso permite que o desenvolvimento de um sistema seja separado da sua implantação. Além disso, os elementos/imagens criados são muito mais portáveis entre ambientes (desenvolvimento, teste, validação, produção) do que as abordagens mais tradicionais, como imagens de máquina virtual (VM, pela sigla em inglês). Por fim, as implantações se tornam mais atômicas. Os sistemas tradicionais de gerenciamento de configurações (Puppet, Chef, Salt, Ansible) podem facilmente deixar os sistemas configurados pela metade, o que será difícil de depurar. Também é fácil ter uma diferença de versão não intencional entre as máquinas sem perceber isso.

Em segundo lugar, os contêineres podem ser mais leves que os sistemas completos, o que acarretará o aumento da utilização de recursos. Esse foi o principal impulsionador quando o Google introduziu os cgroups: uma das principais tecnologias de kernel subjacentes aos contêineres. Ao compartilhar um kernel e permitir um overcommit muito mais fluido, os contêineres podem facilitar a maximização do uso dos recursos de processamento. Ao longo do tempo, a expectativa é que vejamos formas muito mais sofisticadas de equilibrar as necessidades dos contêineres que coabitam um único host sem problemas com vizinhos barulhentos.

Por fim, muitos usuários consideram os contêineres como uma fronteira de segurança. Embora os contêineres possam ser mais seguros do que os processos simples do Unix, é necessário ter cuidado antes de considerá-los como uma fronteira de segurança confiável. As garantias de segurança fornecidas pelos namespaces do Linux podem ser apropriadas para a multilocação flexível (em que as cargas de trabalho são semiconfiáveis), mas não para a multilocação rígida (em que as cargas de trabalho são ativamente antagônicas).

"Os clusters ajudam a eliminar a complexidade das operações."

JOE BEDA
ENGENHEIRO-CHEFE
VMWARE

"A arquitetura de aplicativos deve crescer de maneira prática e orgânica."

JOE BEDA
ENGENHEIRO-CHEFE
VMWARE

Há vários trimestres, tem sido realizados trabalhos contínuos para eliminar as barreiras entre contêineres e VMs. As primeiras pesquisas em sistemas como o unikernels são interessantes, mas levará anos para ser disponibilizada para ampla produção.

Embora os contêineres sejam uma maneira fácil de atingir as metas mencionadas anteriormente, eles não são absolutamente necessários. A Netflix, por exemplo, tem executado uma pilha muito moderna ao empacotar e usar imagens de VM de maneira semelhante à de outras empresas que usam contêineres.

Embora grande parte do impulso original em torno dos contêineres tenha se concentrado no gerenciamento do software em um único nó de maneira mais confiável e previsível, a próxima etapa dessa evolução baseia-se nos clusters (também conhecidos como orquestradores). Adotar vários nós e vinculá-los a sistemas automatizados cria um novo conjunto de autoatendimento de infraestrutura lógica para as equipes de desenvolvimento e de operações.

Com um cluster de contêiner, os computadores ficam com a responsabilidade de descobrir qual carga de trabalho deve ser executada em qual máquina. Os clusters também fazem correções silenciosamente quando o hardware apresenta falha no meio da noite, em vez de acionar o pager de alguém.

A primeira providência dos clusters é habilitar a especialização em operações (conforme descrito anteriormente), o que permite que as operações de aplicativos desenvolvam-se como uma disciplina separada. Por ter uma interface de cluster bem definida, as equipes de aplicativos podem se concentrar na solução dos problemas que atingem diretamente o aplicativo em si.

O segundo benefício dos clusters é a possibilidade de iniciar e gerenciar mais serviços. Isso permite novas arquiteturas (por meio dos microsserviços descritos na próxima seção desta série) que podem promover a agilidade das equipes de desenvolvimento.

Microsserviços

Os microsserviços são um novo nome para um conceito que já existe há muito tempo. Basicamente, é uma maneira de fragmentar um grande aplicativo em partes menores, para que elas possam ser desenvolvidas e gerenciadas de forma independente. Vamos analisar alguns aspectos importantes aqui:

- Interfaces fortes e claras: a estreita associação entre os serviços deve ser evitada. Interfaces documentadas e com versões ajudam a solidificar esse acordo e manter um certo grau de liberdade, tanto para os consumidores quanto para os produtores desses serviços.
- Implantado e gerenciado de modo independente: um único microsserviço poderá ser atualizado sem sincronização com todos os outros serviços. Poder reverter facilmente a versão de um microsserviço também é algo desejável. Isso significa que os binários implantados devem ser compatíveis com versões anteriores e futuras, tanto em termos de API quanto de qualquer esquema de dados. Isso favorecerá o teste dos mecanismos de cooperação e comunicação entre as equipes apropriadas de operações e desenvolvimento.
- Resiliência integrada: os microsserviços devem ser criados e testados para oferecerem resiliência independente. O código que consome um serviço terá dificuldade para continuar trabalhando e oferecer um resultado razoável caso o serviço consumido seja desativado ou se comporte incorretamente. Da mesma forma, qualquer serviço oferecido deve ter algumas defesas contra carga imprevista e entrada inválida.

O dimensionamento de microsserviços pode ser algo complicado. Eu aconselharia a evitar serviços muito pequenos e, em vez disso, focar na divisão de serviços que ultrapassam as fronteiras naturais (idiomas, filas assíncronas, requisitos de dimensionamento) e manter tamanhos de equipe razoáveis (ou seja, equipes de duas pizzas).

Em vez de começar com 20 serviços, comece com dois a três serviços e divida-os à medida que a complexidade nessa área aumentar. Muitas vezes, a arquitetura de um aplicativo só é bem compreendida depois que o aplicativo já está na fase de desenvolvimento. Isso também confirma a tese de que os aplicativos raramente são concluídos, mas são sempre um trabalho em andamento.

"Quando uma imagem vulnerável é encontrada, os problemas deixam de ser técnicos e se tornam processuais ou de fluxo de trabalho."

JOE BEDA
ENGENHEIRO-CHEFE
VMWARE

MICROSSEGMENTAÇÃO PARA CONTÊINERES

A microssegmentação usa a virtualização de redes para dividir um data center e suas cargas de trabalho em segmentos lógicos, cada um deles contendo uma única carga de trabalho. Em seguida, você pode aplicar controles de segurança a cada segmento, restringindo a capacidade de um invasor de se mover para outro segmento ou carga de trabalho.

Os microsserviços são um novo conceito? Não exatamente. Eles são, na verdade, outro tipo de componentização de software. Sempre dividimos o código em bibliotecas. Trata-se apenas de converter o conceito de tempo de construção do vinculador em conceito de tempo de execução. Isso é também muito semelhante ao impulso da SOA de vários anos atrás, mas sem todo aquele volume de XML. Visto por outro ângulo, o banco de dados quase sempre foi um microsserviço, pois é frequentemente implantado de uma maneira que atende aos pontos destacados anteriormente.

As restrições podem resultar em maior produtividade. Embora seja tentador permitir que cada equipe escolha um idioma ou uma estrutura diferente para cada microsserviço, considere a padronização de algumas linguagens e estruturas. Essa medida aumentará a transferência de conhecimento e a mobilidade na organização. No entanto, esteja aberto para fazer exceções à política quando necessário. Essa é uma das principais vantagens deste mundo em relação a uma plataforma como serviço (PaaS, pela sigla em inglês) mais estruturada e integrada verticalmente. Em outras palavras, as restrições devem ser uma questão de política, não de capacidade.

Embora a maioria considere os microsserviços como uma técnica de implementação para um aplicativo grande, existem outros tipos de serviços que compõem o espectro de serviços:

1. Serviço como detalhes de implementação: conforme descrito acima, isso é útil para dividir uma grande equipe de aplicativos em equipes menores, que vão desde o desenvolvimento até as operações.
2. Elemento compartilhado, instância privada: neste cenário o processo, de desenvolvimento é compartilhado entre várias instâncias do serviço. Talvez haja uma equipe de desenvolvedores e várias equipes de operações, ou uma equipe de operações unificada que trabalhe em todas as instâncias dedicadas. Muitos bancos de dados se enquadram nessa categoria, na qual muitas equipes estão executando instâncias privadas de um único binário do MySQL.
3. Instância compartilhada: neste caso, uma única equipe fornece um serviço compartilhado a vários aplicativos e equipes em uma organização. O serviço pode particionar dados e ações por usuário (multilocatário) ou fornecer um serviço único e simples que seja amplamente utilizado (atendendo à UI de HTML em uma barra de marca comum, atendendo a modelos de aprendizado de máquina etc.).
4. Serviço Big-S: a maioria das empresas não produzirá um serviço como esse, mas poderá consumi-lo. Esse é o serviço típico de multilocatário rígido criado para atender a um grande número de clientes diversos. Esse tipo de serviço exige um nível de responsabilidade e reforço de segurança que geralmente não é necessário em uma empresa. Algo como o SendGrid ou o Twilio se enquadraria nessa categoria.

À medida que os serviços deixam de ser um detalhe de implementação para se tornar uma infraestrutura comum oferecida em uma empresa, a rede de serviços migra de um conceito por aplicativo para algo que pode abranger toda a empresa. Existe a oportunidade de permitir esses tipos de dependências, mas isso também representa um perigo.

Segurança

A segurança ainda é uma grande questão no mundo nativo da nuvem. As antigas técnicas não se aplicam nitidamente e, portanto, em princípio, a abordagem nativa de nuvem pode parecer um retrocesso. Mas esse admirável mundo novo também cria oportunidades.

Segurança da imagem de contêiner

Algumas ferramentas ajudam os usuários a auditar suas imagens de contêiner para garantir que sejam totalmente corrigidas. Não tenho uma opinião formada sobre as várias opções nessa área.

Este é o problema real: o que você faz quando se depara com uma imagem de contêiner vulnerável? Essa é uma área para a qual o mercado não forneceu um grande conjunto de soluções. Você precisará identificar quais grupos em sua organização são afetados, em que local da árvore de imagens de contêiner o problema será corrigido e qual a melhor forma de testar e distribuir uma nova versão corrigida.

SAIBA MAIS SOBRE A TECNOLOGIA NATIVA DE NUVEM DA VMWARE

Para saber mais sobre como a VMware pode ajudá-lo a criar, executar e gerenciar aplicativos nativos de nuvem, consulte

https://cloud.vmware.com/

BIOGRAFIA DO AUTOR

Joe Beda é engenheiro-chefe da VMware. Ele foi cofundador e CTO da Heptio. Anteriormente, no Google, Joe participou da criação do Google Compute Engine e registrou o primeiro commit do projeto Kubernetes. Ele é defensor convicto do software de código aberto e colaborador frequente da comunidade.

O CI/CD é peça fundamental do quebra-cabeça porque permitirá processos de liberação automatizados e rápidos para as novas imagens. Além disso, a integração com sistemas de orquestração permitirá que você identifique quais usuários estão usando quais imagens vulneráveis. Isso também permitirá que você verifique se uma nova versão corrigida está mesmo sendo executada na produção. Por fim, a política no sistema de implantação pode ajudar a impedir que novos contêineres sejam iniciados com uma imagem inválida conhecida (no mundo do Kubernetes, essa política é chamada de admissão).

Microsserviço e rede de segurança

No entanto, mesmo que todo o conteúdo executado no cluster seja corrigido, isso não garante que todas as atividades da rede sejam confiáveis.

As ferramentas de segurança tradicionais com base em rede não funcionam corretamente em um mundo de contêineres de curta duração e agendados dinamicamente. Os contêineres de curta duração podem não permanecer tempo suficiente para serem verificados pelas ferramentas de verificação tradicionais. E, quando um relatório for gerado, o contêiner em questão pode ter desaparecido.

Com os orquestradores dinâmicos, os endereços IP não têm significado de longo prazo e podem ser reutilizados automaticamente. A solução é integrar ferramentas de análise de rede ao orquestrador para que nomes lógicos (e outros metadados) possam ser usados além dos endereços IP brutos. Isso provavelmente tornará os alertas mais fáceis de serem acionados.

Muitas das tecnologias de sistema de redes aproveitam o encapsulamento para implementar um endereço IP por contêiner. Isso pode criar problemas para ferramentas de rastreamento e inspeção de rede. Elas terão que ser adaptadas se esses sistemas de redes forem implantados na produção. Felizmente, grande parte disso foi padronizado em VXLANs e VLANS, ou não há encapsulamento; portanto, o suporte pode ser aproveitado em muitos desses sistemas.

No entanto, na minha opinião, os maiores problemas estão em torno dos microsserviços. Quando há muitos serviços em execução na produção, é necessário garantir que apenas clientes autorizados estejam chamando qualquer serviço específico. Além disso, com a reutilização de endereços IP, os clientes precisam saber que estão se comunicando com o serviço correto. Existem duas maneiras (não mutuamente exclusivas) de abordar esse problema.

Na primeira abordagem, os sistemas de redes mais flexíveis têm a oportunidade de implementar regras de firewall no nível do host (fora de qualquer contêiner) para permitir que políticas de acesso refinadas de contêineres chamem outros contêineres. Eu chamei essa abordagem de microssegmentação de rede. O desafio aqui é configurar essa política perante o agendamento dinâmico. Embora ainda seja cedo, várias empresas estão trabalhando para tornar isso mais fácil por meio do suporte na rede, da coordenação com o orquestrador e das definições de aplicativos de nível superior. Uma ressalva importante: a microssegmentação se torna menos eficaz quanto maior é o alcance do uso de qualquer serviço específico. Se um serviço tiver centenas de chamadores, os modelos simples de “acesso implica autorização” não serão mais eficazes.

A segunda abordagem é que os aplicativos desempenhem um papel mais importante na implementação da autenticação e da criptografia no data center. Isso funciona na medida em que os serviços assumem muitos clientes e se tornam multilocatários flexíveis em uma grande organização. Isso requer um sistema de identidade para serviços de produção. Paralelamente, iniciei um projeto chamado SPIFFE (Secure Production Identity Framework For Everyone). Essas ideias são comprovadas em empresas como o Google.

A segurança é um tópico complexo e, certamente, há ameaças e considerações que não foram mencionadas neste documento. Portanto, essa discussão não pode parar.

Existe um ponto de partida para começarmos a considerar a abordagem nativa de nuvem. Se você estiver disposto a continuar a discussão, entre em contato conosco.

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com
Rua Surubim, 504 4 andar CEP 04571-050 Cidade Monções – São Paulo – SP Tel: (+55) 11 5509-7200 www.vmware.com/br
Copyright © 2019 VMware, Inc. Todos os direitos reservados. Este produto é protegido pelas leis norte-americanas e internacionais de direitos autorais e propriedade intelectual. Os produtos da VMware estão cobertos por uma ou mais patentes listadas no site http://www.vmware.com/go/patents. VMware é uma marca registrada ou comercial da VMware, Inc. e de suas filiais nos Estados Unidos e/ou em outras jurisdições. Todas as outras marcas e nomes aqui mencionados podem ser marcas comerciais de suas respectivas empresas.
Item No.: 217250aq-vmw-wp-cloud-native-A4 4/19