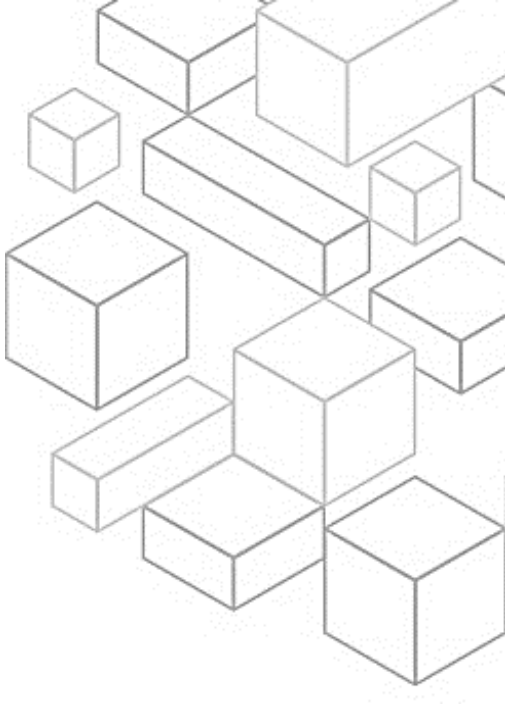


Módulo 6 - Análise detalhada do pilar de confiabilidade

1.1 Boas-vindas!

Boas-vindas ao módulo seis do AWS Well-Architected: Análise detalhada do pilar de confiabilidade.

1.2 Objetivos de aprendizado

Neste módulo, você aprenderá sobre o pilar de confiabilidade do AWS Well-Architected Framework Você também aprenderá os princípios de design e as práticas recomendadas do pilar de confiabilidade.

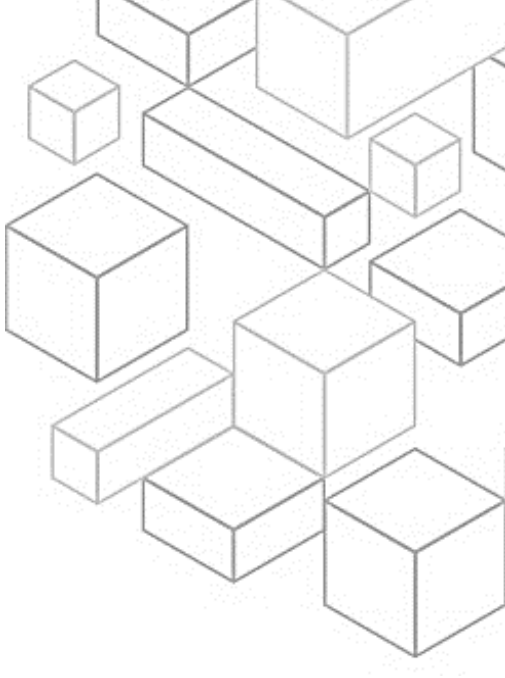
1.3 Visão geral do pilar de confiabilidade

Para começar, você terá uma visão geral do pilar de confiabilidade.

1.4 Pilares do AWS Well-Architected

Atualmente, há seis pilares do Well-Architected Framework: excelência operacional, segurança, confiabilidade, eficiência de desempenho, otimização de custos e sustentabilidade. Esses pilares são os fundamentos da arquitetura de suas soluções de tecnologia na nuvem.

Este módulo se concentrará no pilar de confiabilidade.



1.5 O que é o pilar de confiabilidade?

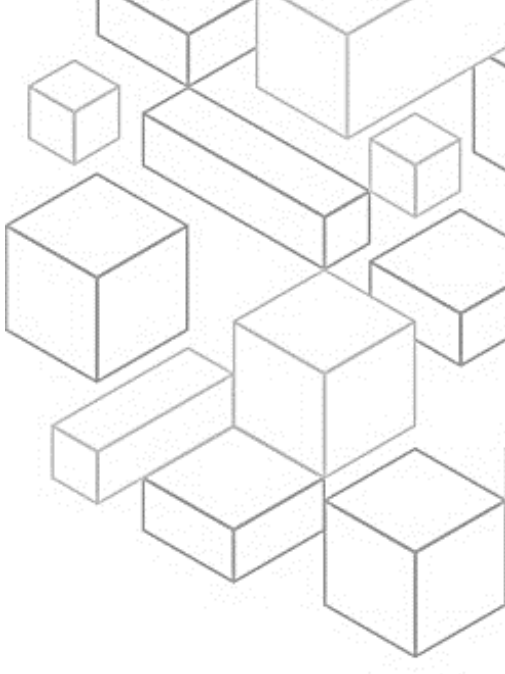
O que é o pilar de confiabilidade? Como devemos avaliar isso? Confiabilidade é a capacidade de uma carga de trabalho de executar sua função pretendida de forma correta e consistente durante um período de tempo esperado. O pilar de confiabilidade se concentra na capacidade de uma carga de trabalho de executar sua função pretendida corretamente e de forma consistente quando é esperado. Isso inclui a capacidade de operar e testar a carga de trabalho por meio de seu ciclo de vida total.

1.6 Princípios de Design de Confiabilidade

Na nuvem, vários princípios podem ajudar a aumentar a confiabilidade. Agora você vai saber mais sobre esses princípios de design.

1.7 Princípios de design de confiabilidade

Primeiro, analise os princípios de design de confiabilidade antes de se aprofundar nas práticas recomendadas. Os princípios de design ajudam a moldar um modelo mental para o pilar de confiabilidade, especialmente se você estiver vindo de um ambiente tradicional on-premises. Um dos princípios é recuperar-se automaticamente de falhas. Para isso, monitore uma carga de trabalho para obter os principais indicadores de desempenho (KPIs) e, em seguida, inicie uma automação para executar um trabalho específico quando o limite for violado. Esses KPIs devem ser uma medida do valor comercial, não dos aspectos técnicos da operação do serviço. Você pode ter notificação e rastreamento automáticos de falhas e processos de recuperação automatizados que contornam ou reparam

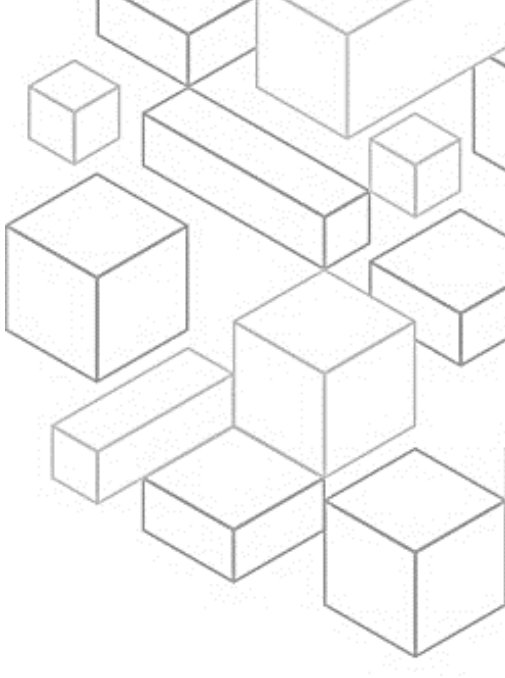


a falha. Com automação mais sofisticada, é possível prever e corrigir falhas antes que elas ocorram.

O próximo princípio é testar os procedimentos de recuperação. Em um ambiente on-premises, os testes geralmente são realizados para provar que a carga de trabalho funciona em um cenário específico. Normalmente, o teste não é usado para validar estratégias de recuperação. Na nuvem, você pode testar como sua carga de trabalho falha e validar seus procedimentos de recuperação. Você pode usar a automação para simular falhas diferentes ou recriar cenários que antes geraram falhas. Essa abordagem expõe caminhos de falha que você pode testar e corrigir antes que ocorra um cenário de falha real, reduzindo assim o risco.

Dimensionar horizontalmente para aumentar a disponibilidade agregada da carga de trabalho. Substitua um recurso grande por vários recursos pequenos para reduzir o impacto de uma única falha na carga de trabalho geral. Distribua solicitações entre vários recursos menores para garantir que eles não compartilhem um ponto comum de falha.

A seguir, pare de adivinhar a capacidade. Uma causa comum de falha em cargas de trabalho on-premises é a saturação de recursos, quando as demandas colocadas em uma carga de trabalho excedem a capacidade dessa carga de trabalho. Um exemplo são os ataques de negação de serviço. Na nuvem, você pode monitorar a demanda e a utilização da carga de trabalho e automatizar a adição ou remoção de recursos para manter o nível ideal para atender à demanda sem excesso ou falta de provisionamento. Ainda há limites, mas algumas cotas podem ser controladas e outras podem ser gerenciadas.



Outro princípio é como gerenciar as alterações por meio da automação. As alterações em sua infraestrutura devem ser feitas por meio da automação. As alterações que precisam ser gerenciadas incluem alterações na automação, que podem ser rastreadas e analisadas.

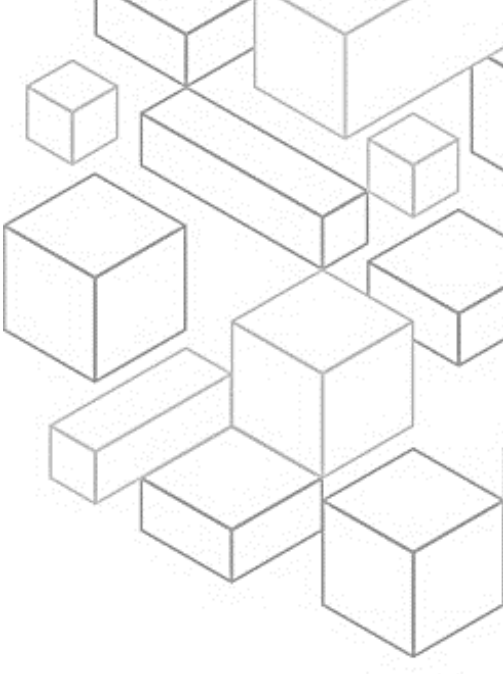
1.8 Práticas recomendadas de confiabilidade

Na nuvem, as práticas recomendadas podem ajudar a aumentar a confiabilidade. Nesta seção, você saberá mais sobre essas práticas recomendadas.

1.9 Áreas de práticas recomendadas de sustentabilidade

As práticas recomendadas no pilar de confiabilidade estão organizadas em quatro áreas, sendo a primeira a de fundamentos. O escopo dos requisitos fundamentais vai além de uma única carga de trabalho ou projeto. Antes de projetar qualquer sistema, é necessário implementar requisitos fundamentais que influenciam a confiabilidade. Por exemplo, você deve ter largura de banda de rede suficiente para o seu data center. Em um ambiente on-premises, esses requisitos podem causar tempos de execução longos devido a dependências e, portanto, devem ser incorporados durante o planejamento inicial. No entanto, com a AWS, a maioria desses requisitos fundamentais já está incorporada ou pode ser abordada conforme necessário. A nuvem foi projetada para ser quase ilimitada, portanto, é responsabilidade da AWS satisfazer o requisito de capacidade suficiente computacional e de rede. Isso deixa você livre para alterar o tamanho e as alocações de recursos sob demanda.

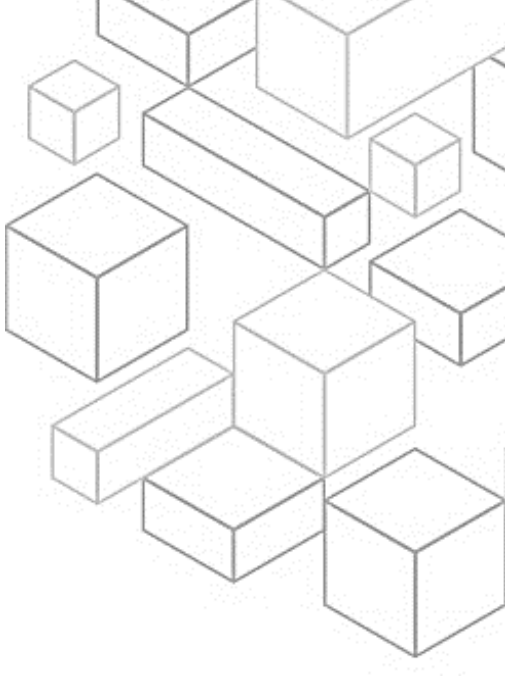
A próxima prática recomendada é a arquitetura da carga de trabalho. Uma carga



de trabalho confiável começa com decisões iniciais de design para o software e a infraestrutura. Suas escolhas de arquitetura afetarão o comportamento da carga de trabalho em todos os seis pilares do AWS Well-Architected. Para obter confiabilidade, há padrões específicos que você deve seguir.

Para a prática recomendada de gerenciamento de alterações, as alterações em sua carga de trabalho ou em seu ambiente devem ser previstas e acomodadas para uma operação confiável da carga de trabalho. As alterações incluem aquelas impostas à sua carga de trabalho, como picos de demanda. Elas também incluem alterações internas, como implantações de recursos e patches de segurança.

Por fim, no que se refere ao gerenciamento de falhas, as falhas de baixo nível dos componentes de hardware são algo com que se lida todos os dias em um data center on-premises. Na nuvem, entretanto, você deve estar protegido contra a maioria desses tipos de falhas. Por exemplo, os volumes do Amazon Elastic Block Store, ou Amazon EBS, são colocados em uma Zona de Disponibilidade específica, onde são replicados automaticamente para proteger você contra a falha de um único componente. Todos os volumes do EBS são projetados para uma disponibilidade de cinco noves. Os objetos do Amazon Simple Storage Service, ou Amazon S3, são armazenados em um mínimo de três Zonas de Disponibilidade, proporcionando onze noves de durabilidade para os objetos em um determinado ano. Independentemente do seu provedor de nuvem, há a possibilidade de falhas afetarem sua carga de trabalho. Portanto, você deve tomar medidas para implementar a resiliência se precisar que sua carga de trabalho seja confiável. Um pré-requisito para aplicar as práticas recomendadas discutidas aqui é garantir que as pessoas que estão projetando,



implementando e operando suas cargas de trabalho sejam treinadas para os objetivos comerciais e as metas de confiabilidade para alcançá-los.

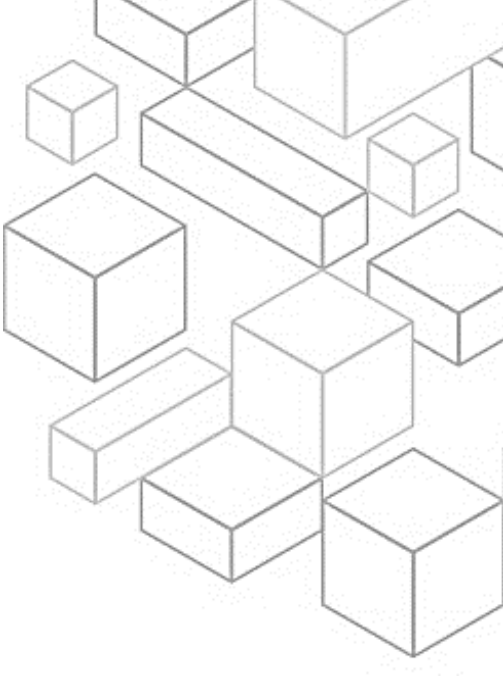
1.10 Fundamentos

Na área de práticas recomendadas de fundamentos, o escopo pode ir além de uma única carga de trabalho ou projeto. Ela precisa ser compreendida e implementada antes de arquitetar qualquer sistema. Caso contrário, você poderá enfrentar longos prazos de entrega e bloqueios ao longo do caminho. Cuide disso com antecedência para que você possa ficar livre para alterar o tamanho e a alocação dos recursos sob demanda. Nesta seção, você aprenderá a gerenciar cotas ou restrições de serviço e a planejar a topologia da rede.

1.11 Gerenciar cotas e restrições de serviço

Gerencie cotas e restrições de serviço. Para arquiteturas de carga de trabalho baseadas na nuvem, existem cotas de serviço (que também são chamadas de limites de serviço). Essas cotas existem para evitar o provisionamento acidental de mais recursos do que o necessário. Elas também ajudam a limitar as taxas de solicitação nas operações de API para proteger os serviços contra abusos. Há também restrições de recursos, por exemplo, a taxa de bits que você pode enviar até o cabo de fibra ótica ou a quantidade de armazenamento em um disco físico.

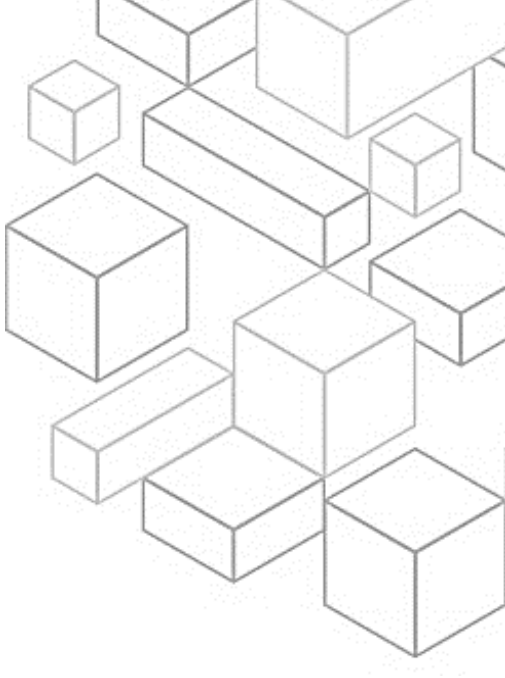
É uma boa ideia estar ciente das cotas padrão e das solicitações de aumento de cota para sua arquitetura de carga de trabalho. Além disso, você deve saber quais restrições de recursos, como disco ou rede, são potencialmente



impactantes. Você também pode gerenciar cotas de serviço em contas e Regiões. Se você estiver usando várias contas ou Regiões AWS, certifique-se de solicitar as cotas apropriadas em todos os ambientes nos quais as cargas de trabalho de produção são executadas. As cotas de serviço são rastreadas por conta. Salvo indicação em contrário, cada cota é específica da Região AWS. Além dos ambientes de produção, gerencie também as cotas em todos os ambientes de não produção aplicáveis para que os testes e o desenvolvimento não sejam prejudicados.

Em seguida, acomode cotas de serviço fixas e restrições por meio da arquitetura. Esteja ciente das cotas de serviço e dos recursos físicos imutáveis e arquitete para evitar que eles afetem a confiabilidade. Você também pode monitorar e gerenciar cotas. Avalie seu uso potencial e, em seguida, aumente suas cotas adequadamente com espaço para o crescimento planejado do uso. Automatize o gerenciamento de cotas implementando ferramentas para alertá-lo quando um limite se aproximar. Você pode automatizar as solicitações de aumento de cota usando Service Quotas APIs.

Por fim, garanta que haja uma lacuna suficiente entre as cotas atuais e o uso máximo para acomodar o failover. Quando um recurso é reprovado, ele ainda pode ser contabilizado nas cotas até que termine com sucesso. Verifique se suas cotas cobrem a sobreposição de todos os recursos com falha com as substituições antes que os recursos com falha terminem. Considere uma falha na Zona de Disponibilidade ao calcular essa lacuna.

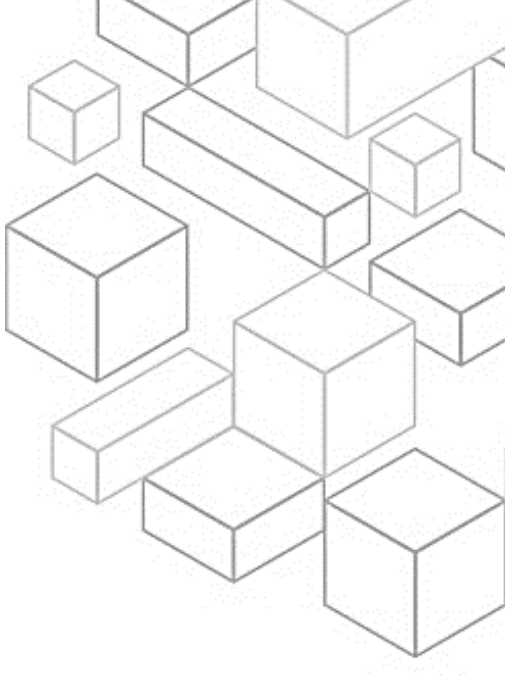


1.12 Planejar a topologia da rede

Planeje sua topologia de rede. As cargas de trabalho geralmente existem em vários ambientes. Isso inclui vários ambientes de nuvem (acessíveis publicamente e privados) e, possivelmente, sua infraestrutura de data center existente. Os planos devem incluir considerações de rede, como conectividade intrassistema e intersistema, gerenciamento de endereços IP públicos, gerenciamento de endereços IP privados e resolução de nomes de domínio. Ao arquitetar sistemas que usam redes baseadas em endereços IP, é necessário planejar a topologia da rede e prever possíveis falhas. É importante acomodar o crescimento futuro e a integração com outros sistemas e suas redes.

Lembre-se de usar conectividade de rede altamente disponível para os endpoints públicos de sua carga de trabalho. Esses endpoints e o roteamento para eles devem estar altamente disponíveis. Para isso, use DNS altamente disponível, rede de entrega de conteúdo (CDN), gateway de API, balanceamento de carga ou proxies reversos. Forneça conectividade redundante entre redes privadas na nuvem e em ambientes on-premises. Use várias conexões do AWS Direct Connect ou túneis de rede privada virtual entre redes privadas implantadas separadamente. Você também pode usar vários locais do Direct Connect para ter alta disponibilidade. Se estiver usando várias Regiões AWS, garanta a redundância em pelo menos duas delas.

Você também pode garantir que a alocação da sub-rede IP leve em conta a expansão e a disponibilidade. Os intervalos de endereços IP do Amazon Virtual Private Cloud, ou Amazon VPC, devem ser grandes o suficiente para acomodar os requisitos de carga de trabalho. Isso inclui levar em conta a expansão futura e a



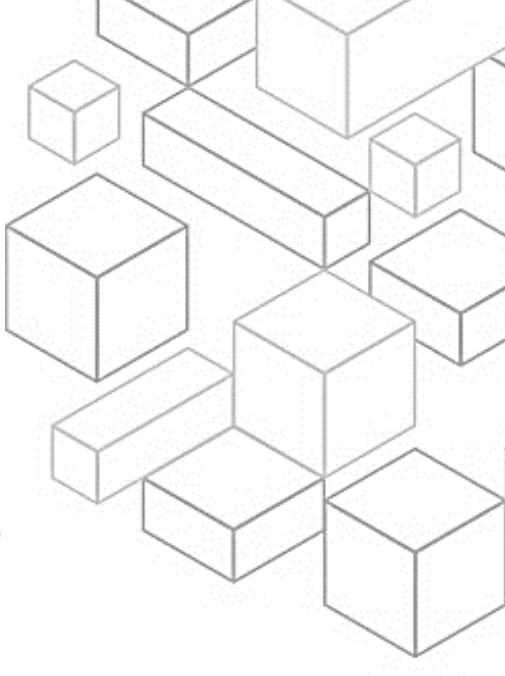
alocação de endereços IP para sub-redes nas Zonas de Disponibilidade. Isso também inclui balanceadores de carga, instâncias EC2 e aplicações baseadas em contêineres.

Considere as topologias hub-and-spoke sobre a malha muitos-para-muitos se mais de dois espaços de endereço de rede, como VPCs e redes on-premises, estiverem conectados por meio de peering de VPC, Direct Connect ou VPN. O AWS Transit Gateway é um exemplo de um modelo hub-and-spoke.

Por fim, implemente intervalos de endereços IP privados não sobrepostos em todos os espaços de endereços privados aos quais eles estão conectados. Os intervalos de endereços IP de cada uma de suas VPCs não devem se sobrepor quando examinados ou conectados por meio de VPN. Da mesma forma, você deve evitar conflitos de endereços IP entre uma VPC e ambientes on-premises ou com outros provedores de nuvem que você usa. Você também deve ter uma maneira de alocar intervalos de endereços IP privados quando necessário.

1.13 Arquitetura de carga de trabalho

A arquitetura de carga de trabalho é a próxima área de práticas recomendadas de confiabilidade. Uma carga de trabalho confiável começa com decisões iniciais de design para o software e a infraestrutura. Suas escolhas de arquitetura afetarão o comportamento da carga de trabalho em todos os seis pilares do AWS Well-Architected. Para obter confiabilidade, há padrões específicos a serem incluídos. A seção a seguir explica as práticas recomendadas a serem usadas com esses padrões para garantir a confiabilidade. Você aprenderá a projetar sua arquitetura de serviço de carga de trabalho. Você também considerará as



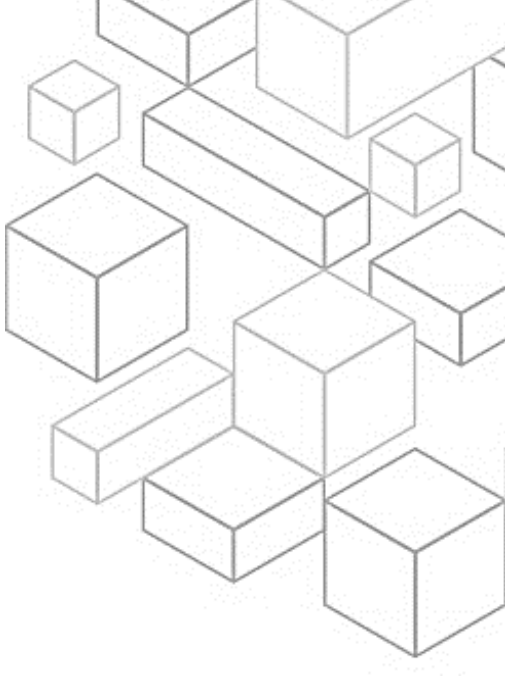
interações em um sistema distribuído para evitar, atenuar ou resistir a falhas.

1.14 Projetar sua arquitetura de serviço de carga de trabalho

Projete sua arquitetura de serviço de carga de trabalho. Crie cargas de trabalho altamente dimensionáveis e confiáveis usando uma arquitetura orientada a serviços ou uma arquitetura de microsserviços. A arquitetura orientada a serviços é a prática de tornar os componentes de software reutilizáveis por meio de interfaces de serviço. A arquitetura de microsserviços vai além para tornar os componentes menores e mais simples.

Para projetar sua arquitetura, primeiro escolha como segmentar sua carga de trabalho. A segmentação da carga de trabalho é importante ao determinar os requisitos de resiliência de sua aplicação. Evite a arquitetura monolítica sempre que possível. Em vez disso, considere cuidadosamente quais componentes da aplicação podem ser divididos em microsserviços. Dependendo dos requisitos de sua aplicação, isso pode acabar combinando a arquitetura orientada a serviços com microsserviços, sempre que possível. As cargas de trabalho que são capazes de ser stateless são mais capazes de serem implantadas como microsserviços.

Em seguida, crie serviços focados em domínios e funcionalidades comerciais específicos. A arquitetura orientada a serviços cria serviços com funções bem delineadas, definidas pelas necessidades comerciais. Os microsserviços usam modelos de domínio e contexto delimitado para limitar ainda mais isso, de modo que cada serviço faça apenas uma coisa. O foco na funcionalidade ajuda a diferenciar os requisitos de confiabilidade de cada serviço e a direcionar melhor

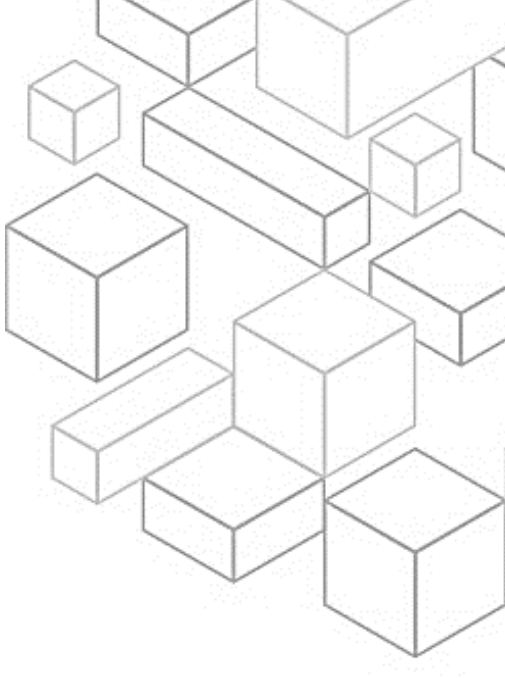


os investimentos. Também é possível fazer um dimensionamento mais rápido da organização com um problema comercial conciso e uma pequena equipe associada a cada serviço.

Por fim, forneça contratos de serviço por API. Os contratos de serviço são acordos documentados entre as equipes sobre a integração de serviços. Eles incluem uma definição de API legível por máquina, limites de taxa e expectativas de desempenho. Uma estratégia de versionamento ajuda seus clientes a continuar usando a API existente para migrar suas aplicações para uma API mais nova quando estiverem prontos. A implantação pode ocorrer a qualquer momento desde que o contrato não seja violado. A equipe do provedor de serviços pode usar a pilha de tecnologia de sua escolha para satisfazer o contrato da API. Da mesma forma, o consumidor do serviço pode usar sua própria tecnologia.

1.15 Projetar interações em um sistema distribuído para evitar falhas

Projete interações em um sistema distribuído para evitar falhas. Os sistemas distribuídos dependem de redes de comunicação para interconectar componentes, como servidores ou serviços. Sua carga de trabalho deve operar de forma confiável, apesar da perda de dados ou latência nessas redes. Os componentes do sistema distribuído devem operar de forma que não impacte negativamente outros componentes ou a carga de trabalho. Essas práticas recomendadas podem ajudar a evitar falhas e melhorar o tempo médio entre falhas, ou MTBF.

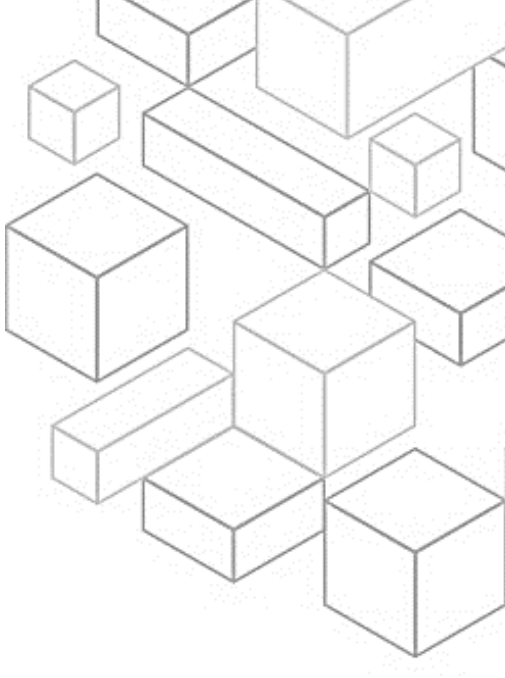


Primeiro, identifique que tipo de sistema distribuído é necessário. Os sistemas distribuídos em tempo real exigem que as respostas sejam dadas de forma síncrona e rápida. Os sistemas flexíveis em tempo real têm uma janela de tempo mais generosa, de minutos ou mais, para resposta. Os sistemas off-line tratam as respostas por meio de processamento em batch ou assíncrono. Os sistemas distribuídos em tempo real rígido têm os requisitos de confiabilidade mais rigorosos.

Em segundo lugar, implemente dependências com acoplamento fraco. Dependências como sistemas de enfileiramento, sistemas de streaming, fluxos de trabalho e balanceadores de carga são fracamente acoplados. O acoplamento fraco ajuda a isolar o comportamento de um componente de outros componentes que dependem dele, aumentando a resiliência e a agilidade.

Terceiro, faça um trabalho constante. Os sistemas podem falhar quando há mudanças grandes e rápidas na carga. Por exemplo, se a sua carga de trabalho estiver fazendo uma health check que monitora a health de milhares de servidores, ela deverá enviar o mesmo tamanho de payload (um snapshot completo do estado atual) todas as vezes. Independentemente de nenhum servidor estar falhando ou de todos eles, o sistema de health check está fazendo um trabalho constante, sem grandes e rápidas alterações.

Por fim, torne todas as respostas idempotentes. Um serviço idempotente promete que cada solicitação seja concluída exatamente uma vez, de modo que fazer várias solicitações idênticas tem o mesmo efeito que fazer uma única solicitação. Um serviço idempotente ajuda um cliente a implementar novas tentativas sem medo de que uma solicitação seja processada erroneamente



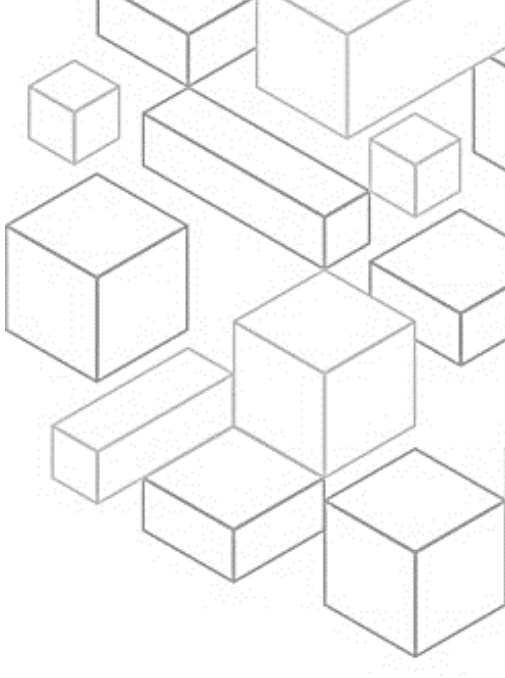
várias vezes. Para fazer isso, os clientes podem emitir solicitações de API com um token de idempotência - o mesmo token é usado sempre que a solicitação é repetida. Uma API de serviço idempotente usa o token para retornar uma resposta idêntica à resposta que foi retornada na primeira vez em que a solicitação foi concluída.

1.16 Projetar interações em um sistema distribuído para mitigar

Projete interações em um sistema distribuído para mitigar ou resistir a falhas. Os sistemas distribuídos dependem de redes de comunicação para interconectar componentes, como servidores ou serviços. Sua carga de trabalho deve operar de forma confiável, apesar da perda de dados ou da latência nessas redes. Os componentes do sistema distribuído devem operar de forma que não impacte negativamente outros componentes ou a carga de trabalho. Essas práticas recomendadas ajudam as cargas de trabalho a resistir a estresses ou falhas a se recuperar mais rapidamente e a atenuar o impacto dessas deficiências. O resultado é um melhor tempo médio de recuperação, ou MTTR.

Implemente a degradação graciosa para transformar as dependências rígidas aplicáveis em dependências flexíveis. Quando as dependências de um componente não estão saudáveis, o próprio componente ainda pode funcionar, embora de forma degradada. Por exemplo, quando uma chamada de dependência falhar, faça o failover para uma resposta estática predeterminada.

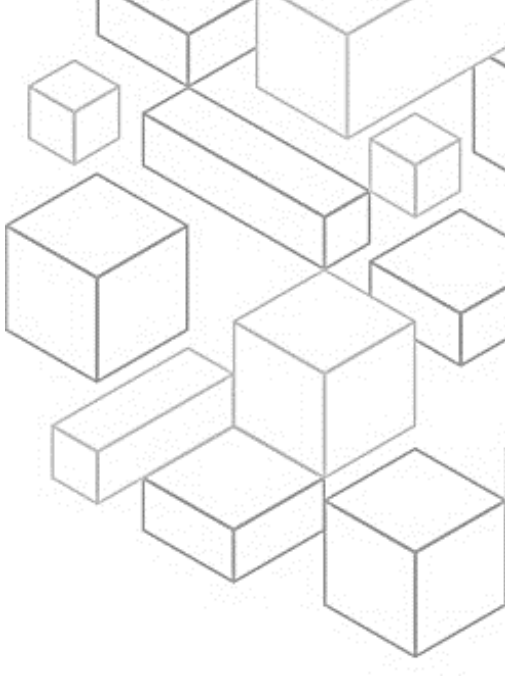
A limitação de solicitações é um padrão de atenuação para responder a um aumento inesperado na demanda. Algumas solicitações são atendidas, mas as que ultrapassam um limite definido são rejeitadas e retornam uma mensagem



indicando que foram limitadas. A expectativa dos clientes é que eles recuem e abandonem a solicitação ou tentem novamente em um ritmo mais lento.

Controle e limite as chamadas de repetição. Use o backoff exponencial para tentar novamente após intervalos progressivamente maiores. Apresente jitter para randomizar esses intervalos de novas tentativas e limitar o número máximo de tentativas. Falhe rapidamente e limite filas: se a carga de trabalho não puder responder com êxito a uma solicitação, falhe rapidamente. Isso ajuda a liberar os recursos associados a uma solicitação e permite que o serviço se recupere se estiver ficando sem recursos. Se a carga de trabalho for capaz de responder com êxito, mas a taxa de solicitações for muito alta, você poderá usar uma fila para armazenar as solicitações em buffer. No entanto, não permita filas longas que possam resultar no atendimento de solicitações obsoletas das quais o cliente já tenha desistido.

Defina os tempos limite do cliente adequadamente, verifique-os sistematicamente e não confie nos valores padrão, pois eles geralmente são definidos como muito altos. Essa prática recomendada se aplica ao lado do cliente, ou remetente, da solicitação. Torne os serviços stateless sempre que possível. Os serviços não devem exigir estado ou devem descarregar o estado de forma que, entre diferentes solicitações de clientes, não haja dependência de dados armazenados localmente no disco e na memória. Isso ajuda os servidores a serem substituídos à vontade, sem causar um impacto na disponibilidade. Por fim, implemente alavancas de emergência. Esses processos rápidos podem reduzir o impacto da disponibilidade em sua carga de trabalho.



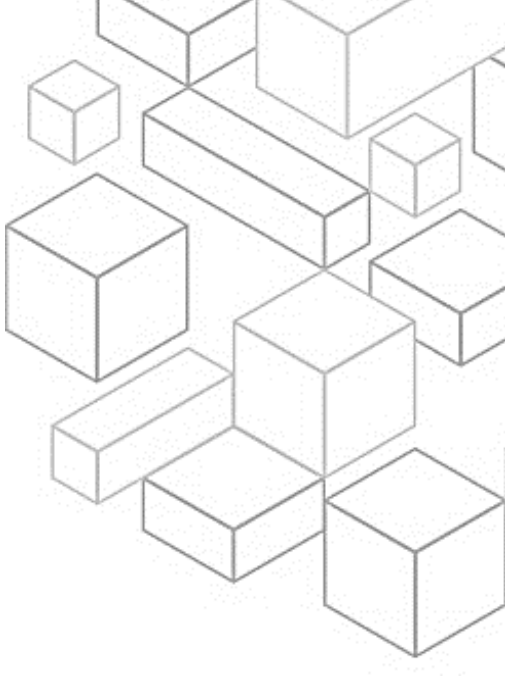
1.17 Gerenciamento de alterações

O gerenciamento de alterações é a próxima área de práticas recomendadas de confiabilidade. Alterações à sua carga de trabalho ou ao seu ambiente devem ser previstas e acomodadas para que a operação da carga de trabalho seja confiável. As alterações incluem aquelas impostas à sua carga de trabalho, como picos de demanda. Elas também incluem as de origem interna, como implantações de recursos e patches de segurança. A seção a seguir explica as práticas recomendadas para o gerenciamento de alterações. Você aprenderá a monitorar os recursos da carga de trabalho, a projetar uma carga de trabalho para se adaptar às mudanças na demanda e a implementar mudanças.

1.18 Monitorar recursos de carga de trabalho

Monitore recursos de carga de trabalho. Os logs e as métricas são ferramentas poderosas para obter informações sobre a integridade da sua carga de trabalho. Você pode configurar sua carga de trabalho para monitorar logs e métricas. Em seguida, eles podem enviar notificações quando os limites são ultrapassados ou quando ocorrem eventos significativos. O monitoramento ajuda sua carga de trabalho a reconhecer quando os limites de baixo desempenho são ultrapassados ou quando ocorrem falhas, para que ela possa se recuperar automaticamente em resposta. Monitore todos os componentes da carga de trabalho. Isso significa que você pode monitorar os componentes da carga de trabalho usando o Amazon CloudWatch ou ferramentas de terceiro. Monitore os serviços AWS com o AWS Health Dashboard.

Defina e calcular métricas. Armazene dados de log e aplique filtros quando



necessário para calcular métricas, como contagens de um evento de log específico ou latência calculada a partir de carimbos de data/hora de eventos de log. Envie notificações. As organizações que precisam saber podem receber notificações quando ocorrerem eventos significativos.

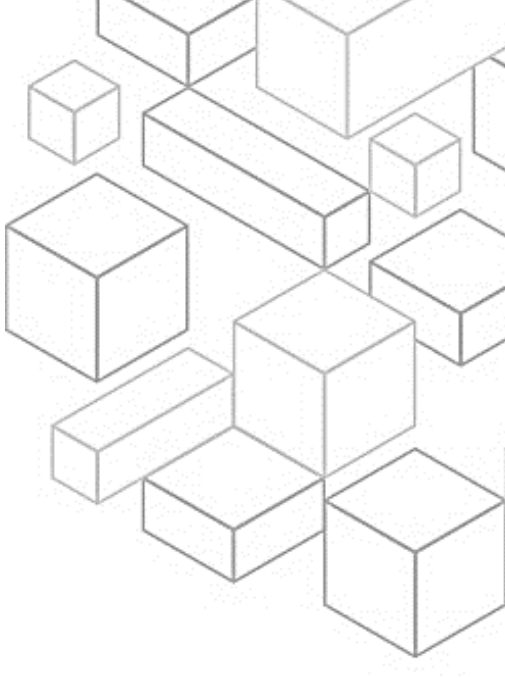
Automatize as respostas implementando o processamento e os alarmes em tempo real. Use a automação para tomar ação quando um evento for detectado, por exemplo, para substituir componentes com falha. Realize analytics coletando arquivos de log e históricos de métricas e, em seguida, analisando-os para obter tendências mais amplas e informações sobre a carga de trabalho.

Conduza análises regularmente. Revise frequentemente como o monitoramento da carga de trabalho é implementado e atualize-o com base em eventos e mudanças significativas. O monitoramento eficaz é orientado pelas principais métricas de negócios. Garanta que essas métricas sejam acomodadas em sua carga de trabalho à medida que as prioridades comerciais mudam.

Por fim, monitore o rastreamento de ponta a ponta das solicitações em seu sistema. Use o AWS X-Ray ou ferramentas de terceiro para que os desenvolvedores possam analisar e depurar rapidamente os sistemas distribuídos. Isso os ajuda a entender o desempenho de suas aplicações e serviços subjacentes.

1.19 Projetar uma carga de trabalho para se adaptar às mudanças na demanda

Projete uma carga de trabalho para se adaptar às mudanças na demanda. Uma



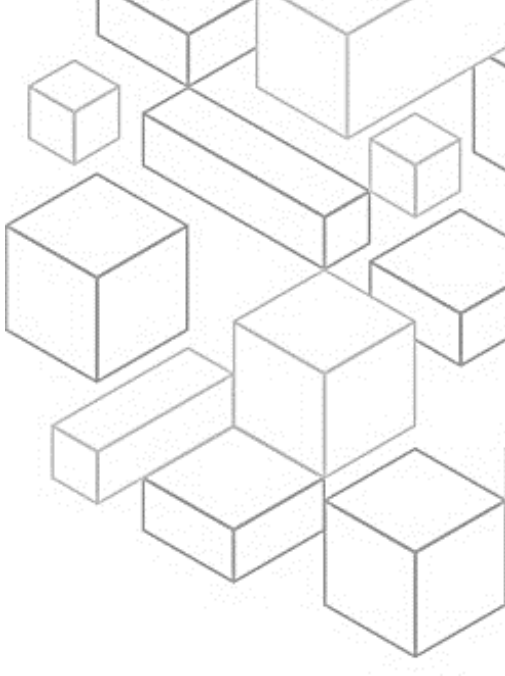
carga de trabalho dimensionável oferece elasticidade para adicionar ou remover recursos automaticamente, de modo que eles se aproximem da demanda atual em um determinado momento. Use a automação ao obter ou scaling recursos. Ao substituir recursos com problemas ou scaling sua carga de trabalho, automatize o processo usando serviços gerenciados da AWS, como o Amazon S3 e o AWS Auto Scaling. Você também pode usar ferramentas de terceiro e SDKs do AWS para automatizar o scaling.

Obtenha recursos após a detecção de comprometimento de uma carga de trabalho. Dimensione os recursos de forma reativa quando necessário, se a disponibilidade for afetada, para restaurar a disponibilidade da carga de trabalho. Primeiro, você deve configurar as health checks e os critérios dessas verificações para indicar quando a disponibilidade é afetada pela falta de recursos. Em seguida, notifique a equipe apropriada para dimensionar manualmente o recurso ou ative a automação para dimensioná-lo automaticamente.

Obtenha recursos ao detectar que são necessários mais recursos para uma carga de trabalho. Dimensione os recursos de forma proativa para atender à demanda e evitar o impacto na disponibilidade. Adote uma metodologia de teste de carga para medir se a ação de scaling atende aos requisitos de carga de trabalho.

1.20 Implementar alterações

As alterações controladas são necessárias para implantar novas funcionalidades e ajudar a garantir que as cargas de trabalho e o ambiente operacional estejam executando software conhecido e com patches adequados. Se essas alterações



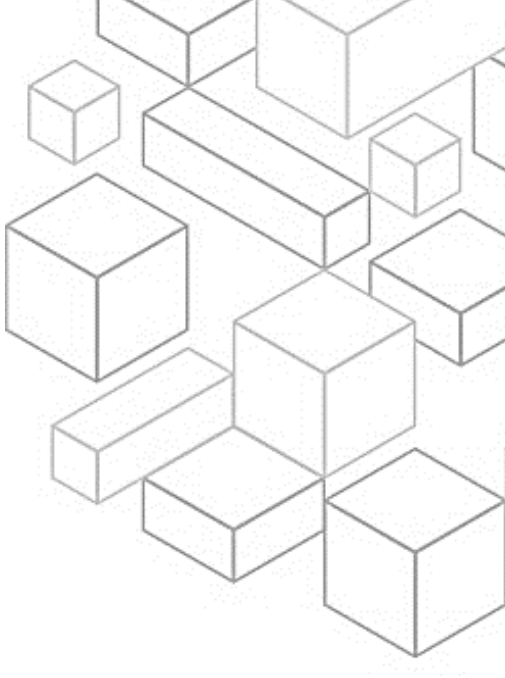
não forem controladas, será difícil prever o efeito dessas alterações ou resolver os problemas que surgirem devido a elas.

Use runbooks para atividades padrão, como a implantação. Runbooks são procedimentos predefinidos usados para alcançar resultados específicos. Use runbooks para executar atividades padrão, seja de forma manual ou automática. Os exemplos incluem a implantação de uma carga de trabalho, a aplicação de patches em uma carga de trabalho ou a realização de modificações no DNS.

Integre o teste funcional como parte de sua implantação. Os testes funcionais são executados como parte da implantação automatizada. Se os critérios de sucesso não forem atendidos, o pipeline será interrompido ou revertido. Esses testes são executados em um ambiente de pré-produção, que é preparado antes da produção no pipeline. O ideal é que isso seja feito como parte de um pipeline de implantação.

Integre o teste de resiliência como parte de sua implantação. Testes de resiliência, usando os princípios da engenharia do caos, são preparados e executados como parte do pipeline de implantação automatizada em um ambiente de pré-produção. Eles também devem ser executados na produção como parte dos dias de teste.

Implante usando uma infraestrutura imutável. A infraestrutura imutável é um modelo que exige que nenhuma atualização, patch de segurança ou alteração de configuração ocorra nas cargas de trabalho de produção. Quando uma alteração é necessária, a arquitetura é construída em uma nova infraestrutura e implantada em produção. Automatize implantações e aplicação de patches para



eliminar o impacto negativo.

1.21 Gerenciamento de falhas

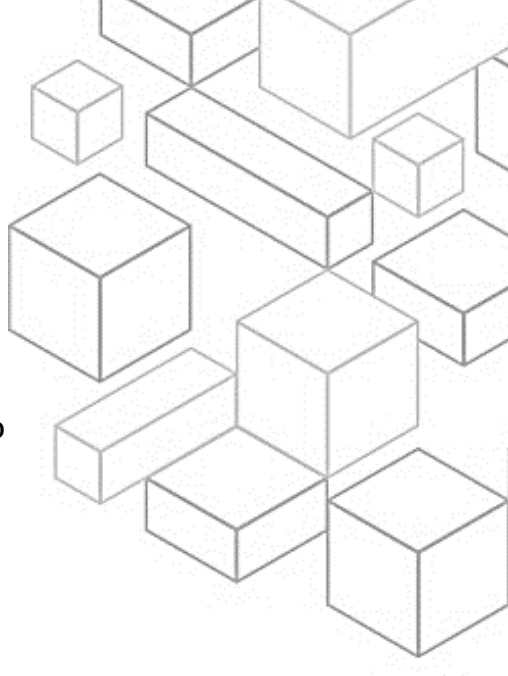
O gerenciamento de falhas é a próxima área de práticas recomendadas de confiabilidade. As falhas de baixo nível dos componentes de hardware são algo com que se lida todos os dias em um data center on-premises. Na nuvem, entretanto, você deve estar protegido contra a maioria desses tipos de falhas. Independentemente do seu provedor de nuvem, há a possibilidade de falhas afetarem sua carga de trabalho. Portanto, você deve tomar medidas para implementar a resiliência se precisar que sua carga de trabalho seja confiável.

Um pré-requisito para aplicar as práticas recomendadas discutidas aqui é garantir que as pessoas que projetam, implementam e operam suas cargas de trabalho estejam cientes dos objetivos comerciais e das metas de confiabilidade para alcançá-los. Essas pessoas devem estar cientes e treinadas para esses requisitos de confiabilidade.

As seções a seguir explicam as práticas recomendadas de gerenciamento de falhas para evitar o impacto na sua carga de trabalho.

1.22 Backup de dados

Faça backup de dados, aplicações e configurações para atender aos requisitos de objetivos de tempo de recuperação (RTO) e objetivos de ponto de recuperação (RPO). Identifique e faça backup de todos os dados que precisam de backup ou reproduza os dados a partir de fontes. Ao selecionar uma estratégia de backup, considere o tempo necessário para recuperar os dados. O tempo necessário para



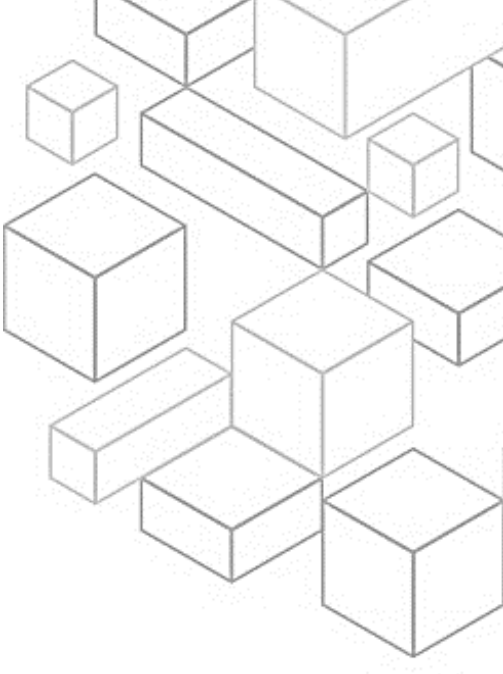
recuperar os dados depende do tipo de backup (no caso de uma estratégia de backup) ou da complexidade do mecanismo de reprodução de dados. Esse tempo deve estar dentro do RTO para a carga de trabalho.

Proteja e/ou criptografe backups. Controle e detecte o acesso aos backups usando autenticação e autorização, como o AWS Identity and Access Management, ou IAM. Previna e detecte se a integridade dos dados dos backups está comprometida usando criptografia. Realize backup de dados automaticamente. Configure os backups para serem feitos automaticamente com base em uma programação periódica informada pelo RPO ou por alterações no conjunto de dados. Conjuntos de dados críticos com baixos requisitos de perda de dados precisam ser copiados automaticamente com frequência, enquanto dados menos críticos, nos quais alguma perda é aceitável, podem ser copiados com menos frequência.

Realize recuperação periódica de dados para verificar a integridade e os processos de backup. Valide se a implementação do seu processo de backup atende ao RTO e ao RPO realizando um teste de recuperação.

1.23 Usar o isolamento de falhas para proteger sua carga de trabalho

Use o isolamento de falhas para proteger sua carga de trabalho. Limites isolados de falhas limitam o efeito de uma falha dentro de uma carga de trabalho a um número limitado de componentes. Os componentes fora do limite não são afetados pela falha. Ao usar vários limites isolados de falhas, você pode limitar o impacto sobre sua carga de trabalho. Primeiro, implante a carga de trabalho em



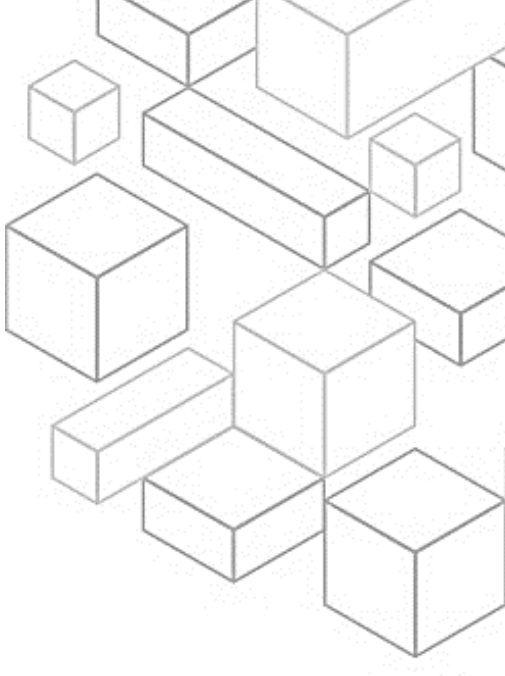
vários locais. Distribua dados e recursos de carga de trabalho em várias Zona de Disponibilidade ou, quando necessário, entre Regiões AWS. Esses locais podem ser tão diversos quanto necessário.

Em seguida, selecione os locais apropriados para sua implantação em vários locais. Para obter alta disponibilidade, sempre tente implantar os componentes de sua carga de trabalho em várias Zonas de Disponibilidade. Para cargas de trabalho com requisitos extremos de resiliência, avalie cuidadosamente as opções de uma arquitetura de multirregiões. Você também pode automatizar a recuperação de componentes restritos a um único local. Se os componentes da carga de trabalho só puderem ser executados em uma única Zona de Disponibilidade ou em um data center on-premises, implante o recurso para fazer uma reconstrução completa da carga de trabalho dentro dos objetivos de recuperação definidos.

Use arquiteturas de anteparo para limitar o escopo do impacto. Como os anteparos em um navio, esse padrão garante que uma falha seja contida em um pequeno subconjunto de solicitações ou clientes. Isso ajuda a limitar o número de solicitações prejudicadas para que a maioria possa continuar sem erros. Os anteparos para dados são geralmente chamados de partições, enquanto os anteparos para serviços são conhecidos como células.

1.24 Projetar a carga de trabalho para resistir a falhas de componentes

Projete a carga de trabalho para resistir a falhas de componentes. As cargas de trabalho com requisitos de alta disponibilidade e baixo tempo médio de

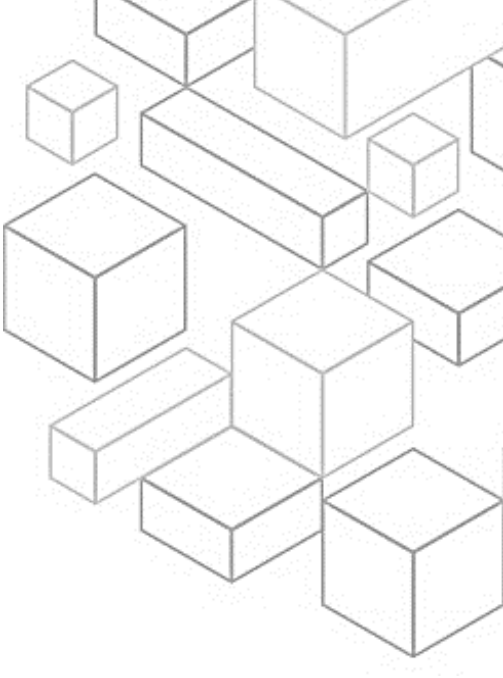


recuperação, ou MTTR, devem ser arquitetadas para resiliência. Monitore todos os componentes da carga de trabalho para detectar falhas. Monitore continuamente a integridade de sua carga de trabalho para que você e seus sistemas automatizados estejam cientes da degradação ou da falha assim que elas ocorrerem. Monitore os KPIs com base no valor comercial. Em seguida, faça o failover para recursos íntegros. Garanta que, se ocorrer uma falha de recurso, os recursos íntegros possam continuar a atender às solicitações. Em caso de falhas de local, como Zona de Disponibilidade ou Região AWS, certifique-se de que você tenha sistemas implementados para fazer o failover para recursos íntegros em locais não afetados.

Automatize a recuperação em todas as camadas. Após a detecção de uma falha, use recursos automatizados para executar ações de correção. A capacidade de reiniciar é uma ferramenta importante para corrigir falhas. Conforme discutido anteriormente para sistemas distribuídos, uma prática recomendada é tornar os serviços stateless sempre que possível. Isso evita a perda de dados ou a disponibilidade na reinicialização.

Confie no plano de dados, e não no plano de controle, durante a recuperação. O plano de controle é usado para configurar recursos, e o plano de dados fornece serviços. Normalmente, os planos de dados têm objetivos de projeto de disponibilidade mais altos do que os planos de controle e, em geral, são menos complexos. Ao implementar respostas de recuperação ou atenuação a eventos que podem afetar a resiliência, o uso de operações de plano de controle pode reduzir a resiliência geral da sua arquitetura.

Use a estabilidade estática para evitar o comportamento bimodal.

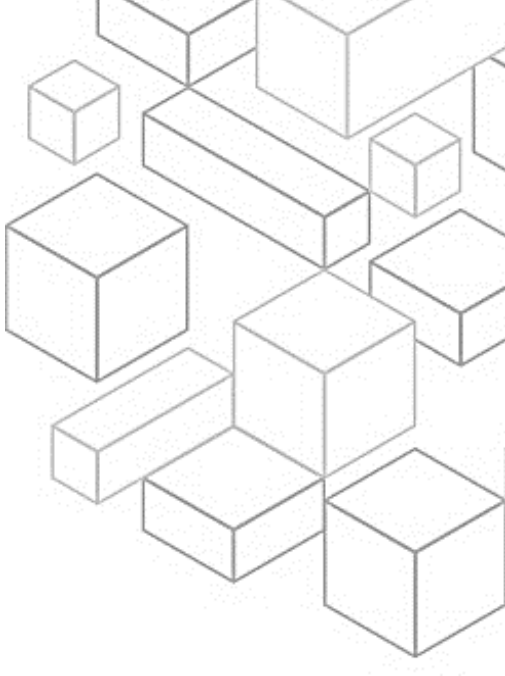


Comportamento bimodal é quando sua carga de trabalho exibe um comportamento diferente nos modos normal e de falha. Um exemplo é confiar na inicialização de novas instâncias se uma zona de disponibilidade falhar. Em vez disso, você deve criar cargas de trabalho que sejam estaticamente estáveis e operem em apenas um modo. Nesse caso, provisione instâncias suficientes em cada Zona de Disponibilidade para lidar com a carga de trabalho se uma zona for removida. Em seguida, use o Elastic Load Balancing ou as health checks do Amazon Route 53 para desviar a carga das instâncias prejudicadas.

Envie notificações quando os eventos afetarem a disponibilidade. As notificações são enviadas após a detecção de eventos significativos, mesmo que o problema causado pelo evento tenha sido resolvido automaticamente. Por fim, arquitete seu produto para atender às metas de disponibilidade e aos contratos de nível de serviço de tempo de atividade, ou SLAs. Se você publicar ou concordar privadamente com metas de disponibilidade ou SLAs de tempo de atividade, verifique se sua arquitetura e seus processos operacionais foram projetados para dar suporte a eles.

1.25 Testar a confiabilidade

Teste a confiabilidade. Depois de projetar sua carga de trabalho para ser resiliente aos estresses da produção, o teste é a única maneira de garantir que ela funcionará conforme projetado para oferecer a resiliência esperada. Teste para validar se a carga de trabalho atende aos requisitos funcionais e não funcionais, pois bugs ou gargalos de desempenho podem afetar sua confiabilidade. Teste a resiliência de sua carga de trabalho para ajudar a



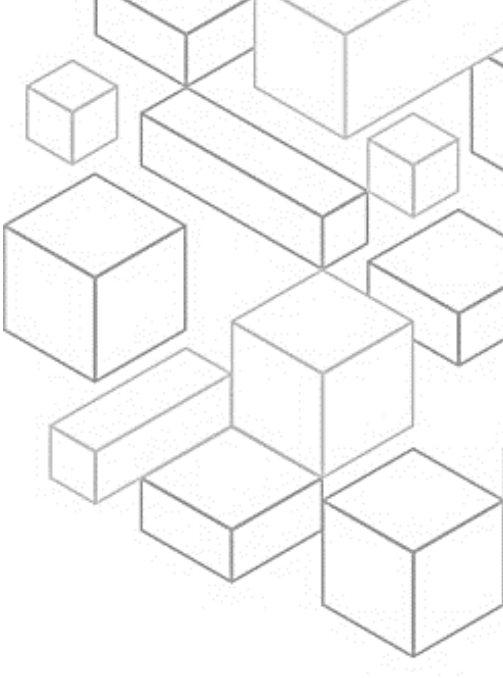
encontrar bugs latentes que só aparecem na produção. Faça esses testes regularmente.

Use playbooks para investigar falhas. Configure respostas consistentes e rápidas para cenários de falha que não são bem compreendidos, documentando o processo de investigação em playbooks. Os playbooks são as etapas predefinidas executadas para identificar os fatores que contribuem para um cenário de falha. Os resultados de qualquer etapa do processo são usados para determinar as próximas etapas até que o problema seja identificado ou encaminhado.

Realize análise pós-incidente. Analise os eventos que afetam o cliente e identifique os fatores contribuintes e os itens de ação preventiva. Use essas informações para desenvolver mitigações para limitar ou evitar a recorrência. Desenvolva procedimentos para respostas rápidas e eficazes. Comunique os fatores que contribuíram para isso e as ações corretivas conforme apropriado, adaptadas aos públicos-alvo. Documente um método para comunicar essas causas a outras pessoas, conforme necessário.

Teste os requisitos funcionais usando técnicas como testes de unidade e testes de integração para validar a funcionalidade necessária. Você também deve testar os requisitos de scaling e desempenho. Use técnicas como o teste de carga para validar se a carga de trabalho atende aos requisitos de scaling e desempenho. Teste a resiliência usando a engenharia do caos. Execute experimentos de caos regularmente em ambientes que estejam em produção ou o mais próximo possível da produção para entender como o seu sistema responde a condições adversas.

Por fim, realize dias de teste regularmente. Use os dias de teste para exercitar



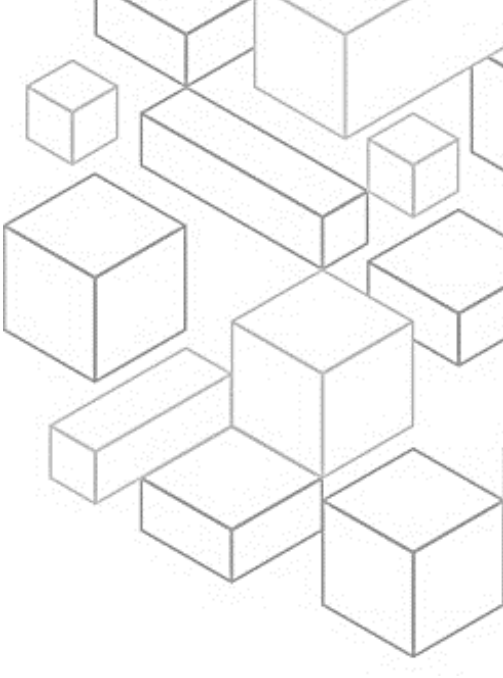
consistentemente seus procedimentos de resposta a eventos e falhas o mais próximo possível da produção. Inclua ambientes de produção e as pessoas que estarão envolvidas em cenários reais de falha. Os dias de teste aplicam medidas para ajudar a garantir que os eventos de produção não afetem os usuários.

1.26 Planejar para a recuperação de desastres

Planeje para a recuperação de desastres. Ter backups e componentes de carga de trabalho redundantes é o início de sua estratégia de recuperação de desastres. RTO e RPO são seus objetivos para restaurar sua carga de trabalho. Defina-os com base nas necessidades da empresa. Implemente uma estratégia para atender a esses objetivos, considerando os locais e a função dos recursos e dados da carga de trabalho. A probabilidade de interrupção e o custo da recuperação também são fatores importantes que ajudam a informar o valor comercial do fornecimento de recuperação de desastres para uma carga de trabalho.

Tanto a disponibilidade quanto a recuperação de desastres dependem das mesmas práticas recomendadas. Os exemplos incluem o monitoramento de falhas, a implantação em vários locais e a execução de failover automático. No entanto, a disponibilidade se concentra nos componentes da carga de trabalho, enquanto a recuperação de desastres se concentra em cópias discretas de toda a carga de trabalho. A recuperação de desastres tem objetivos diferentes da disponibilidade, concentrando-se no tempo de recuperação após um desastre.

Defina objetivos de recuperação para tempo de inatividade e perda de dados. A carga de trabalho tem um RTO e um RPO. O RTO é o atraso máximo aceitável



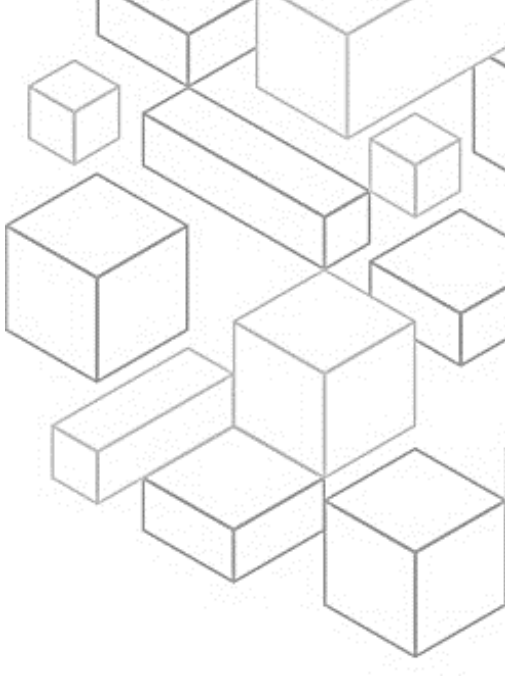
entre a interrupção e a restauração do serviço. Isso determina o que é considerado uma janela de tempo aceitável quando o serviço está indisponível. O RPO é a quantidade máxima aceitável de tempo desde o último ponto de recuperação de dados. Isso determina o que é considerado uma perda aceitável de dados entre o último ponto de recuperação e a interrupção do serviço.

Use estratégias de recuperação definidas para atender aos objetivos de recuperação. Defina uma estratégia de recuperação de desastres que atenda aos objetivos de recuperação de sua carga de trabalho. Escolha uma estratégia, como backup e restauração, ativo-passivo ou ativo-ativo. Teste regularmente o failover em seu site de recuperação para garantir a operação adequada e que o RTO e o RPO sejam atendidos.

Gerencie o desvio de configuração no site ou na Região de recuperação de desastres. Assegure-se de que a infraestrutura, os dados e a configuração estejam de acordo com o necessário no local ou na região de recuperação de desastres. Por exemplo, verifique se as AMIs e as cotas de serviço estão atualizadas. Por fim, use a AWS ou ferramentas de terceiros para automatizar a recuperação do sistema e encaminhar o tráfego para o site ou a Região de recuperação de desastres.

1.30 Resumo

Neste módulo, você aprendeu sobre a importância do pilar de confiabilidade no Well-Architected Framework e a proposta de valor para a confiabilidade em suas arquiteturas. Você também aprendeu sobre os princípios de design e as práticas



recomendadas do pilar de confiabilidade.

1.31 Agradecemos sua atenção

Agradecemos sua participação!