

Recursos e terminologia em Hubs de Eventos do Azure

Artigo • 22/12/2021 • 18 minutos para o fim da leitura

Esta página é útil?  

Neste artigo

[Namespace](#)

[Editores de eventos](#)

[Capturar](#)

[Partições](#)

[Tokens SAS](#)

[Consumidores de evento](#)

[Próximas etapas](#)

Os Hubs de Evento do Azure é um serviço de processamento de evento escalonável que recebe e processa grandes volumes de eventos e dados, com baixa latência e alta confiabilidade. Confira [O que são Hubs de Eventos?](#) para obter uma visão geral de alto nível.

Este artigo se baseia nas informações do [artigo de visão geral](#) e fornece detalhes técnicos e de implementação sobre recursos e componentes dos Hubs de Eventos.

Dica

O suporte a protocolo para clientes Apache Kafka (versões = 1.0) fornece pontos de extremidade de rede que habilitam aplicativos criados para usar o Apache Kafka com qualquer cliente que use o Hubs de Eventos. A maioria dos aplicativos Kafka existentes pode simplesmente ser reconfigurada para apontar para um namespace do Hub de Eventos em vez de um servidor de inicialização de cluster Kafka.

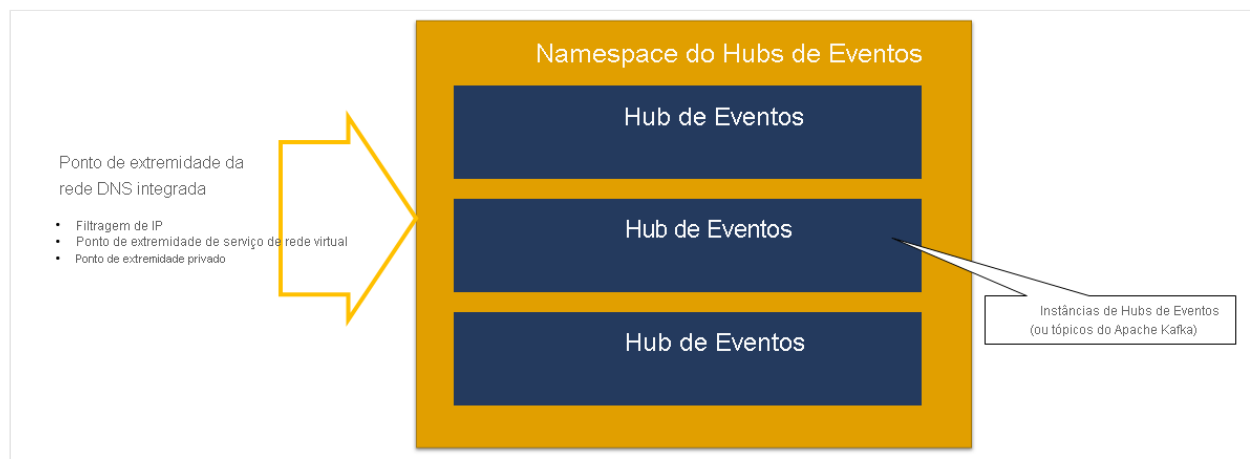
Da perspectiva de custo, esforço operacional e confiabilidade, o Hubs de Eventos do Azure é uma ótima alternativa para implantar e operar clusters Kafka e Zookeeper próprios e para ofertas de Kafka como serviço não nativas para o Azure.

Além de obter a mesma funcionalidade básica do agente de Apache Kafka, você também recebe acesso a recursos do Hub de Eventos do Azure, como envio em lote automático e arquivamento por meio da **Captura do Hubs de Eventos**, dimensionamento automático e balanceamento, recuperação de desastres, suporte

a zona de disponibilidade neutra em termos de custo, integração de rede segura e flexível e suporte a vários protocolos, como o AMQP-over-WebSockets.

Namespace

Um namespace dos Hubs de Eventos é um contêiner de gerenciamento para os hubs de eventos (ou tópicos, na linguagem do Kafka). Ele fornece pontos de extremidade de rede integrados ao DNS e uma variedade de recursos de controle de acesso e gerenciamento de integração de rede, como [filtragem de IP](#), [ponto de extremidade de serviço de rede virtual](#) e [Link Privado](#).



Editores de eventos

Qualquer entidade que envie dados para um Hub de Eventos é um *editor de eventos* (sinônimo de *produtor de eventos*). Os editores de eventos podem publicar eventos usando HTTPS, AMQP 1.0 ou o protocolo Kafka. Os editores de eventos usam autorização baseada no Azure Active Directory com tokens JWT emitidos por OAuth2 ou um token SAS (Assinatura de Acesso Compartilhado) específico do Hub de Eventos para obter acesso à publicação.

Publicar um evento

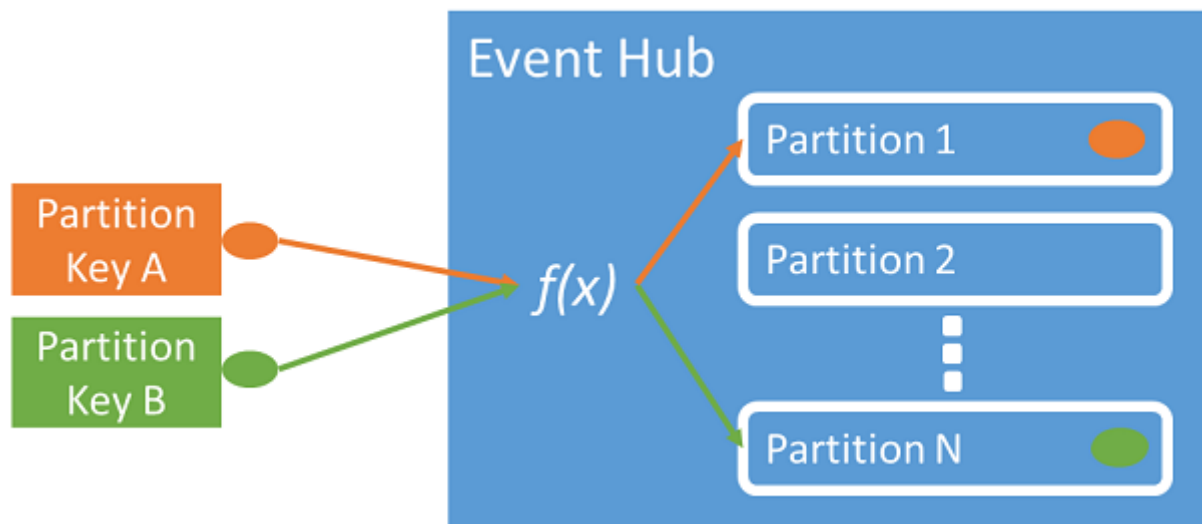
Você pode publicar um evento por meio do AMQP 1.0, do protocolo Kafka ou de HTTPS. Além da [API REST](#), o serviço Hubs de Eventos fornece bibliotecas de cliente [.NET](#), [Java](#), [Python](#), [JavaScript](#) e [Go](#) para publicar eventos em um hub de eventos. Para outras plataformas e runtimes, você pode usar qualquer cliente AMQP 1.0, como o [Apache Qpid](#).

A opção de usar AMQP ou HTTPS é específica para o cenário de uso. O AMQP requer o estabelecimento de um soquete bidirecional persistente, além do TLS (segurança de

nível de transporte) ou SSL/TLS. Embora o AMQP tenha custos de rede mais altos ao inicializar a sessão; o HTTPS requer sobrecarga adicional de TLS para cada solicitação. O AMQP apresenta um desempenho significativamente maior para editores frequentes e pode atingir latências muito menores quando usado com código de publicação assíncrono.

Você pode publicar eventos individualmente ou em lotes. Uma única publicação tem um limite de 1 MB, independentemente de ser um lote ou um evento único. A publicação de eventos maiores que esse limite será rejeitada.

A taxa de transferência do Hubs de Eventos é dimensionada pelo uso de partições e alocações de unidades de produtividade (veja a seguir). É uma prática recomendada que os editores permaneçam cientes do modelo de particionamento escolhido para um Hub de Eventos e especifiquem apenas uma *chave de partição* para atribuir eventos relacionados de forma consistente à mesma partição.



Os Hubs de Eventos garantem que todos os eventos que compartilham um valor de chave de partição sejam armazenados juntos e entregues em ordem de chegada. Se forem usadas chaves de partição com políticas de editor, a identidade do editor e o valor da chave de partição devem corresponder. Caso contrário, ocorrerá um erro.

Retenção de eventos

Os eventos publicados são removidos de um hub de eventos com base em uma política de retenção configurável e baseada em tempo. Veja a seguir alguns pontos importantes:

- O valor **padrão** e **menor** período de retenção possível é de **um dia (24 horas)**.
- Para o Hubs de Eventos **Standard**, o período de retenção máximo é de **sete dias**.
- Para Hubs de Eventos **Premium** e **Dedicados**, o período de retenção máximo é de **90 dias**.

- Se você alterar o período de retenção, ele será aplicável a todas as mensagens, incluindo aquelas que já estão no hub de eventos.

Os Hubs de Eventos mantêm os eventos por um período de retenção configurado que se aplica a todas as partições. Os eventos são removidos automaticamente quando o período de retenção é atingido. Se você especificar um período de retenção de um dia, o evento ficará indisponível exatamente 24 horas depois de ser aceito. Não é possível excluir eventos explicitamente.

Se você precisar arquivar eventos além do período de retenção permitido, poderá armazená-los automaticamente no Armazenamento do Azure ou no Azure Data Lake ativando o [recurso de Captura dos Hubs de eventos](#). Caso precise pesquisar ou analisar esses arquivos tão bem guardados, você poderá importá-los facilmente para o [Azure Synapse](#) ou outros repertórios e plataformas de análise similares.

O motivo para o limite dos Hubs de Eventos na retenção de dados com base no tempo é evitar que grandes volumes de dados históricos do cliente sejam interceptados em um repositório profundo que está indexado apenas por um carimbo de data/hora e que permita acesso sequencial. A filosofia arquitetônica aqui aplicada é de que os dados históricos precisam de indexação mais avançada e de acesso mais direto do que a interface de eventos em tempo real que os Hubs de Eventos ou o Kafka fornecem. Os mecanismos de fluxo de eventos não são adequados para desempenhar a função de data lakes ou de arquivos de longo prazo para fornecimento de eventos.

❗ Observação

O Hubs de Eventos é um mecanismo de fluxo de eventos em tempo real. Ele não foi projetado para ser usado em lugar de um banco de dados e/ou como um armazenamento permanente para fluxos de eventos mantidos infinitamente.


Quanto mais detalhado for o histórico de um fluxo de eventos, mais você precisará de índices auxiliares para localizar uma determinada fatia histórica de um fluxo específico. A inspeção de cargas de eventos e a indexação não estão no escopo de recursos dos Hubs de Eventos (ou do Apache Kafka). Portanto, os meios mais adequados para armazenar eventos históricos são bancos de dados, além de repositórios e mecanismos de análise especializados, como **Azure Data Lake Store**, **Azure Data Lake Analytics** e **Azure Synapse**.

A **Captura do Hubs de Eventos** se integra diretamente ao Armazenamento de Blobs do Azure e ao Azure Data Lake Storage e, por meio dessa integração, também permite o **fluxo de eventos diretamente para o Azure Synapse**.

Se você deseja usar o padrão de **Fornecimento de Eventos** para seu aplicativo, deve alinhar sua estratégia de instantâneo com os limites de retenção do Hubs de Eventos. Não se proponha a recompilar exibições materializadas de eventos brutos desde o começo. Você certamente se arrependerá dessa estratégia depois que seu aplicativo for bem utilizado na produção por um tempo e seu construtor de projeção exigir rotatividade de eventos de alteração durante anos tentando obter as alterações mais recentes em andamento.

Política de editor

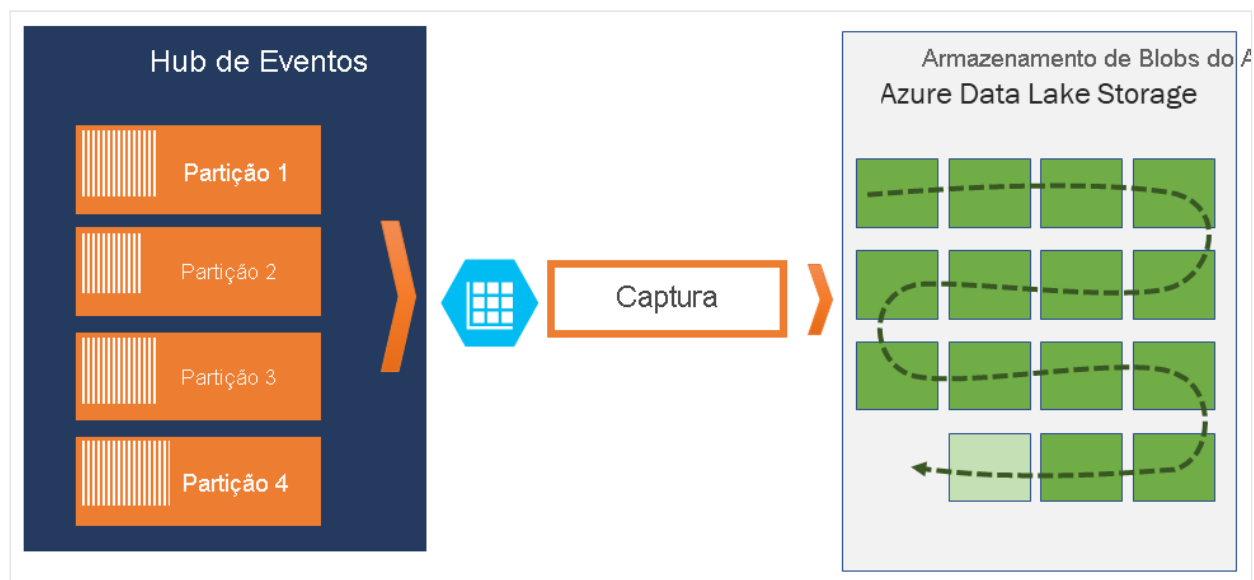
Os Hubs de Eventos permitem um controle granular sobre os editores de eventos por meio de *políticas do editor*. As políticas do editor são recursos de tempo de execução criado para facilitar um grande número de editores de eventos independentes. Com as políticas do editor, cada editor usa seu próprio identificador exclusivo ao publicar eventos em um hub de eventos usando o mecanismo a seguir:

HTTP	 Copiar
<pre>//<my namespace>.servicebus.windows.net/<event hub name>/publishers/<my publisher name></pre>	

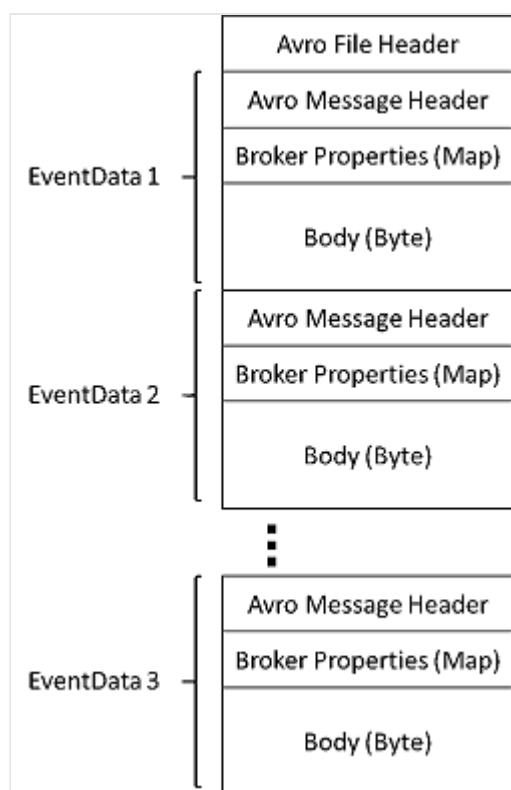
Você não precisa criar nomes de editor com antecedência, mas eles devem coincidir com o token SAS usado ao publicar um evento, para garantir identidades de editores independentes. Ao usar políticas de editor, o valor **PartitionKey** é definido como o nome do editor. Para funcionar adequadamente, esses valores devem corresponder.

Capturar

A [Captura dos Hubs de Eventos](#) permite que você capture automaticamente os dados de streaming nos Hubs de Eventos e salve-os em uma conta de Armazenamento de Blobs ou em uma conta de serviço do Azure Data Lake Storage de sua escolha. Você pode habilitar a Captura no portal do Azure e especificar um tamanho mínimo e a janela de tempo para executar a captura. A Captura dos Hubs de Eventos permite que você especifique sua própria conta de Armazenamento de Blobs do Azure e contêiner, ou uma conta de serviço do Azure Data Lake Storage. Uma delas será usada para armazenar os dados capturados. Os dados capturados são gravados no formato Apache Avro.



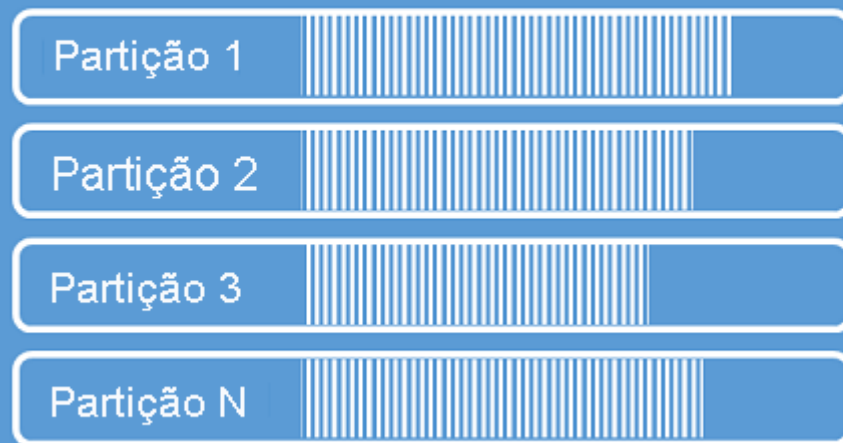
Os arquivos produzidos pela Captura de Hubs de Eventos têm o seguinte esquema Avro:



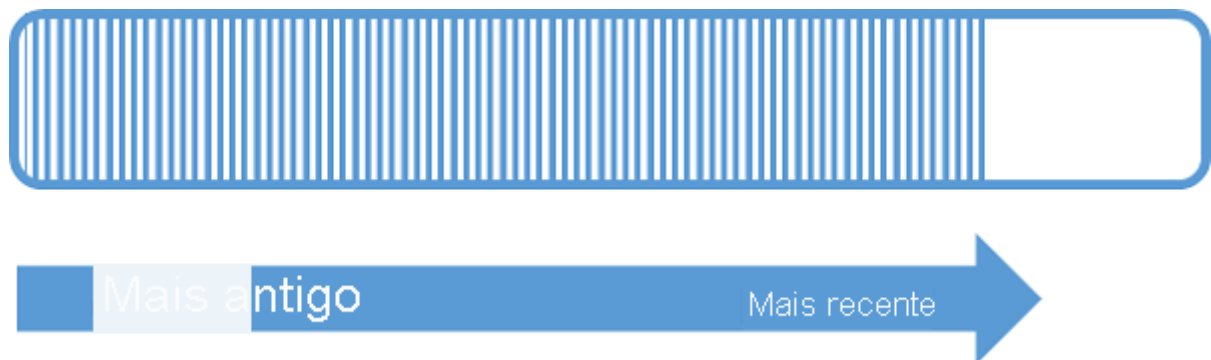
Partições

Os Hubs de Eventos organizam sequências de eventos enviados a um hub de eventos em uma ou mais partições. À medida que novos eventos chegam, eles são adicionados ao final dessa sequência.

Hub de Eventos



Uma partição pode ser considerada como um "log de confirmação". As partições armazenam dados de evento que contêm o corpo do evento, um recipiente de propriedades definido pelo usuário que descreve o evento e os metadados como o deslocamento na partição, o número na sequência de fluxo e o carimbo de data/hora do lado do serviço no qual ele foi aceito.



Vantagens do uso de partições

Os Hubs de Eventos são projetados para ajudar no processamento de grandes volumes de eventos, e o particionamento ajuda com isso de duas maneiras:

- Embora os Hubs de Eventos sejam um serviço de PaaS, há uma realidade física subjacente. Além disso, a manutenção de um log que preserve a ordem dos eventos exige que esses eventos sejam mantidos juntos no armazenamento subjacente e nas respectivas réplicas, o que resulta em um teto de taxa de transferência para esse log. O particionamento permite que vários logs paralelos sejam usados para o mesmo hub de eventos, multiplicando, portanto, a capacidade de taxa de transferência de E/S bruta disponível.

- Seus aplicativos precisam conseguir acompanhar o processamento do volume de eventos que estão sendo enviados para um hub de eventos. Isso pode ser complexo e exige uma capacidade substancial, de expansão e de processamento paralelo. A capacidade de um só processo de lidar com os eventos é limitada. Portanto, é necessário ter vários processos. As partições são como a sua solução alimenta esses processos, garantindo, ainda, que cada evento tenha um proprietário de processamento claro.

Número of partições

O número de partições é especificado no momento da criação de um hub de eventos. Ele precisa estar entre um e a contagem máxima de partições permitidas para cada tipo de preço. Para obter o limite de contagem de partições para cada camada, confira [este artigo](#).

Recomendamos que você escolha, pelo menos, o máximo de partições que deverão ser necessárias durante o pico de carga do seu aplicativo para esse hub de eventos específico. Não é possível alterar a contagem de partições para um hub de eventos após a criação dele, exceto para um hub de eventos em um cluster dedicado na camada Premium. A contagem de partições de um hub de eventos em um [cluster de Hubs de Eventos dedicados](#) pode ser [aumentada](#) após o hub de eventos ter sido criado, mas a distribuição de fluxos entre partições será alterada quando for feita como o mapeamento de chaves de partição para alterações de partições. Portanto, você deve tentar evitar essas alterações se a ordem relativa dos eventos é importante em seu aplicativo.

Definir o número de partições com o valor máximo permitido é tentador, mas sempre tenha em mente que os fluxos de eventos precisam ser estruturados de um jeito que você possa tirar proveito de várias partições. Se você precisar de preservação de ordem absoluta em todos os eventos ou apenas em alguns subfluxos, talvez não seja possível tirar proveito de muitas partições. Além disso, muitas partições tornam o lado de processamento mais complexo.

Quando se trata de preço, o número de partições existentes em um hub de eventos não é relevante. Isso depende do número de unidades de preço – [TUs \(unidades de produtividade\)](#) da camada standard, [PUs \(unidades de processamento\)](#) da camada premium e [CUs \(unidades de capacidade\)](#) da camada dedicada – do namespace ou do cluster dedicado. Por exemplo, um hub de eventos da camada standard com 32 partições ou uma partição gera exatamente o mesmo custo quando o namespace é definido com uma capacidade de 1 TU. Além disso, é possível escalar as TUs ou as PUs

no namespace ou as CUs do cluster dedicado independentemente da contagem de partições.

Mapeamento de eventos para partições

Você pode usar uma chave de partição para mapear dados de evento de entrada em partições específicas para fins de organização de dados. A chave de partição é um valor fornecido pelo remetente passado para um hub de eventos. Ela é processada por meio de uma função de hash estática, que cria a atribuição de partição. Se você não especificar uma chave de partição ao publicar um evento, uma atribuição de round robin será usada.

O editor de eventos só está ciente da sua chave de partição, não da partição para a qual os eventos são publicados. Essa desassociação de chave e partição isenta o remetente da necessidade de saber muito sobre o processamento de downstream. Uma identidade por dispositivo ou exclusiva do usuário é uma boa chave de partição, mas outros atributos, como geografia, também podem ser usados para agrupar eventos relacionados em uma única partição.

A especificação de uma chave de partição permite manter eventos relacionados juntos na mesma partição e na ordem exata em que eles chegaram. A chave de partição é uma cadeia de caracteres derivada do contexto do aplicativo e identifica a relação entre os eventos. Uma sequência de eventos identificados por uma chave de partição é um *fluxo*. Uma partição é um repositório de logs multiplexado para muitos desses fluxos.

ⓘ Observação

Embora você possa enviar eventos diretamente para as partições, não recomendamos isso, especialmente quando a alta disponibilidade é importante para você. Isso faz o downgrade da disponibilidade de um hub de eventos para o nível de partição. Para obter mais informações, confira [Disponibilidade e consistência](#).

Tokens SAS

Os hubs de eventos usam *assinaturas de acesso compartilhado*, que estão disponíveis nos níveis de namespace e de hub de eventos. Um token SAS é gerado a partir de uma chave de SAS e é um hash SHA de uma URL, codificado em um formato específico. Usando o nome da chave (política) e o token, os Hubs de Evento podem regenerar o hash e, portanto, autenticar o remetente. Normalmente, os tokens SAS para editores de

eventos são criados apenas com privilégios de **enviar** em um hub de eventos específico. Esse mecanismo de URL de token SAS é a base para a identificação de editor abordada na política do editor. Para saber mais sobre como trabalhar com SAS, confira [Autenticação de assinatura de acesso compartilhado com o Barramento de Serviço](#).

Consumidores de evento

Qualquer entidade que lê dados de evento de um hub de eventos é um *consumidor de eventos*. Todos os consumidores de Hubs de Eventos se conectam por meio de sessão do AMQP 1.0, e os eventos são entregues por meio da sessão à medida que são disponibilizados. O cliente não precisa buscar pela disponibilidade de dados.

Grupos de consumidores

O mecanismo de publicação/assinatura dos Hubs de Evento é habilitado por meio de *grupos de consumidores*. Um grupo de consumidores é uma exibição (estado, posição ou deslocamento) de todo um hub de eventos. Os grupos de consumidores permitem que vários aplicativos de consumo tenham um modo de exibição separado do fluxo de eventos e leiam o fluxo de forma independente em seu próprio ritmo e com seus próprios deslocamentos.


Em um arquitetura de processamento de fluxo, cada aplicativo downstream equivale a um grupo de consumidores. Se você quiser gravar dados de evento em um armazenamento de longo prazo, isso quer dizer que esse aplicativo gravador de armazenamento é um grupo de consumidores. O processamento de eventos complexos pode então ser executado por outro grupo separado de consumidores. Você pode acessar partições somente por meio de um grupo de consumidores. Há um grupo padrão de consumidores em um hub de eventos, e você pode criar até o **número máximo de grupos de consumidores** para o tipo de preço correspondente.

Pode haver no máximo cinco leitores simultâneos em uma partição por grupo de consumidores. No entanto **recomenda-se que haja apenas um receptor ativo em uma partição por grupo de consumidores**. Dentro de uma única partição, cada leitor recebe todas as mensagens. Se você tiver vários leitores na mesma partição, processará mensagens duplicadas. Você precisa lidar com isso em seu código, o que pode não ser trivial. No entanto, é uma abordagem válida em alguns cenários.

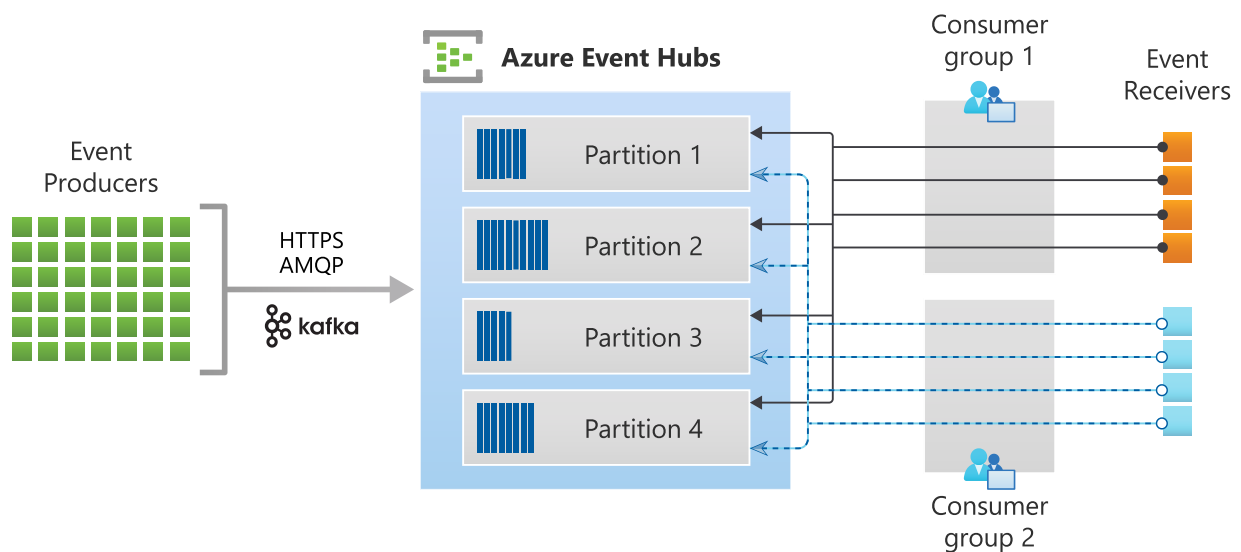
Alguns clientes oferecidos pelos SDKs do Azure são agentes inteligentes de consumidor que gerenciam automaticamente os detalhes necessários para garantir que cada partição tenha um único leitor e que todas as partições de um hub de eventos estejam sendo lidas. Isso permite que o código se concentre no processamento dos eventos

que estão sendo lidos do hub de eventos para que possa ignorar muitos dos detalhes das partições. Para obter mais informações, confira [Conectar-se a uma partição](#).

Os exemplos a seguir mostram a convenção de URI do grupo de consumidores:

HTTP	 Copiar
<pre>//<my namespace>.servicebus.windows.net/<event hub name>/<Consumer Group #1> //<my namespace>.servicebus.windows.net/<event hub name>/<Consumer Group #2></pre>	

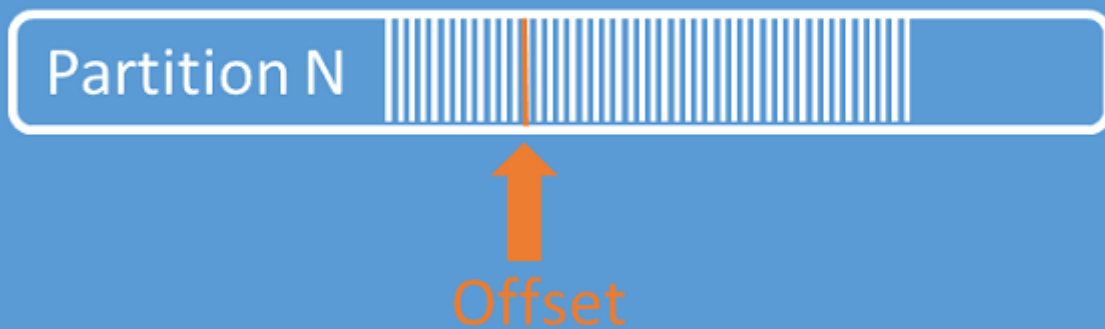
A figura a seguir mostra a arquitetura de processamento de fluxo dos Hubs de Eventos:



Deslocamentos de fluxo

Um *deslocamento* é a posição de um evento dentro de uma partição. Você pode pensar em um deslocamento como um cursor do lado do cliente. O deslocamento é uma numeração em bytes do evento. Esse deslocamento permite que um consumidor de eventos (leitor) especifique um ponto no fluxo de eventos a partir do qual deseja começar a ler eventos. Você pode especificar o deslocamento como um carimbo de data hora ou um valor de deslocamento. Os consumidores são responsáveis por armazenar seu próprios valores de deslocamento fora do serviço de Hubs de Evento. Dentro de uma partição, cada evento inclui um deslocamento.

Event Hub



Ponto de verificação

Ponto de verificação é um processo pelo qual os leitores marcam ou confirmam sua posição em uma sequência de eventos da partição. O ponto de verificação é responsabilidade do consumidor e ocorre em uma base por partição dentro de um grupo de consumidores. Essa responsabilidade significa que, para cada grupo de consumidores, cada leitor de partição deve manter o controle da sua posição atual no fluxo de eventos e pode informar o serviço quando considerar o fluxo de dados concluído.

Se um leitor se desconecta de uma partição, ao se reconectar, ele começa a ler no ponto de verificação que foi anteriormente enviado pelo último leitor dessa partição nesse grupo de consumidores. Quando o leitor se conecta, ele passa esse deslocamento para o hub de eventos para especificar o local para começar a ler. Assim, você pode usar o ponto de verificação para marcar eventos como "concluídos" por aplicativos de downstream e oferecer resiliência caso ocorra um failover entre leitores em execução em máquinas diferentes. É possível retornar aos dados mais antigos, especificando um deslocamento inferior desse processo de ponto de verificação. Por meio desse mecanismo, o ponto de verificação permite resiliência de failover e reprodução de fluxo de eventos.

❗ Importante

Os deslocamentos são fornecidos pelo serviço Hubs de Eventos. É responsabilidade do consumidor criar um ponto de verificação enquanto os eventos são processados.

❗ Observação

Se você estiver usando o Armazenamento de Blobs do Azure como repositório de pontos de verificação em um ambiente que dá suporte a uma versão diferente de SDK do Storage Blob das que normalmente estão disponíveis no Azure, precisará usar o código para alterar a versão de API do serviço de Armazenamento para a versão compatível com esse ambiente. Por exemplo, se você estiver executando o **Hubs de Eventos em uma versão 2002 do Azure Stack Hub**, a versão mais recente disponível para o serviço de armazenamento será a 2017-11-09. Nesse caso, é necessário usar o código para direcionar a versão de API do serviço de armazenamento para a 2017-11-09. Para saber como direcionar uma versão de API específica do Armazenamento, confira estes exemplos no GitHub:

- [.NET](#)
- [Java](#)
- [JavaScript](#) ou [TypeScript](#)
- [Python](#)

Tarefas comuns do consumidor

Todos os consumidores de Hubs de Eventos se conectam por meio de uma sessão do AMQP 1.0, um canal de comunicação bidirecional com reconhecimento de estado. Cada partição tem uma sessão de AMQP 1.0 que facilita o transporte de eventos separados por partição.

Conectar-se a uma partição

Ao se conectar a partições, é comum usar um mecanismo de leasing para coordenar conexões de leitores com partições específicas. Isso possibilita que cada partição em um grupo de consumidores tenha apenas um leitor ativo. Os leitores de ponto de verificação, leasing e gerenciamento são simplificados por meio dos clientes nos SDKs do Hubs de Eventos, que atuam como agentes inteligentes de consumidor. Estes são:

- O [EventProcessorClient](#) para .NET
- O [EventProcessorClient](#) para Java
- O [EventHubConsumerClient](#) para Python
- O [EventHubConsumerClient](#) para JavaScript/TypeScript

Ler eventos

Depois de uma sessão do AMQP 1.0 e o link ser aberto para uma partição específica, os eventos são entregues ao cliente AMQP 1.0 pelo serviço de Hubs de Evento. Esse

mecanismo de entrega permite uma maior taxa de transferência e menor latência que mecanismos baseado em pull, como HTTP GET. Como os eventos são enviados para o cliente, cada instância de dados do evento contém metadados importantes, como o deslocamento e número da sequência que são usados para facilitar o ponto de verificação na sequência de eventos.

Dados de evento:

- Deslocamento
- Número de sequência
- Corpo
- Propriedades do usuário
- Propriedades do sistema

Gerenciar o deslocamento é de sua responsabilidade.

Próximas etapas

Para saber mais sobre Hubs de Eventos, acesse os seguintes links:

- [Introdução aos Hubs de Eventos](#)
 - [.NET](#)
 - [Java](#)
 - [Python](#)
 - [JavaScript](#)
- [Guia de programação dos Hubs de Eventos](#)
- [Disponibilidade e consistência nos Hubs de Eventos](#)
- [Perguntas frequentes sobre os Hubs de Eventos](#)
- [Exemplos de Hubs de Eventos](#)

Conteúdo recomendado

[Hubs de Eventos – Capturar eventos de streaming usando o portal do Azure - Azure Event Hubs](#)

Este artigo descreve como habilitar a captura de eventos de streaming por meio de Hubs de Eventos do Azure usando o portal do Azure.

[Hubs de Eventos do Azure – Eventos do Exchange usando protocolos diferentes - Azure Event Hubs](#)

Este artigo mostra como consumidores e produtores que usam protocolos diferentes (AMQP, Apache Kafka e HTTPS) podem trocar eventos ao usar os Hubs de Eventos do Azure.

Cotas e limites – Hubs de Eventos do Azure - Azure Event Hubs

Este artigo fornece limites e cotas para Hubs de Eventos do Azure. Por exemplo, número de namespaces por assinatura, número de hubs de eventos por namespace.

Exemplos – Hubs de Eventos do Azure - Azure Event Hubs

Este artigo fornece uma lista de exemplos para Hubs de Eventos do Azure que estão no GitHub.

Mostrar mais ▼